Big Data

Técnicas e tecnologias para extração de valor dos dados

Casa do Código

ROSANGELA MARQUESONE

ISBN

Impresso e PDF: 978-85-5519-231-9

EPUB: 978-85-5519-232-6

MOBI: 978-85-5519-233-3

Você pode discutir sobre este livro no Fórum da Casa do Código: http://forum.casadocodigo.com.br/.

Caso você deseje submeter alguma errata ou sugestão, acesse http://erratas.casadocodigo.com.br.

AGRADECIMENTOS

O lado bom dos agradecimentos é você perceber quantas pessoas queridas tem em sua vida. Sou grata à minha família, por todo o apoio em tudo que eu almejo em minha vida. Vocês são minha base! Agradeço também ao meu marido Evandro, pelo incentivo, parceria e amor. É muito bom tê-lo ao meu lado.

Além da família, tive a sorte de ter grandes mentores em minha vida, inspirando-me e dando suporte à minha carreira. Thesko, Karen e Tereza, sou grata por todas as conversas e lições de vida.

Agradeço à minha orientadora Tereza C. M. B. Carvalho, ao diretor Wilson V. Ruggiero e aos amigos do Laboratório de Arquitetura e Redes de Computadores da Universidade de São Paulo (LARC-USP). Tenho orgulho de fazer parte desse time! Agradeço também aos profissionais e amigos do Laboratório de Sustentabilidade (Lassu), Fundação Instituto de Administração (FIA), Fórum de IoT, UTFPR, BSI Tecnologia e SDI Sistemas.

Sempre tive em mente que as oportunidades não aparecem, você precisa criá-las. Foi buscando uma oportunidade que cheguei à Casa do Código, onde apostaram em minha iniciativa e me deram suporte na escrita do livro. Vivian, obrigada pelas valiosas revisões durante todo esse processo.

Um agradecimento especial aos alunos dos quais tive a oportunidade de compartilhar conhecimento sobre Big Data. Foram vocês os principais incentivadores para a criação deste livro.

Por fim, agradeço aos amigos que certamente brindarão comigo essa realização: Thesko, Josane, Thiago, Valéria, Brito, Paschoal, Carlos, Shido, Josi, Jac, Fabiana e Juliana.

SOBRE A AUTORA

Rosangela de Fátima Pereira Marquesone é pesquisadora nas áreas de computação em nuvem e Big Data, com parceria entre a Universidade de São Paulo (USP) e Ericsson Research — Suécia, pelo Laboratório de Arquitetura e Redes de Computadores (LARC-USP).

Possui artigos publicados na área de tecnologias de Big Data e já ministrou mais de 300 horas de palestras e aulas sobre o tema para empresas, entidades públicas e programas de MBA da USP e Fundação Instituto de Administração (FIA).

Atua também como revisora de código no programa Nanodegree em Análise de Dados da rede de cursos on-line Udacity. Fez parte do corpo docente do departamento de computação da Universidade Tecnológica Federal do Paraná (UTFPR) no período de 2011 a 2012.

Graduou-se em Administração de Empresas pela Universidade Estadual do Norte do Paraná (UENP) em 2004, e em Análise e Desenvolvimento de Sistemas pela UTFPR em 2011. Concluiu o curso de Especialização Lato Sensu em Tecnologia Java pela UTFPR em 2010.

Atualmente, é mestranda em Engenharia de Computação pela Escola Politécnica da Universidade de São Paulo (Poli-USP). Acredita profundamente no poder da inovação na vida, nos negócios e na sociedade.

Mais informações podem ser encontradas em seu perfil no LinkedIn: https://www.linkedin.com/in/rosangelafpm.

PREFÁCIO

Público-alvo

Este livro é indicado para estudantes universitários, pesquisadores e profissionais que desejam compreender o que é Big Data, o motivo desse tema ser tão discutido atualmente e o que faz um profissional que atua nessa área. Embora tenha um viés técnico, o livro não é destinado somente aos profissionais da área de computação. As tecnologias e soluções de Big Data são apresentadas em uma abordagem mais conceitual, com o objetivo de detalhar as características e capacidades de cada uma delas.

O enfoque do livro é dado aos processos existentes em um projeto de Big Data. Para isso, cada capítulo foi planejado para apresentar informações sobre as principais atividades em um projeto, desde a captura até a visualização de dados.

A partir dessa estrutura, profissionais de diferentes áreas que desejam atuar com esse tema poderão também se beneficiar do conteúdo do livro, obtendo informações relevantes para inspirá-los na aplicação de Big Data em sua área de atuação.

Por abordar técnicas e linguagens de forma abrangente, o livro não é indicado ao leitor que busca um material de estudo que o capacite em uma tecnologia ou técnica específica. Porém, o leitor pode utilizar o livro como base para identificar quais áreas de estudo em relação a Big Data ele pode se aprofundar.

Ao término da leitura deste livro, o leitor poderá compreender as diversas possibilidades existentes para atuar nesse universo tão promissor. Também compreenderá o ciclo existente em um projeto de Big Data, descobrindo quais são os aspectos e desafios existentes.

Espero que o conhecimento adquirido com a leitura do livro sirva de motivação para os leitores mergulharem com maior profundidade nesse tema.

Organização do livro

Durante minha jornada em pesquisas e aulas sobre Big Data, ficou evidente para mim o quanto esse conceito é recente, porém apresenta um imenso potencial. Percebi também que Big Data tem despertado o interesse de profissionais de diferentes áreas, tais como agricultura, medicina, finanças, telecomunicação e varejo.

Durante as aulas que ministrei para esses profissionais, a maioria das dúvidas era relacionada à implementação de um projeto de Big Data, às indicações sobre como dar início a um projeto e às mudanças organizacionais necessárias para atuar com Big Data.

Partindo desses e outros questionamentos, este livro foi planejado para apresentar aspectos das fases principais em projetos de Big Data: captura, armazenamento, processamento, análise e visualização de dados. Para seguir esse planejamento, o livro está organizado em seis capítulos.

Antes de apresentar detalhes sobre cada uma das fases de um projeto de Big Data, você encontrará no *Capítulo 1* uma visão geral sobre Big Data e os tipos de dados existentes nesse contexto. Serão apresentados os famosos 3 Vs de Big Data (volume, variedade e velocidade), as tendências para o aumento de volume de dados digitais no decorrer dos anos, e a aplicabilidade dos dados gerados por humanos e por máquinas.

A fase de captura e armazenamento dos dados será apresentada no *Capítulo 2*. Ele contém exemplos de dados utilizados nas soluções atuais de Big Data, bem como os novos modelos de armazenamento por meio de tecnologias NoSQL.

No *Capítulo 3* você confere a fase de processamento de dados. Aqui são apresentadas as tecnologias criadas para obter processamento escalável de grande volume de dados. Serão apresentados os frameworks Hadoop e Storm, sendo o primeiro uma das soluções para processamento em lote e o segundo uma solução para processamento de dados em *streaming*.

A fase de análise de dados é apresentada no *Capítulo 4*, sendo abordadas as técnicas usadas nas diferentes análises, incluindo mineração de dados e aprendizado de máquina. Você encontrará exemplos de aprendizado supervisionado e não supervisionado, tais como classificação e agrupamento.

O *Capítulo 5* é destinado à fase de visualização de dados. O propósito aqui é enfatizar o potencial que uma visualização efetiva dos dados oferece, gerando meios intuitivos para representar uma análise. Também serão apresentados recursos gráficos que podem acelerar o aumento de percepções no processo de tomada de decisão. Em todas essas fases, serão apresentados exemplos práticos das tecnologias existentes em um projeto de Big Data.

Por fim, no último capítulo do livro, *Capítulo 6*, você encontrará considerações adicionais sobre Big Data. Serão abordados tópicos como o perfil do profissional cientista de dados, as tendências de Big Data para os próximos anos, a questão da privacidade dos dados e uma reflexão sobre os novos modelos de negócios gerados a partir desse conceito.

Espero que este livro possa lhe motivar a mergulhar nesse tema e assim auxiliar na criação de soluções inovadoras que Big Data pode oferecer. Boa leitura!

Código-fonte

Para falar com a autora e buscar materiais adicionais do livro,

acesse o site: http://www.livrobigdata.com.br.

Além do site, todos os códigos e bases de dados utilizados no livro estão disponíveis no GitHub:

https://github.com/rosangelapereira/livrobigdata.git

Para fazer uma cópia local desse conteúdo, utilize o seguinte comando:

```
$ git clone https://github.com/rosangelapereira/livrobigdata.git
```

DEDICATÓRIA

Àquela que me dá força e luz para seguir minha jornada. Este livro é dedicado a você, mãe Maria.

"É um erro capital teorizar antes de ter dados. Sem se perceber, começa-se a distorcer os fatos para ajustá-los às teorias, em vez de mudar as teorias para que se ajustem aos fatos."

— Arthur Conan Doyle, em *Sherlock Holmes*

Sumário

Capítulo 1

INTRODUÇÃO A BIG DATA

"O que sabemos é uma gota; o que ignoramos é um oceano." — Isaac Newton

Acredito que não importa qual seja sua profissão, seu cargo e as pessoas com quem você conversa, você possivelmente já deve ter ouvido falar em algum momento nessa expressão tão popular atualmente: Big Data. Essa minha premissa ocorre pelo fato de que não é preciso muito esforço para encontrarmos uma notícia referente a esse termo nos dias atuais.

Seja em sites, jornais ou revistas das áreas de astronomia, biologia, educação, economia, política ou até culinária, podemos encontrar alguma publicação que relate o potencial e as características de Big Data. De fato, Big Data tem sido alvo de muita atenção no mundo dos negócios, no governo e no meio acadêmico.

Podemos encontrar casos de uso em que Big Data permitiu a redução do número de fraudes, aumento de lucros, conquista de eleitores, redução de custos na produção, eficiência energética, aumento de segurança, entre outros benefícios tão almejados em diversos domínios. Muito embora o interesse esteja em alta, Big Data ainda é um termo incipiente, gerando incertezas sobre sua definição, características, aplicabilidade e desafios.

Quais dados fazem parte do contexto de Big Data? Qual a definição desse conceito? Como obter dados de diferentes fontes?

Como extrair valor a partir dos dados? Qual a infraestrutura necessária para criar uma solução de Big Data? Quais habilidades são necessárias para se atuar com Big Data?

Essas são apenas algumas das questões geradas por profissionais interessados nesse tema. Mas vamos com calma.

Para dar início ao esclarecimento dessas e de outras questões, você verá neste capítulo uma visão inicial sobre Big Data, que inclui a definição desse conceito e a descrição dos tipos de dados existentes nesse cenário.

Além dessas informações, será também apresentado um resumo dos processos em um projeto de Big Data e os mitos ainda existentes sobre o termo. Acredito que esse conteúdo servirá de base para a compreensão das demais questões, abordadas nos próximos capítulos.

1.1 POR QUE ESTAMOS NA ERA DOS DADOS

Suponha que estamos em 1996. Ao acordar, desligo meu despertador e me preparo para ir ao trabalho. Ao sair de casa, meu telefone fixo toca e, ao atender, a secretária da empresa em que trabalho me avisa que estou atrasada para a reunião que havia começado a uma hora.

Corro para pegar minha agenda dentro da bolsa e vejo que de fato havia marcado a reunião para aquele horário. Peço desculpas à secretária e aviso que irei rapidamente para a empresa.

Arrumo-me às pressas e saio de casa na expectativa que um táxi apareça rapidamente, para que eu possa chegar o quanto antes na reunião. Por sorte, um taxista aparece em 10 minutos.

Chego na empresa, porém percebo que esqueci de levar os

relatórios que havia elaborado para apresentar aos gerentes. E agora? Ligo para meu marido que está em casa e peço para ele me enviar uma cópia via fax. Assim ele faz, e consigo finalmente participar da reunião.

Bem, poderia dar continuidade ao relato de um dia de trabalho no ano de 1996, mas acredito que apenas essa breve descrição já foi suficiente para percebermos o quanto a tecnologia da informação e comunicação transformou nosso dia a dia nos últimos anos.

Atualmente, é comum usarmos nosso smartphone desde o primeiro instante em que acordamos, por meio de um alarme com nossa música favorita e por intervalos de tempos pré-determinados. Nosso smartphone também pode nos avisar antecipadamente o horário de uma reunião, para que assim possamos evitar esquecimentos.

Enquanto tomamos café, podemos solicitar um serviço de transporte de passageiros por meio de um aplicativo. Se necessitamos de um documento que não esteja conosco, podemos facilmente acessar a internet e buscá-lo em um serviço de computação em nuvem para armazenamento de dados.

O exemplo também nos revela que a tecnologia está em constante evolução. Vinte anos se passaram e temos atualmente uma variedade de soluções capazes de facilitar nossas ações diárias, transformar o modo como nos comunicamos e gerar novas estratégias de negócios.

Por exemplo, você é capaz de imaginar como seria sua rotina sem os recursos tecnológicos disponíveis atualmente? Para auxiliar essa compreensão, verifique a tabela mais adiante e perceba como a tecnologia tem influência direta na maneira com que realizamos nossas atividades. Seja para lazer, viagens, compras ou trabalho, ela nos proporciona facilidades que antes eram inimagináveis.

Categoria	Como ocorre atualmente
Viagem	Comparação de preços de passagens; Compra de passagem pela internet; Check-in online; Recomendação de serviços de hospedagem; Serviços de reserva de hospedagem; Definição de trajeto por auxílio de GPS.
Trabalho	Reuniões por videoconferência; Agenda de compromissos online; Hospedagem de arquivos online; Serviços de financiamento coletivo (*crowdfunding*); Busca e candidatura de vagas de trabalho online.
Lazer	Serviços de streaming de filmes, seriados e músicas; Compartilhamento de momentos em redes sociais; Leitura de livros eletrônicos; Jogos online.
Compras	Compras via comércio eletrônico; Avaliação online de produtos; Comparação de preços; Compras coletivas; Pedidos online de serviços alimentícios; SAC online; Internet banking.

E você sabe o que essa diversidade de serviços existentes tem em comum? A quantidade de dados que eles geram. Os avanços em hardware, software e infraestrutura de redes foram os responsáveis para que chegássemos à "era dos dados".

Nos anos 80 e 90, a maioria dos dados era armazenada em formato analógico. Discos de vinil, fitas de vídeo VHS e fitas cassete eram meios comuns para armazenar um dado. Tais recursos, comparados com o formato digital, eram frágeis e dificultavam o seu compartilhamento.

Embora esses recursos ainda existam, eles foram gradativamente sendo substituídos por recursos com tecnologias digitais. Isso é tão real que, um estudo feito pela revista *Science* apontou que, em 1996, somente 0.8% dos dados eram armazenados em formato digital, enquanto em 2007 a quantidade de dados digitais já era de 94%.

Essa transformação é facilmente percebida no mundo atual. Por exemplo, você saberia responder como as pessoas utilizam e armazenam uma música, um vídeo ou um documento nos dias atuais? Tenho certeza de que a resposta da grande maioria dos leitores envolve um dispositivo digital.

Comparando os recursos que temos atualmente com o que

tínhamos alguns anos atrás, imagino que você possa estar pensando: como ocorreu essa transformação? Conforme apresentado na figura a seguir, uma série de fatores ocorreu com o passar dos anos, possibilitando o avanço tecnológico atual.

Certamente, a internet foi e continua sendo um dos fatores mais influentes no crescimento dos dados. Porém, além dela, outro fator que causou grande impacto foi a ampla adoção de dispositivos móveis nos últimos anos.

O poder de armazenamento, os recursos computacionais e o acesso à internet oferecidos por esses dispositivos ampliaram não somente a quantidade de dados únicos gerados, mas também a quantidade de vezes que eles eram compartilhados. Um vídeo gerado em um smartphone, por exemplo, pode facilmente ser compartilhado nas redes sociais, enviado por aplicativos de troca de mensagens e disponibilizado em diversos sites da Web.

Agora imagine esse compartilhamento sendo feito diariamente por parte dos 168 milhões de aparelhos de smartphones existentes somente no Brasil. Esse amplo compartilhamento é um dos fatores que levaram ao crescimento exponencial dos dados. Mídias sociais como o Facebook, Twitter, Pinterest e Instagram são exemplos de soluções que alavancaram esse compartilhamento e, consequentemente, a comunicação entre os usuários.

Figura 1.1: Principais fatores para o aumento do volume de dados

Além da crescente adoção de dispositivos móveis, dois outros fatores que contribuíram significativamente para o aumento do volume de dados foram o aumento do poder de processamento e a redução de custo de armazenamento de dados. Para exemplificar essas mudanças, temos o fato de que a primeira versão do iPhone, lançada em 2007, possuía uma capacidade de processamento muito superior a todo o sistema computacional utilizado para levar o homem à lua nos anos 60. Imagine então se compararmos esse sistema com a última versão do aparelho? Esse avanço é um dos resultados previstos pela Lei de Moore, que observou que a capacidade de processamento dos computadores dobraria aproximadamente a cada 18 meses.

Em relação ao armazenamento de dados, enquanto em 1990 o custo para armazenar 1 megabyte era de aproximadamente U$ 12.000, a média de custo atual é de apenas U$ 0,03. Ou seja, mesmo que empresas já identificassem possibilidades de extração de valor sobre uma vasta quantia de dados na década de 90, elas optavam muitas vezes por descartá-los, devido ao alto custo de armazenamento.

Ao passo que o volume de dados crescia e novas tecnologias habilitadoras para a geração desses dados eram criadas, empresas de diversos segmentos passaram a perceber o potencial que diferentes tipos de dados poderiam oferecer, seja para aperfeiçoar um processo, aumentar a produtividade, melhorar o processo de tomada de decisão, ou até mesmo para desenvolver novos produtos e serviços. A partir dessa visão, passam a surgir soluções que utilizam uma série de dados, internos e externos, para inúmeros propósitos.

Temos como exemplo a indústria varejista, que com a adoção de etiquetas de identificação por radiofrequência, ou RFID (do inglês *Radio-Frequency IDentification*), as empresas desse segmento

passaram a otimizar seu processo de armazenamento, catalogação e transporte de mercadorias. Assim, tiveram uma maior agilidade no gerenciamento de seus processos. Na agricultura, temos a utilização de redes de sensores, que coletavam fluxos de dados em tempo real para fornecer suporte às ações referentes ao processo de plantação, cultivo e colheita.

Entretanto, mesmo havendo um avanço na quantidade de dados usada como apoio para as soluções, um estudo da EMC apontou que, em 2012, de todos os 643 exabytes de dados existentes no mundo digital, somente 3% foram utilizados. Ou seja, podemos concluir que ainda há um vasto número de oportunidades a serem exploradas.

Diante desse fato, pesquisadores consideram que estamos vivenciando o início de uma nova revolução industrial, na qual os dados passam a ser elementos chaves dessa mudança. Podemos concluir, portanto, que esse é o momento ideal para criarmos oportunidades a partir dos dados.

1.2 TODOS OS VS DE BIG DATA

É comum, ao ouvir pela primeira vez o termo Big Data, pensarmos que ele está unicamente relacionado a um grande volume de dados (o que é normal, já que o nome diz exatamente isso). Entretanto, o volume de dados não é sua única característica.

Além dessa, pelo menos outras duas propriedades devem ser consideradas: a variedade e a velocidade dos dados. Tais propriedades são popularmente denominadas os 3 Vs de Big Data, conforme apresentado na figura a seguir.

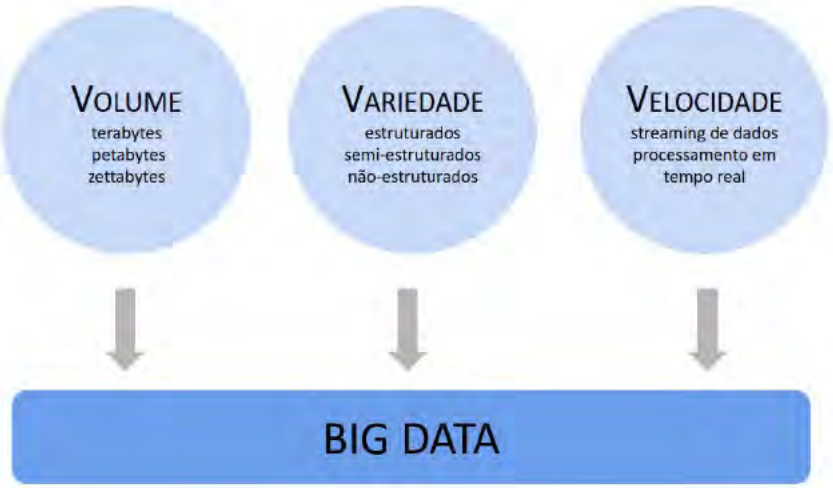

Figura 1.2: Os 3 Vs de Big Data

Volume

O atributo volume é a característica mais significativa no conceito de Big Data. Ele faz referência à dimensão sem precedentes do volume de dados.

Estimativas geradas por consultorias de TI relatam que, de todos os dados digitais existentes, 90% foram criados nos últimos dois anos. Mas você saberia responder qual é a origem para tantos dados? Confira a seguir algumas estatísticas que nos fazem perceber o que esse volume representa:

- A cada segundo, cerca de 40.000 buscas são realizadas no Google.
- A empresa Walmart manipula mais de 1 milhão de transações dos clientes por hora.
- A rede social Facebook contabilizou em junho de 2016 uma média de 1.13 bilhão de usuários, 2.5 bilhões de compartilhamentos e 2.7 bilhões de "curtidas" diariamente.

- A rede social de compartilhamento de fotos Instagram recebe atualmente cerca de 80 milhões de fotos por dia.
- Em 2013, a plataforma de blogs WordPress relatou a quantidade de 42 milhões de comentários por mês, entre os 3.6 bilhões de páginas existentes na plataforma.

Uau! Você consegue imaginar a quantidade de espaço em disco necessária para armazenar todos esses dados? Esse volume gerou uma mudança de escala de petabytes para exabytes e zettabytes de dados nos últimos anos.

Segundo a consultora EMC, estima-se que, em 2013, havia 4.4 zettabytes (4.4 trilhões de gigabytes) de dados em todo o mundo, e que esse número deverá chegar a 44 zettabytes em 2020. Dada essa dimensão, a complexidade envolvida com essa mudança de escala torna-se difícil de ser mensurável por nós humanos.

Para tentarmos compreender o impacto desse crescimento, imagine se a população mundial que hoje (2016) possui 7.4 bilhões de pessoas aumentasse para 1 trilhão daqui 10 anos. Como prover, adaptar e gerir recursos para suportar esse crescimento populacional tão expressivo e em tão pouco tempo?

Analogamente falando, é isso que vem ocorrendo com os dados digitais na atualidade: um crescimento exorbitante, que requer mudanças de gerenciamento e controle em todos os âmbitos. No aspecto técnico, especificamente, mudanças são necessárias para superar desafios em relação à escalabilidade, eficiência, custo e complexidade para analisar os dados, uma vez que as tecnologias tradicionais não foram projetadas para suportar esse volume.

Uma dúvida frequente relacionada ao volume de dados é a identificação de quando um determinado conjunto de dados pode ser considerado Big Data. Será que necessito de uma solução de Big Data somente se possuo petabytes de dados? A resposta é não.

O tamanho dos dados é algo relativo quando se fala em Big Data. Por exemplo, uma clínica médica pode necessitar de soluções de Big Data para visualizar imagens de 30 gigabytes de dados. Por outro lado, uma empresa de biotecnologia pode necessitar de tecnologias capazes de analisar 300 gigabytes de dados genéticos; e uma empresa na área de entretenimento pode necessitar de uma solução capaz de processar 2 petabytes de dados.

Dessa forma, o que de fato define se o atributo volume requer uma tecnologia de Big Data é a limitação das ferramentas tradicionais para lidar com determinado volume de dados.

Variedade

O banco de dados relacional é o modelo de armazenamento de dados mais usado nos últimos 40 anos pelas corporações. Nesse modelo, dados são armazenados em formato de tabelas, de acordo com uma estrutura previamente definida.

Isso significa que, antes de armazenar alguma informação, é necessário definir a estrutura, a sequência, o tamanho e os tipos de dados em questão. Outra notável característica desse modelo é o suporte à propriedade ACID, que garante a integridade dos dados por meio dos seguintes recursos:

- **Atomicidade:** garante que todas as alterações realizadas por uma transação serão efetivadas no banco de dados, ou nenhuma delas, caso ocorra algum problema. Ou seja, não há atualização parcial da transação.
- **Consistência:** nesse caso é garantido que novas transações somente serão completadas se elas não ferirem nenhuma regra do banco de dados que possa torná-lo inconsistente.
- **Isolamento:** propriedade que permite que os eventos em uma transação não interfiram nos eventos de outra

transação concorrente.

- **Durabilidade:** garante que o resultado de toda transação executada com sucesso deverá ser mantido no banco de dados, mesmo na ocorrência de falhas.

Embora seja muito eficiente e aplicado a diversos cenários, o banco de dados relacional é projetado para armazenar majoritariamente dados estruturados, isto é, dados com esquemas rígidos e adequados para o formato de tabelas. Isso se torna uma limitação para Big Data, uma vez que esse termo também inclui dados semiestruturados e não estruturados.

Podemos entender como dados semiestruturados aqueles que possuem uma estrutura pré-definida, porém não com o mesmo rigor dos dados relacionais. Essas estruturas são usadas normalmente apenas como um meio de marcação dos dados, como é o caso dos arquivos no formato JSON (*JavaScript Object Notation*) e XML (*eXtensible Markup Language*).

Na classe de dados não estruturados estão inclusos os vídeos, imagens, e alguns formatos de textos. Por não terem um formato que pode ser facilmente armazenado em tabelas, eles se tornam complexos para serem processados em ferramentas tradicionais de armazenamento e gerenciamento de dados.

Mesmo com a predominância do uso de sistemas de bancos de dados relacionais no mercado, há estimativas que, de todos os dados disponíveis globalmente, apenas 20% são considerados dados estruturados. Diante desse fato, onde 80% dos dados restantes devem ser armazenados?

Estes precisam de um modelo que ofereça flexibilidade quanto a sua estrutura, que não exija um esquema rígido previamente definido (como é exigido em bancos de dados relacionais), e que sejam adequados para ambientes distribuídos, dependendo do

volume. Esse fato trouxe a necessidade de não somente uma solução que fosse complementar aos bancos de dados relacionais, mas também de uma variedade de soluções e tecnologias, cada qual para atender às necessidades específicas de uma aplicação de Big Data.

Quando nos referimos à variedade, também cabe destacar a variedade de áreas das quais Big Data tem sido aplicado. O que antes era limitado a empresas do Vale do Silício, atualmente os dados e as tecnologias são utilizados em diversos segmentos, como por exemplo:

- Na área governamental, com a utilização de tecnologias para rastrear os perfis dos eleitores na campanha do presidente dos Estados Unidos, Barack Obama;
- No setor financeiro, com soluções na área de análise de risco e detecção de fraude;
- Na área de transporte e automação, com o monitoramento de tráfego e rastreamento de carga;
- No setor de varejo, com a possibilidade de gerar ofertas baseadas na análise de vendas e no perfil do consumidor;
- Nas diversas possibilidades na área de marketing, por meio da análise de redes sociais;
- Na área de seguros, com a possibilidade de ofertas de planos baseados no comportamento do segurado.

Ou seja, há uma diversidade de dados sendo utilizada por uma variedade de soluções, cada qual com necessidades específicas. Embora a variedade dos dados seja algo já explorado, ainda são poucas as empresas que conseguem criar soluções eficientes por meio dessa abordagem, principalmente em relação à integração de

tais dados.

Por não ser explicitamente claro o valor que essa variedade de dados oferece, ainda é comum que muitas fontes de dados e possibilidades de análises sejam simplesmente ignoradas. O pensamento de que somente os dados transacionais são suficientes para o processo de tomada de decisão ainda existe, porém essa mudança vem ocorrendo aos poucos.

Velocidade

Além dos desafios impostos pelo volume e variedade dos dados, Big Data também faz referência a outra propriedade: a velocidade com que os dados são coletados, analisados e utilizados.

Imagine, por exemplo, que um cliente receba recomendações de um produto em um e-commerce somente uma semana após ele ter realizado uma compra. Embora tal recomendação ainda possa gerar um efeito positivo, é muito provável que o impacto teria sido superior, caso essa tivesse sido realizada no momento da compra.

Esse é um exemplo que nos mostra que os dados coletados perdem seu valor com o decorrer do tempo. Por exemplo, um relatório atualizado a cada 5 minutos sobre a quantidade de produtos vendidos e em estoque oferece muito mais eficácia no gerenciamento de vendas se comparado a um relatório atualizado semanalmente.

Uma empresa que compreende bem o benefício da velocidade é a varejista Amazon, que adota um mecanismo de precificação dinâmica, podendo chegar a atualizar os valores de seus produtos a cada 10 minutos, de acordo com a análise da demanda em tempo real de seus suprimentos. O fator velocidade está se tornando tão importante, ao ponto que empresas que não conseguirem agilizar o tempo de análise dos dados terão dificuldades em se manterem

competitivas no mercado.

Além da velocidade de análise, o fator velocidade também está relacionado à rapidez com que os dados estão sendo gerados. Estatísticas mencionam que, em apenas 1 minuto, mais de 2 milhões de pesquisas são realizadas no buscador Google, 6 milhões de páginas são visitadas no Facebook e 1.3 milhão de vídeos são vistos no YouTube. Em complemento, temos os inúmeros aplicativos que mantém seus serviços em execução 24 horas por dia e os sensores que geram dados continuamente a cada segundo.

Resumos dos 3 Vs

Diante dos fatos apresentados, cabe ressaltar que, quando nos referimos a Big Data, o importante não é somente a mudança quantitativa dos dados. É possível que uma organização tenha grandes conjuntos de dados e não faça nenhuma análise relevante sobre deles. A grande mudança está no valor que se consegue obter a partir do volume, variedade e velocidade de dados, ou seja, uma mudança qualitativa.

Por exemplo, duas empresas de telecomunicação podem obter milhões de registros de arquivos CDR (*Call Detail Record*). Entretanto, só terá vantagens a empresa que conseguir gerar conhecimento a partir desses dados e utilizá-lo para diferentes aplicações, tais como a segmentação dos assinantes, identificação de fraude e predição de falhas.

A partir dos três atributos mencionados, podemos chegar à seguinte definição de Big Data criada pela consultora Gartner: "*Big Data faz referência não somente ao volume, mas também à variedade e à velocidade de dados, necessitando de estratégias inovadoras e rentáveis para extração de valor dos dados e aumento da percepção*".

Ou seja, precisamos ter consciência que Big Data exige a quebra

de paradigmas. Precisamos lidar com novos tamanhos de dados, novas velocidades, novas tecnologias e novos métodos de análise de dados. Não há como atuar com Big Data estando resistente a mudanças.

Mas então Big Data faz referência somente ao volume, variedade e velocidade dos dados? Não. Além dos 3 Vs, você pode encontrar outros atributos utilizados na definição de Big Data. Alguns pesquisadores adotam os 5 Vs, em que são acrescentados os atributos **valor** e **veracidade** dos dados.

O valor é um atributo que faz referência ao quão valioso e significativo um dado pode ser em uma solução. Por exemplo, qual o valor dos dados de mídias sociais para uma solução de Big Data no departamento de marketing de uma empresa? É importante fazer essa análise de valor para se determinar quais dados serão priorizados pela empresa.

O atributo veracidade está relacionado à confiabilidade dos dados. Pelo fato de que Big Data está inserido em um contexto de dados em grande volume e variedade, é comum a existência de dados inconsistentes. Assim, a veracidade refere-se ao quão confiável é um conjunto de dados usado em uma solução de Big Data.

Além desses atributos, ainda há outros Vs que você pode encontrar em suas pesquisas. Mas não se preocupe, pois os 3 Vs formam a base necessária para o conhecimento de Big Data.

Para dar continuidade ao entendimento sobre Big Data, a seguir serão apresentados exemplos de tipos de dados utilizados nesse contexto. Aproveite a leitura para já tentar identificar quais soluções podem ser criadas na área em que você atua, a partir dessa variedade de dados.

1.3 DADOS GERADOS POR HUMANOS

Na seção anterior, vimos que atualmente os dados são gerados por inúmeras fontes. Podemos classificar os dados em diferentes categorias, tais como dados internos, externos, textuais e transacionais.

Para simplificar nosso entendimento, os dados serão aqui apresentados a partir das seguintes categorias: dados gerados por humanos e dados gerados por máquinas. O conteúdo gerado em cada categoria implica em funcionalidades e características específicas que devem ser consideradas em um projeto.

Dados gerados por humanos são aqueles em que o conteúdo foi gerado a partir do pensamento de uma pessoa, na qual a propriedade intelectual está integrada ao dado. Além disso, podemos entender também como sendo os dados que refletem a interação das pessoas no mundo digital.

Atualmente, grande parcela dos dados gerados por humanos é oriunda de mídias sociais, onde usuários podem publicar o que pensam sobre algo, gerar debates, publicar suas preferências e suas emoções. Essas informações são geradas em formatos de texto, imagem, áudio e vídeo, resultando em uma base de dados diversificada e volumosa.

Se somarmos esses dados aos que são gerados pelos aplicativos de trocas de mensagens, como o WhatsApp, Snapchat e os dados de videoconferência por meio de aplicativos como o Skype, já temos um ritmo acelerado da quantidade de dados que nós, humanos, geramos diariamente. Mas esses não são os únicos que nós geramos.

Além das mídias sociais, sempre que estamos conectados à internet geramos diversos outros tipos de dados. Temos, por exemplo, os blogs, com conteúdo gerado e compartilhado por

milhões de pessoas. Temos ainda as avaliações sobre produtos e serviços que geramos em sites de e-commerce, como a Americanas.com e Amazon.com, e os serviços de *crowdsourcing* como o TripAdvisor.

Essas informações são usadas para gerar recomendações aos usuários, para avaliar o nível de satisfação com um determinado serviço ou produto, e para segmentar os usuários de acordo com os perfis analisados. Dessa forma, dependendo da análise realizada sobre tais dados, a varejista Walmart pode, por exemplo, descobrir quais são as preferências de seus clientes, e a empresa de *streaming* de vídeos Netflix pode descobrir quais filmes recomendar para seus usuários.

Além dos dados já citados, não podemos esquecer daqueles que geramos para documentar algo. Documentos de texto, e-mails, apresentações de slides e planilhas eletrônicas são geradas diariamente para documentar alguma informação, tanto pessoal quanto referente aos negócios de uma empresa. Entretanto, pouco ainda se faz para extrair valor a partir desses dados.

Por exemplo, na sua empresa é feito algum tipo de análise sobre esse conjunto de dados? São poucas as que gerenciam essas informações, possibilitando a descoberta de padrões e melhoria dos processos.

Os sites colaborativos também representam uma parcela significativa de dados gerados por humanos. Dois exemplos notórios são o Wikipédia, a maior enciclopédia online com conteúdo gerido por usuários; e o Flickr, um serviço online de compartilhamento de imagens. Porém, você pode encontrar dados gerados por humanos em inúmeros sites com propostas similares a esses exemplos.

Todos os exemplos aqui mencionados são considerados dados

explicitamente gerados por humanos, em que o usuário possui o conhecimento de quando e como eles são criados. Entretanto, muitos serviços atualmente capturam dados de nós, humanos, implicitamente, ou seja, sem que saibamos que eles estão sendo capturados.

Temos, por exemplo, a relação das URLs que visitamos, os tamanhos de tela dos dispositivos que utilizamos, a descrição desses dispositivos, nossa localização, entre outras informações. Ou seja, são dados oriundos de eventos realizados por nós, porém geradas automaticamente por máquinas, conforme veremos na sequência.

1.4 DADOS GERADOS POR MÁQUINAS

Enquanto dados gerados por humanos são aqueles oriundos do pensamento de uma pessoa, podemos definir dados gerados por máquinas como dados digitais produzidos por processos de computadores, aplicações e outros mecanismos, sem necessitar explicitamente de intervenção humana.

Quando utilizamos uma aplicação Web para fazer o upload de uma foto ou vídeo, para publicar um comentário, jogar ou assistir um vídeo via *streaming*, não temos muita percepção da infraestrutura necessária para suportar tais serviços. Quantos servidores são necessários para armazenar todos os dados que geramos nessas ações? É difícil obtermos essa informação exata.

Entretanto, dado o conhecimento da quantidade de dados gerados diariamente e a imensa quantidade de aplicações Web disponíveis, podemos facilmente concluir que são necessários milhares de servidores em todo o mundo para suportar essa demanda. Além de servidores, a infraestrutura de um data center é formada por diversos equipamentos, como cabos, roteadores e switches.

Para monitorar o status desses componentes, são gerados registros de log sempre que um evento ocorre. Uma vez que tais data centers ficam em execução 24 horas por dia, 7 dias na semana, milhares de registros são gerados ao final de um curto período. Apesar da grande quantidade, é importante manter esses dados, pois a partir deles pode ser possível obter informações úteis aos provedores de serviços.

Por exemplo, arquivos de log podem conter as URLs visitadas por um usuário de um e-commerce, que se forem analisadas, podem prover informações sobre quais compras não foram concluídas e possíveis motivos por isso ter ocorrido. De outra forma, os arquivos de log também podem ser úteis para descobrir a causa de problemas ocorridos e identificar padrões que permitam prever a existência de ocorrências similares no futuro.

Além dos já mencionados, os dados gerados por máquinas têm sido amplamente gerados com o advento da tecnologia de comunicação máquina a máquina (*Machine-to-Machine* — M2M). Uma tecnologia integrada ao paradigma de Internet das Coisas (*Internet of Things* — IoT) que permite a comunicação direta entre dispositivos.

Nesse paradigma, além dos computadores, demais objetos passam a fazer parte da infraestrutura global da internet, gerando, consumindo e interagindo com outras pessoas e objetos, no mundo físico e virtual. Temos como exemplo desses objetos as etiquetas RFID, os sensores, os atuadores, os vestíveis e os smartphones.

Há uma projeção feita pela Cisco que o número de objetos inseridos no contexto de IoT será em torno de 50 bilhões até o ano de 2020. Dada essa quantidade, um relatório da International Data Corporation (IDC) prevê que, em 2020, os dados gerados por máquinas representarão 42% de todos os dados existentes.

Embora os dados usados no contexto de IoT sejam valiosos, o processo de abstração, contextualização, análise e gerenciamento desses dados ainda é considerado um grande desafio. Por esse motivo, além de armazenar os dados gerados, é importante armazenar o seu significado, como informações sobre o tempo e espaço em que eles foram produzidos. A fusão dos dados gerados por diferentes objetos também é necessária para aferir novos conhecimentos, tornando assim o ambiente mais inteligente.

Outros dados fabricados por máquinas e muito usados atualmente no universo de Big Data são os dados genéticos. Temos, por exemplo, a bioinformática, uma área multidisciplinar que tem como foco o estudo da aplicação de técnicas computacionais e matemáticas à (bio)informação, na qual pesquisadores manipulam grandes volumes de dados genéticos para descobrir padrões ocultos sobre eles.

O surgimento de tecnologias de Big Data e o baixo custo nos últimos anos para realizar o sequenciamento de DNA resultou em significantes avanços de pesquisa nessa área. Isso possibilitou a realização de análises que até então eram inviáveis.

Para mensurarmos o volume de dados existente nesse contexto, podemos tomar como referência nosso DNA. Uma sequência de pares do DNA humano possui 3.2 bilhões de pares de base ACTG (Adenina, Citosina, Timina e Guanina). Isso apenas de um ser.

Entretanto, já existem projetos como o *The 1000 Genomes Project* (http://www.1000genomes.org/), que por meio do sequenciamento em larga escala, agregaram centenas de genomas humanos em 2012, resultando em um extenso catálogo de variação genética. Demais estudos já permitem obter sequências de genomas de milhões de espécies de plantas e animais, fornecendo percepções para estudos na área biológica, ambiental, de energia e agricultura.

Os exemplos apresentados formam apenas uma parcela dos dados que podemos conceituar como dados gerados por máquinas. A cada momento, milhões de diferentes tipos de dados digitais são gerados por recursos tecnológicos, resultando em uma fonte quase inesgotável de informação. Somados aos dados gerados por humanos, podemos perceber quantas possibilidades nos são apresentadas no universo de Big Data.

1.5 MITOS SOBRE BIG DATA

Por se tratar de um conceito ainda recente, ainda há muitas dúvidas sobre o que é verdade e o que é mito sobre Big Data. Por esse motivo, antes de darmos continuidade ao conteúdo, confira a seguir algumas informações que você já pode ter escutado em algum momento, mas que **não** retratam a realidade.

- *Big Data engloba somente dados não estruturados.* — Com o crescente volume de dados nos últimos anos, o banco de dados relacional precisou ser complementado com outras estruturas, devido principalmente à escalabilidade e flexibilidade de armazenamento. Entretanto, os dados relacionais continuam sendo valiosos e são muito utilizados em soluções de Big Data. O que mudou de fato foi a inclusão de mais tipos de dados, além dos estruturados.

- *Big Data refere-se somente a soluções com petabytes de dados.* — Embora o volume de dados seja o fator que impulsionou o fenômeno Big Data, aplicações que utilizam conjuntos de dados em uma escala menor do que petabytes também podem se beneficiar das tecnologias de Big Data. Afinal, o mais importante nessas aplicações é a capacidade de extrair valor dos dados.

- *Big Data é aplicado somente às empresas do Vale do Silício.* — Quando se fala sobre Big Data, é comum que sejam usados como exemplos as grandes empresas de serviços Web do Vale do Silício, tais como o Facebook, Twitter e Netflix. Embora elas tenham sido as primeiras a serem desafiadas com o grande volume, variedade e velocidade de dados, atualmente empresas de diversos outros domínios, como agricultura e varejo, também necessitam de tecnologias de Big Data para atender suas necessidades em relação aos dados que elas adquirem.

- *Big Data é aplicado somente em grandes empresas.* — Ainda há essa percepção de que Big Data oferece valor somente para grandes organizações. Entretanto, pequenas e médias empresas também podem obter vantagem competitiva por meio de soluções de Big Data, oferecendo uma melhor experiência aos seus clientes, otimizando processos, reduzindo custos ou criando novos produtos e serviços orientados por dados.

- *Big Data requer o uso de dados externos.* — Embora a adoção de dados de diferentes fontes seja uma prática muito adotada em soluções de Big Data, a aquisição de dados externos não é um requisito obrigatório. Na verdade, a sugestão para quem inicia um projeto de Big Data é buscar extrair valor primeiramente dos dados internos, para somente depois ampliar sua jornada utilizando dados de terceiros.

- *As tecnologias de Big Data já estão bem estabelecidas.* — Infelizmente (ou felizmente, se pensarmos nas oportunidades) não. Estamos vivendo um momento de transição de soluções tradicionais para tecnologias de

Big Data. Portanto, se você for atuar em um projeto de Big Data, deve ficar sempre atento ao surgimento de novas versões das tecnologias adotadas, bem como verificar o surgimento de tecnologias complementares presentes no mercado.

1.6 UM MUNDO DE OPORTUNIDADES

Os exemplos apresentados nos mostram a diversidade de dados que existe atualmente. São dados de diferentes formatos, gerados em períodos e locais diferentes e por diferentes agentes. Mas uma vez que esses dados existem, o que podemos fazer com eles? Eis a grande questão.

Por exemplo, o que um registro de log pode fornecer de informação para meu e-commerce? O que posso fazer com os dados coletados de redes de sensores? O que as opiniões das redes sociais podem me fornecer de valioso? São empresas capazes de responder essas questões que estão potencializando seu negócio a partir de Big Data.

Mas será que preciso capturar todos esses dados para obter oportunidades com as tecnologias de Big Data? A resposta é não.

Muitas empresas já possuem quantidades significativas de dados e não as utilizam para obtenção de valor. Isso pode ocorrer por diversos aspectos em relação à manipulação dos dados. Por exemplo, oportunidades podem ser desperdiçadas pelo fato de que:

- *Os dados não estão integrados.* — Eles já são gerados pela empresa, mas por serem armazenados em diferentes sistemas e bases, não fornecem uma visão ampliada da solução de um problema.
- *Os dados demoram para ser analisados.* — Nesse caso,

gasta-se muito tempo no processo de análise dos dados, o que impede a identificação de informações no momento adequado.

- *Os dados não estão categorizados.* — São casos em que os registros dos conjuntos de dados estão armazenados de diferentes maneiras, sem uma padronização dos campos, impedindo a identificação de anomalias e categorias existentes nos dados.

- *Os dados estão obscuros.* — Casos em que só é possível obter informações a partir da análise de outros dados, como a identificação de padrões em *streaming* de vídeos, extração de informações em imagens e dados manuscritos.

- *Os dados não são usados na tomada de decisão.* — São os que poderiam ser utilizados no processo de apoio à tomada de decisão, mas por não serem integrantes dos dados tradicionais da empresa, são descartados desse processo.

- *Os dados não são visualizados com clareza.* — São situações nas quais os dados já são armazenados, porém não são analisados e apresentados de maneira efetiva para gerar percepções sobre eles.

- *Os dados não são medidos.* — Refere-se a casos nos quais não se utilizam as métricas que os dados podem fornecer para a compreensão de um fato, até então, imperceptível.

Perceba que muitas empresas já têm a possibilidade de aperfeiçoar a utilização de seus dados, mas não conseguem por fatores como os descritos anteriormente. Medidas como a adoção de

novas tecnologias ou uma nova forma de organização dos dados podem trazer grandes transformações em relação à utilização de dados para extração de valor.

Um exemplo é o que ocorreu com uma sede da Microsoft que possuía mais de mil funcionários. Com foco em traçar um plano de eficiência energética dentro da sede, a empresa possuía mais de 30 mil sensores gerando dados a todo instante sobre o consumo de energia.

O problema é que esses dados estavam espalhados em diversos sistemas da empresa, impedindo que ela tivesse uma visão ampla do consumo energético. Com a integração dos dados em um sistema único de eficiência energética, a empresa conseguiu identificar, entre outras análises, salas que consumiam energia sem ter a necessidade.

Como resultado, essa integração evitou um gasto de 60 milhões de dólares com investimento em tecnologias de eficiência energética. Perceba que, nesse caso, a empresa já gerava os dados necessários, o problema estava no modo com que eles estavam organizados.

Outro exemplo é o da Pirelli, empresa multinacional do setor de produção de pneus. Essa empresa estava tendo problemas para entregar seus produtos aos milhares de clientes no tempo correto, sem que houvesse atraso nos pedidos.

Um dos fatores que ocasionavam esses atrasos era a demora para a geração dos relatórios, impedindo os analistas de terem uma visão atualizada da situação do estoque e dos pedidos de vendas. A solução nesse caso foi utilizar uma tecnologia de Big Data que agilizasse o processo de geração de relatórios para os analistas.

Com essa medida, utilizando uma tecnologia de processamento

em memória, relatórios que demoravam um dia inteiro para serem atualizados passaram a ser gerados em 10 minutos. Essa otimização fez com que a empresa tivesse menos desperdício em sua produção, entregasse os produtos de acordo com o prazo e otimizasse o processo de entrega.

Perceba que a empresa já analisava os dados obtidos, mas ela aumentou o valor dos dados agilizando o processo de análise. Esses casos denotam como os dados podem gerar diferentes oportunidades, dependendo da maneira com que são usados.

Uma das famosas frases relacionadas a Big Data é: *"Big Data é o novo petróleo"*. Podemos pensar que isso é uma analogia coerente, dado que, assim como o petróleo, o maior valor é obtido após um processo de refinamento, ou seja, após a transição de dados brutos para um determinado produto. Entretanto, diferente do petróleo, os dados não são recursos escassos.

Um mesmo conjunto de dados que utilizamos para uma determinada estratégia pode ser usado em outra, sem perda nesse processo. Temos assim um leque de oportunidades a serem criadas.

Para tornar mais compreensível o entendimento sobre como essas estratégias podem resultar em oportunidades, na tabela adiante é apresentado alguns exemplos de soluções de Big Data, criadas por diferentes áreas de conhecimento. A área de cuidados de saúde é um exemplo notável.

Com o uso da tecnologia móvel, de dispositivos de IoT e da computação em nuvem, surgem soluções inovadoras para melhorar o cuidado de pacientes, tais como o monitoramento em tempo real do paciente e a previsão de demanda para os leitos do hospital. Soluções de Big Data também estão sendo usadas para identificar padrões em bases de dados históricas de doenças, permitindo acelerar e aperfeiçoar o diagnóstico realizado por uma equipe

médica.

Área	Onde Big Data está sendo aplicado
Cuidados da saúde e medicina	Monitoramento de pacientes em tempo real; Análise de dados de redes sociais para descobertas de pandemias; Análise de padrões de doenças; Extração de informação em imagens médicas; Descoberta e desenvolvimento de medicamentos; Análise de dados genéticos.
Serviços financeiros	Análise de risco; Detecção de fraude; Programas de lealdade; Venda cruzada.
Setor público	Digitalização dos dados; Detecção de fraude e ameaças; Vigilância por vídeo; Manutenção preventiva de veículos públicos; Otimização de rotas no transporte público.
Telecomunicação	Análise de registro de chamadas; Alocação de banda em tempo real; Desenvolvimento de novos produtos; Planejamento da rede; Análise de *churn*; Gerenciamento de fraude; Monitoramento de equipamentos.
Varejo	Análise de sentimento; Segmentação de mercado e cliente; Marketing personalizado; Previsão de demanda; Precificação dinâmica.

Os exemplos apresentados demonstram diferentes formas de como Big Data pode ser utilizado. Entretanto, projetar uma solução não é uma tarefa simples, existindo diversos percalços no decorrer de seu desenvolvimento.

Para que se possa chegar à etapa final de um projeto de Big Data, existe um conjunto de etapas que deverão ser executadas. De forma bastante resumida, descrevo uma sequência de passos existentes nesses projetos.

1. O primeiro passo a ser feito (e que muitas vezes ainda é descartado) é identificar quais perguntas se deseja responder com os dados. É nesse momento que deve ser determinado quais informações pretende-se extrair de um conjunto de dados. Essa tarefa não é fácil. Necessita de pessoas com pensamento analítico, capazes de identificar possíveis análises sobre diferentes dados. Quanto mais claras forem as respostas obtidas nessa fase, mais fácil se torna a execução das fases

seguintes.

2. O próximo passo refere-se a captura e armazenamento dos dados. Devemos então identificar quais fontes serão utilizadas e como os dados serão capturados. Para isso, torna-se necessário identificar a solução adequada para armazenar cada tipo de dado. É nessa fase que identificamos a ordem com que os dados serão usados, definimos quais campos serão utilizados e quais informações devem ser tratadas.

3. Estando os dados armazenados, passamos para a fase de processamento e análise. Tecnologias de Big Data são cruciais nessa fase, para oferecer escalabilidade e desempenho para a aplicação. É nessa fase também que se determina qual algoritmo de análise de dados será usado. Inserem-se aqui os mecanismos de aprendizado de máquina, métodos estatísticos, fundamentos matemáticos e mineração de dados.

4. Por fim, Big Data também inclui a etapa de visualização de dados, em que são utilizadas técnicas de criação de gráficos dinâmicos e interativos. Essa etapa pode também ser usada em conjunto com a fase de análise de dados, para facilitar o processo de descoberta dos dados.

Nos demais capítulos, será abordada com maior detalhe cada uma dessas etapas para a implementação de um projeto de Big Data. É importante ressaltar, porém, que os passos descritos são referentes a uma parte das atividades apenas, havendo inúmeras outras ações necessárias para a execução do projeto, como a aquisição de profissionais habilitados, preparação da infraestrutura e análise de custo. Porém, acredito que o escopo abordado no livro fornecerá uma visão significativa sobre projetos de Big Data.

1.7 CONSIDERAÇÕES

Este capítulo teve como objetivo apresentar ao leitor a base inicial do conceito de Big Data. Podemos descobrir que, além do volume, Big Data também faz referência à grande variedade e velocidade de dados (formando os famosos 3 Vs de Big Data).

Podemos encontrar outros "Vs" na literatura, tais como o valor e a veracidade dos dados, porém são o volume, variedade e velocidade que formam a base de Big Data. Ainda sobre a contextualização de Big Data, foi apresentado que tanto dados gerados por humanos quanto por máquinas são significantes e oferecem diferentes percepções quando analisados.

Antes de iniciar a leitura do próximo capítulo, sugiro que você faça uma reflexão sobre como a empresa que você trabalha, ou a área em que atua, pode ser beneficiada a partir de Big Data. Para auxiliar, sugiro que tente responder às seguintes questões:

1. Quais dados estruturados, semiestruturados e não estruturados são gerados pela minha empresa ou na área que atuo?

2. Como os dados gerados por humanos e por máquinas são utilizados?

3. Há problemas no tempo gasto para analisar os dados?

4. Existem dados que poderiam agregar valor à empresa se fossem adquiridos?

Para dar sequência a essas questões, no próximo capítulo daremos início à fase de captura e armazenamento dos dados.

Para saber mais

1. CARTER, Keith B. *Actionable Intelligence:* A Guide to Delivering Business Results with Big Data Fast!. John Wiley &

Sons, 2014.

2. DUMBILL, Edd. *Planning for Big Data.* O'Reilly Media, Inc., 2012.

3. GANDOMI, Amir; HAIDER, Murtaza. Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management,* v. 35, issue 2, abril 2015, p. 137–144.

4. NEEDHAM, Jeffrey. *Disruptive possibilities:* how big data changes everything. O'Reilly Media, Inc., 2013.

5. O'REILLY RADAR TEAM. *Big Data Now:* current Perspectives from O'Reilly Radar. O'Reilly Media, Inc., 2015. Disponível em: http://www.oreilly.com/data/free/big-data-now-2014-edition.csp.

Capítulo 2

CAPTURANDO E ARMAZENANDO OS DADOS

"Quem não sabe o que busca não identifica o que acha." — Immanuel Kant

Após ter identificado o foco do projeto de Big Data e definido as respostas que deseja obter por meio de dados, você pode dar início à identificação de quais dados deverão ser capturados. Esses dados já existem ou ainda precisam ser gerados? São internos ou externos? Em qual formato eles estão?

Essa série de perguntas é necessária para dar início a uma das fases iniciais do projeto: a captura dos dados. Aliado a essa etapa, deve ser traçada uma estratégia para definir como os dados capturados serão armazenados.

O que é necessário para armazenar dados em grande volume, variedade e em alta velocidade? Será que o banco de dados relacional é a melhor opção? Caso não seja, quais são as outras opções? Confira neste capítulo as respostas para essas questões e as características das etapas de captura e armazenamento de dados.

2.1 FORMAS DE OBTENÇÃO DE DADOS

Para facilitar nossa compreensão sobre as etapas de captura e armazenamento de dados, vamos utilizar como exemplo um projeto de Big Data para uma empresa da área de varejo, que aqui chamaremos de Big Compras. Pensando em oferecer uma melhor experiência aos seus milhares de clientes, os executivos da Big Compras desenvolveram um aplicativo com as seguintes funcionalidades:

- Permitir a pesquisa e compra das centenas de produtos da empresa;
- Permitir que o cliente avalie um produto e verifique os comentários de outros clientes;
- Permitir que o cliente compartilhe as informações de produtos e listas de compras nas redes sociais.

Poucos meses após o lançamento, o aplicativo se tornou um sucesso, atingindo a marca de 1 milhão de usuários. Com esse crescimento, a empresa percebeu que o aplicativo já não estava suportando a quantidade de acessos, gerando a insatisfação dos usuários com as ocorrências de queda de serviço e lentidão de processamento.

Além disso, o volume de dados gerado durante as interações dos usuários com o aplicativo foi crescendo exponencialmente. Entretanto, a empresa não sabia ao certo o que fazer com os dados coletados.

Para resolver essas questões, ela contratou uma equipe de profissionais com a missão de aperfeiçoar o desempenho do aplicativo e entender quais dados eram relevantes para a empresa e como eles poderiam ser usados como vantagem competitiva. Como será que essa equipe poderia resolver esses problemas?

Para compreendermos os desafios da Big Compras, nessa seção

identificaremos as estratégias de captura de dados que podem ser adotadas em um projeto de Big Data. O objetivo é apresentar como cada tipo de dado requer uma estratégia diferente para ser utilizado no projeto.

Dados internos

A primeira abordagem da equipe contratada pela Big Compras foi investigar quais eram os dados internos da empresa. Podemos definir dados internos como sendo aqueles dos quais a empresa é dona e possui controle. Ou seja, a equipe estava interessada em descobrir quais dados já eram gerados e controlados pela empresa, antes de buscar soluções que envolviam a aquisição de novas fontes de dados.

Após uma extensa investigação, os membros da equipe chegaram à surpreendente lista de conjuntos de dados:

- **Dados de sistemas de gerenciamento da empresa:** sistemas de gerenciamento de projetos, automação de marketing, sistema CRM (*Customer Relationship Management*), sistema ERP (*Enterprise Resource Planning*), sistema de gerenciamento de conteúdo, dados do departamento de recursos humanos, sistemas de gerenciamento de talentos, procurações, dados da intranet e do portal da empresa.
- **Arquivos:** documentos escaneados, formulários de seguros, correspondências, notas fiscais, arquivos sobre relação da empresa com seus clientes.
- **Documentos gerados por colaboradores:** planilhas em formato XML, relatórios em formato PDF, dados em formato CSV e JSON, e-mails, documentos em formato Word, apresentações em formato PPT, páginas Web

em formato HTML e XML.

- **Sensores:** dados de medidores inteligentes, sensores de carros, câmeras de vigilância, sensores do escritório, maquinários, aparelhos de ar condicionado, caminhões e cargas.
- **Registros de logs:** logs de eventos, dados de servidores, logs de aplicação, logs para auditoria, localização móvel, logs sobre uso de aplicativos móveis e logs da Web.

Uau, quantos dados já são coletados pela Big Compras. Perceba que, somente com dados internos, as empresas em geral já possuem uma diversidade de dados a serem explorados.

Dessa forma, uma recomendação feita às empresas que pretendem iniciar sua jornada em Big Data é identificar formas de organizar, analisar e utilizar seus dados internos para melhoria dos negócios. Além de esses dados serem mais fáceis de serem adquiridos do que os externos, eles podem revelar informações importantes para as decisões da empresa.

Dados de sensores

Vimos que um dos conjuntos de dados internos refere-se a dados oriundos de sensores. Com tais dados, a equipe da Big Compras visionou uma solução na qual um sensor captaria o momento de entrada do cliente em uma loja física da Big Compras. A partir desse evento, ofertas personalizadas seriam enviadas ao aplicativo no smartphone do cliente e um sinal seria emitido para um atendente poder recepcioná-lo.

Essa solução está inserida no contexto de IoT, na qual os objetos possuem capacidade de comunicação com outros objetos e pessoas.

Para viabilizar uma solução similar a essa, torna-se necessário identificar um meio de transmissão de dados entre os sensores e um servidor para prover o armazenamento, uma vez que tais sensores possuem em sua maioria baixo poder de armazenamento.

Sabemos que atualmente há uma variedade de tecnologias capazes de transmitir informações via um meio de conexão sem fio. A tabela a seguir apresenta uma lista dessas possíveis tecnologias, suas características e exemplos de utilização.

Não somente na área de varejo, os dados coletados a partir do uso de sensores e sinais de smartphones podem gerar uma base de dados muito valiosa para ajudar as empresas a terem uma visão mais ampla de seus clientes. Como resultado, essas empresas conseguem oferecer uma melhor experiência aos seus clientes, tanto online quanto offline.

Tecnologia	Característica	Aplicação
Bluetooth	Comunicação econômica usada para transmissão de dados em pouco alcance	Comunicação contínua entre os dispositivos e aplicações
Celular (2G, 3G, 4G)	Serviços de telefonia móvel para a comunicação entre uma ou mais estações móveis	Atividades gerais na internet
NFC	Comunicação por campo de proximidade (*Near Field Communication*) que permite a troca de informações sem fio e de forma segura	Pagamentos e captura de informação de produtos
Wifi	Comunicação que permite a transmissão de dados em alta velocidade em diversas distâncias	Uso intensivo dos dados como streaming, Voip e download
Zigbee	Comunicação entre dispositivos com baixa potência de operação, baixa taxa de transmissão de dados e baixo custo	Aplicações que exigem baixo consumo de energia e baixas taxas de dados

Além do exemplo de aplicação da Big Compras, atualmente muitas aplicações de Big Data já fazem uso de sensores e demais

dispositivos no contexto de IoT. Na área de transporte e logística, por exemplo, já existem frotas equipadas com centenas de sensores, gerando informações a cada segundo sobre o estado dos pneus, gasto de combustível e qualidade da direção.

Isso resulta na captura de informações em tempo real que auxiliam a manutenção, reduz custos e fornece maior segurança das frotas. Em relação aos 3 Vs, na maioria das aplicações que fazem uso de dados da IoT, o maior desafio está no volume e velocidade com que esses dados são gerados.

Dados da Web

Continuando a investigação sobre os dados que poderiam ser usados pela Big Compras, a equipe identificou dados da Web que poderiam ser coletados de fontes externas, com o propósito de verificar quais poderiam ser relevantes no projeto de Big Data. O resultado dessa investigação chegou à seguinte lista:

- **Dados de domínio público:** dados disponibilizados pelo governo, dados sobre o clima, tráfego e regulamentações, dados econômicos, dados do censo, de finanças públicas, legislação, comércio exterior e Wikipédia.
- **Dados de sites de terceiros:** imagens, vídeos, áudios, podcasts, textos de comentários e revisões em sites da Web.
- **Mídias sociais online:** Twitter, LinkedIn, Facebook, Tumblr, SlideShare, YouTube, Google+, Instagram, Flickr, Pinterest, Vimeo, Wordpress, RSS, Yammer, entre outras.

Nesses tipos de dados, não somente o volume e a velocidade,

mas também a variedade de dados disponíveis tornam sua captura, armazenamento e análise um desafio. No caso da Big Compras, foi identificado que, uma vez que os clientes compartilhavam informações de seus produtos nas mídias sociais online, seria importante analisar esses dados para descobrir se a empresa estava sendo bem ou mal avaliada pelos seus serviços.

A partir dessa estratégia, seria possível identificar quais aspectos eram mais comentados e também gerar novas interações com os clientes por diferentes canais. Para isso, a equipe precisou desenvolver uma técnica para obter esses dados. Mas como capturar esses dados?

A principal forma de captura de dados de mídias sociais online é por meio de uma API (do inglês *Application Programming Interface*), que podemos definir como um conjunto de instruções e padrões de programação, para que os usuários tenham acesso aos dados de um aplicativo ou plataforma. As mídias sociais online (como o Facebook, Twitter e YouTube) disponibilizam APIs para que usuários interajam com os dados que circulam dentro de suas redes, seja capturando-os ou inserindo novos.

A lista a seguir apresenta links para o acesso à documentação de APIs de algumas mídias sociais online:

- Facebook — https://developers.facebook.com/
- Flickr — https://www.flickr.com/services/api/
- Instagram — https://www.instagram.com/developer/
- LinkedIn — https://developer.linkedin.com/
- Pinterest — https://developers.pinterest.com/
- Twitter — https://dev.twitter.com/
- YouTube — https://developers.google.com/youtube/

Utilizando a API do Twitter, por exemplo, um desenvolvedor pode fazer requisições ao servidor da rede e obter uma lista de

mensagens postadas que fazem menção a uma determinada palavra. No caso da Big Compras, por exemplo, a equipe poderia utilizar a API do Twitter para capturar o fluxo de mensagens que contém a hashtag *#BigCompras*.

Além do conteúdo dessas mensagens, a equipe poderia coletar informações adicionais sobre os usuários, tais como a quantidade de seguidores, saber quantas vezes a mensagem foi compartilhada (*retweet*), a data e horário da postagem e até mesmo a localização do usuário que fez a postagem.

Seguindo a mesma abordagem, agora com a API do Instagram, a equipe poderia capturar imagens e comentários enviados pelos usuários da rede e, com a API do Facebook, poderia obter as mensagens, informações do perfil, preferências e curtidas dos usuários. Para que essa captura seja possível, torna-se necessário desenvolver uma solução que receba constantemente o fluxo de dados gerados nas redes, uma vez que novas informações chegam a todo instante.

Uma das maneiras que muitas mídias sociais estão fornecendo acesso às suas APIs é por meio do protocolo REST (*REpresentational State Transfer*). Esse protocolo oferece um estilo que facilita a comunicação entre aplicações Web. O Twitter, por exemplo, oferece uma API que permite ao usuário fazer declarações REST e obter o retorno das declarações em formato JSON.

Dados abertos

Até o momento, a equipe da Big Compras priorizou a utilização de inúmeros tipos de dados, que já possibilitariam inúmeras análises a serem realizadas. Porém, a equipe decidiu identificar também uma nova fonte: os dados de domínio público.

O acesso a dados de domínio público tem sido cada vez mais

facilitado a partir do conceito de dados abertos (*open data*). Segundo a Open Knowledge, uma organização sem fins lucrativos que promove o conhecimento livre, são considerados dados abertos aqueles que qualquer pessoa pode livremente usar, reutilizar e redistribuir, sem restrições legais, tecnológicas ou sociais (https://okfn.org/opendata/).

Diferentes áreas de atuação estão se beneficiando do conceito de dados abertos. A partir da adoção desse conceito, cientistas estão conseguindo acelerar o desenvolvimento de pesquisas tendo acesso a bases de dados que, até então, eram difíceis ou onerosas para se obter. Essa iniciativa também tem sido adotada por entidades públicas, com o livre acesso a diferentes tipos de dados públicos, como por exemplo, dados sobre a economia do país, indicadores de exportação e importação, dados sobre a inflação, entre inúmeras outras opções.

O livre acesso a esses dados pode gerar como resultado o aumento de qualidade da informação para a sociedade, uma vez que facilita a compreensão sobre a situação de um contexto público. Embora ainda com necessidade de aperfeiçoamento, o conceito de dados abertos já foi aderido pelo governo brasileiro e entidades públicas. Uma lista dos dados disponíveis pode ser obtida no Portal Brasileiro de Dados Abertos (http://dados.gov.br/). As entidades privadas também estão aderindo a esse conceito, que estão aos poucos criando novas formas de colaboração, a partir da estratégia de inovação aberta.

Com todos esses dados disponíveis (dados internos, de sensores, da Web e abertos), a equipe da Big Compras chegou à conclusão de que precisaria de uma estratégia para armazenar toda essa variedade. Isso deu início a uma nova investigação: os requisitos para o armazenamento desses dados.

2.2 NECESSIDADES DE ARMAZENAMENTO

Você saberia me dizer qual foi a ferramenta de edição de textos mais usada nos últimos 10 anos? E para a geração de planilhas, fórmulas e gráficos? E qual é o site de busca mais utilizado pelos usuários da internet?

Acredito que são perguntas para quais a maioria das pessoas chega a um consenso. Isso ocorre pelo fato de que tais tecnologias dominam o mercado em que atuam e assim se tornam o padrão de uso para seu segmento.

Seguindo essa mesma lógica, eu pergunto: qual solução foi adotada nos últimos 40 anos para armazenamento de dados? Essa também é fácil de responder. Foi o Sistema de Gerenciamento de Bancos de Dados Relacionais (SGBDR).

Por algumas décadas, o banco de dados relacional se tornou um padrão mundial na forma com que os dados são armazenados. Como já é conhecido por todos da área de TI, nesse modelo os dados são armazenados em estruturas de tabelas, que podem estar relacionadas com outras da mesma base de dados. Por isso o nome relacional. Para criar bancos de dados no modelo relacional, surgiram diferentes SGBRDs, tais como Oracle, PostgreSQL e MySQL.

Como já vimos anteriormente, uma das características marcantes de um SGBDR é o suporte a transações ACID (acrônimo de Atomicidade, Consistência, Isolamento e Durabilidade), que oferecem alta integridade aos dados armazenados. Uma outra característica é o uso da *Structured Query Language* (SQL) para operações de criação e manipulação dos dados.

Com o suporte a esses dois importantes recursos, os SGBDRs revolucionaram a área de gerenciamento de dados, oferecendo

garantias de integridade e a possibilidade de gerar consultas complexas dos dados. Isso fez com que eles se tornassem a escolha dos mais diversos segmentos. Porém, muita coisa mudou com a explosão do uso da internet.

Mesmo com todos os recursos existentes nos SGBDRs, o grande volume e a grande variedade de dados gerados nos últimos anos, principalmente por aplicações Web, trouxeram limitações à adoção desse modelo. Empresas que utilizavam um banco relacional para armazenar dados de um e-commerce, por exemplo, começaram a ter problemas de indisponibilidade de serviço, demora para execução de consultas ao banco e necessidade de muita manutenção para manter o banco de dados compatível com as mudanças do negócio. Em geral, os desafios enfrentados estavam relacionados à escalabilidade, disponibilidade e flexibilidade, como veremos a seguir.

Escalabilidade

Em muitas soluções Web, como é o caso da Big Compras, a quantidade de dados pode crescer aceleradamente à medida que novos usuários e funcionalidades são adicionados à solução. Essa solução é considerada escalável se ela for capaz de manter o desempenho desejável mesmo com a adição de nova carga.

Os SGBDRs conseguem garantir esse desempenho com a adição de mais recursos computacionais de infraestrutura (como processador, memória e disco) ao servidor que hospeda o banco de dados. Essa estratégia é conhecida como escalabilidade vertical e foi suficiente para as soluções por muitos anos.

Entretanto, à medida que o volume de dados aumentou consideravelmente, esse modelo de expansão vertical passou a ser inviabilizado, dado que o custo para adquirir servidores capazes de lidar com a quantidade massiva de dados era alto e, em alguns casos,

eles não ofereciam a capacidade e o desempenho necessários.

Alta disponibilidade

Pelo fato do SGBDR prover a garantia de integridade dos dados, esse sistema pode por vezes tornar um serviço indisponível, em situações nas quais uma transação viole uma regra. Essa garantia é muito útil em diversos cenários, como por exemplo, durante uma transferência bancária entre dois usuários.

Entretanto, existem casos nos quais manter o serviço disponível é mais importante do que garantir todas as propriedades ACID. Esse é o cenário do serviço de carrinho de compras da Big Compras, por exemplo. Mesmo havendo uma inconsistência nas informações do pedido do cliente, de forma que ele não liste todos os produtos que o cliente selecionou, é melhor garantir que o serviço continue disponível e o cliente precise atualizar seu pedido do que interromper o serviço, impedindo-o de finalizar sua compra.

Flexibilidade

Quando utilizamos um SGBDR, temos de ter inicialmente planejado toda a modelagem dos dados antes de armazená-los. Dessa forma, deve ser utilizada uma linguagem formal de banco de dados relacional para definir a estrutura das tabelas, suas colunas, tipos de dados, chaves primárias, chaves secundárias, entre outras características. Isso é o que chamamos de um esquema.

O problema desse requisito é que, em muitas soluções atuais, torna-se inviável o conhecimento antecipado da estrutura dos dados diante da característica não estruturada que eles possuem. Imagine, por exemplo, como a equipe da Big Compras faria se tivesse de definir um esquema para cada um dos campos que ela captura por meio das APIs de inúmeras redes sociais. Isso poderia se tornar inviável, visto que cada registro obtido pode ter uma quantidade

diferente de informações, e a alteração ou adição de novos campos podem ser necessárias para se adequar às mudanças das APIs.

Percebemos, com esses poucos exemplos, que os SGBDRs deixaram de ser a solução ideal para aplicações e serviços que necessitavam de escalabilidade, alta disponibilidade e flexibilidade para gerenciar os dados. Para suprir esses requisitos, novas alternativas foram desenvolvidas, nascendo assim o termo NoSQL.

2.3 TECNOLOGIA NOSQL

NoSQL é uma abreviação de *Not only SQL*, ou seja "não somente SQL". Esse termo foi cunhado para definir os novos modelos de armazenamento de dados, criados para atenderem às necessidades de flexibilidade, disponibilidade, escalabilidade e desempenho das aplicações inseridas no contexto de Big Data.

Diferente do banco de dados relacional, em que o foco principal é voltado à integridade dos dados, os modelos existentes em NoSQL tendem a sacrificar uma ou mais propriedades ACID, para assim oferecer maior desempenho e escalabilidade às soluções que lidam com grande volume de dados.

Assim como não existe um padrão único para criação de aplicações de Big Data, o termo *one-size-fits-all* também não se enquadra em NoSQL. Ou seja, não existe um modelo de armazenamento único que seja adequado para todos os cenários de aplicações, uma vez que cada solução requer necessidades específicas.

Um e-commerce que precisa de rapidez na pesquisa de seus produtos tem necessidades de manipulação de dados diferentes de uma empresa que precisa recomendar produtos em tempo real para seus clientes. Da mesma forma, uma aplicação que precisa

armazenar dados genéticos para analisá-los tem necessidades diferentes de um game online que captura informações dos jogadores.

Enquanto uma solução pode ter como requisito a gravação de informações em fluxos constantes ao banco, outra pode necessitar de leituras periódicas em sua base. Para que cada uma dessas soluções tivessem recursos capazes de atender seus requisitos, diferentes modelos de armazenamento passaram a ser criados no contexto de NoSQL.

Podemos classificar os modelos existentes em NoSQL de acordo com a estrutura que os dados são armazenados. Atualmente, existem 4 modelos principais: o modelo orientado a chave-valor, orientado a documentos, orientado a colunas e orientado a grafos. Confira a seguir as características e aplicabilidade de cada um deles.

Banco de dados orientado a chave-valor

De todos os modelos existentes em NoSQL, o banco de dados orientado a chave-valor é o que possui a estrutura mais simples. Como o próprio nome já indica, esse tipo de armazenamento tem como estratégia o armazenamento de dados utilizando chaves como identificadores das informações gravadas em um campo identificado como valor.

A chave é composta normalmente de um campo do tipo String. Já o campo valor pode conter diferentes tipos de dados, sem necessitar de um esquema predefinido, como acontece em bancos de dados relacionais.

Você pode utilizar o banco de dados orientado a chave-valor tanto para persistir os dados em um banco quanto para mantê-los em memória e assim agilizar o acesso às informações. Nesse segundo caso, é possível recuperar os valores em um banco e

armazená-los em um cache, criando uma chave para cada valor armazenado.

Bancos de dados orientados a chave-valor são adequados para aplicações que realizam leituras frequentes. Considere, por exemplo, o aplicativo de vendas da Big Compras. Os clientes acessam o catálogo de produtos do aplicativo e selecionam os itens desejados para colocá-los no carrinho de compras.

Nesse momento, a aplicação precisa guardar as informações dos produtos selecionados até o momento em que o cliente finalize sua compra. Na figura a seguir, é apresentado um exemplo da estrutura de armazenamento chave-valor para esse cenário.

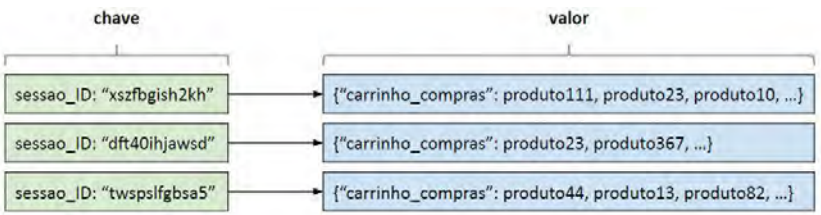

Figura 2.1: Estrutura de um banco de dados orientado a chave-valor

O campo chave usado para fazer a recuperação das informações nesse caso é o ID da sessão de compra do cliente. O campo valor é preenchido com informações sobre os itens inseridos no carrinho de compras.

Perceba como esse modelo possui uma estrutura bem mais simples do que o relacional, não sendo necessária a criação de tabelas, colunas e chaves estrangeiras. O que é necessário apenas é que cada registro tenha uma chave única e que se armazene um conjunto de informações referentes aos valores dessa chave.

Existem atualmente diversas opções de banco de dados orientado a chave-valor. Embora cada um possua suas próprias

características, todas as opções disponíveis são criadas com foco em oferecer flexibilidade, desempenho e escalabilidade no gerenciamento dos dados.

Por esse motivo, esse modelo de banco de dados pode ser uma solução ideal para resolver questões de lentidão para leitura e escrita de dados em grande variedade e volume. Com sua estrutura simples, ele é capaz de otimizar o desempenho da consulta e realizar operações com alta vazão.

Embora a estrutura simples do banco de dados orientado a chave-valor ofereça benefícios, ela também possui algumas limitações. Nesse tipo de banco, o conteúdo do campo valor é "opaco", não sendo possível fazer uma indexação com esse campo e uma consulta mais complexa.

É preciso ter em mente que a única forma de realizar consultas no banco é por meio da chave. Voltando à figura, perceba que não é possível nesse modelo realizar uma consulta que filtre as informações de acordo com os itens do carrinho de compras, por exemplo. Ou seja, em casos nos quais é necessário criar consultas mais elaboradas no banco de dados, esse modelo não é a melhor opção.

Porém, mesmo com essa limitação, bancos de dados orientados a chave-valor podem ser adequados para diversos cenários, como o armazenamento de imagens e de documentos, criação de cache de objetos, armazenamento de dados de sessões do usuário e rastreamento de atributos transientes, como no caso do carrinho de compras.

São exemplos de bancos de dados orientados a chave-valor:

- DynamoDB — https://aws.amazon.com/pt/dynamodb/
- Redis — http://redis.io/

- Riak — http://basho.com/
- Memcached — https://memcached.org/

Banco de dados orientado a documentos

Considerado uma extensão do banco de dados orientado a chave-valor, o banco de dados orientado a documentos é provavelmente a categoria NoSQL mais popular atualmente. Além de também oferecer simplicidade e flexibilidade no gerenciamento dos dados, ele oferece meios de criação de índices sobre os valores dos dados armazenados, enriquecendo as possibilidades de consultas.

Se você já trabalhou com banco de dados relacional, deve estar acostumado com o processo de normalização de dados, executado como uma estratégia para evitar que os dados tenham valores duplicados em um banco. Quando falamos sobre banco de dados orientado a documentos, você pode desconsiderar esse processo. Aliás, você pode desconsiderar também muitos outros conceitos oriundos do banco de dados relacional, tais como a criação de *joins* e definição de esquemas rígidos. Nada disso é necessário nesse modelo.

Podemos definir documentos como sendo estruturas flexíveis que podem ser obtidas por meio de dados semiestruturados, como o formato XML e JSON. Para nos ajudar a compreender melhor sua estrutura, pense em um documento como sendo uma linha de uma tabela, e um conjunto de documentos como sendo a tabela com todos os registros.

A diferença é que cada documento (ou seja, cada linha da tabela) pode conter variações em sua estrutura. Isso é possível pelo fato de que você não precisa definir um esquema antes de adicionar os registros. Veja o exemplo a seguir de um arquivo JSON referente ao armazenamento de informações dos clientes da Big Compras.

Enquanto o usuário `"João"` tem como atributos o campo `fone`, o usuário `"José"` possui o atributo `contato`, contendo uma lista dos atributos (`fonePessoal`, `foneCelular` e `foneComercial` . Ou seja, ambos os registros são referentes aos dados do cliente, porém é possível que cada um tenha diferentes informações.

```
{
    "clientes" :[
      {
        "primeiroNome" : "João",
        "ultimoNome" : "Silva",
        "idade" : 30,
        "email" :"xx@y.com",
        "fone" : "11-984592015"
      },
      {
        "primeiroNome" : "José",
        "ultimoNome" : "Pereira",
        "idade" : 28,
        "email" :"aaa@b.com",
        "contato" {
            "foneFixo" : "11-52356598",
            "foneCelular" : "11-987452154",
            "foneComercial" : "11-30256985"
        }
      }
    ]
}
```

Além dessa flexibilidade, diferente do modelo chave-valor, o banco de dados orientado a documentos permite a criação de consultas e filtros sobre os valores armazenados, e não somente pelo campo chave. Podemos, por exemplo, criar uma consulta que busque todos os clientes com idade superior a 40 anos.

Caso você necessite de uma solução que armazene atributos variados em cada registro, o banco de dados orientado a documentos é uma ótima opção. Além disso, ele oferece grande escalabilidade e velocidade de leitura, pois os dados são armazenados em forma desnormalizada. Por esse motivo, um

documento armazenado deve conter todas as informações relevantes para uma consulta, sem necessitar da criação de *joins*.

Você se lembra da questão de alta disponibilidade? Essa também é uma característica desse banco, que permite trabalhar com a replicação dos dados em um cluster, e assim garantir que o dado ficará disponível mesmo com a ocorrência de falha em um dos servidores.

Esse modelo é indicado para realizar o armazenamento de conteúdo de páginas Web, na catalogação de documentos de uma empresa e no gerenciamento de inventário de um e-commerce, pois são aplicações que trabalham diretamente com coleções de documentos e, portanto, podem se beneficiar da flexibilidade que o armazenamento orientado a documentos oferece.

Além dos cenários apresentados, esse modelo pode também ser muito útil em demais aplicações que utilizem estruturas de dados no formato JSON e que se beneficiam da desnormalização das estruturas dos dados. São exemplos de bancos de dados orientados a documentos:

- Couchbase — http://www.couchbase.com/
- CouchDB — http://couchdb.apache.org/
- MarkLogic — http://www.marklogic.com/
- MongoDB — https://www.mongodb.com/

Banco de dados orientado a colunas

De todos os modelos de armazenamento NoSQL, provavelmente o orientado a colunas seja o mais complexo. Esse modelo também é considerado uma extensão do armazenamento orientado a chave-valor e possui conceitos similares ao do modelo relacional, como a criação de linhas e colunas.

Entretanto, é preciso ficar atento, pois existem diferenças fundamentais no modo como essas estruturas são criadas. Portanto, vamos primeiramente compreender como funciona o armazenamento em um banco de dados relacional.

Para que o armazenamento em um banco de dados relacional ocorra, é necessário definir antecipadamente a estrutura da tabela, indicando suas colunas e tipos de dados. Por exemplo, podemos definir uma tabela simples de nome `cliente` contendo 5 colunas: `id_cliente`, `nome`, `idade`, `email` e `fone`, conforme apresentado na figura a seguir.

CLIENTE	
ID_CLIENTE	INT(10)
NOME	VARCHAR(100)
IDADE	INT(3)
EMAIL	VARCHAR(100)
FONE	VARCHAR(10)

Figura 2.2: Tabela cliente em um banco de dados relacional

Uma vez que definimos essa estrutura, todos os registros de clientes que gravarmos nesse banco deverão conter essas cinco colunas, mesmo que algumas fiquem preenchidas com `NULL`. O SGBDR armazenará e recuperará os dados uma linha por vez, sempre que realizarmos uma consulta.

Essa estrutura de armazenamento pode trazer diversas limitações. Imagine, por exemplo, se esta tabela tem como objetivo armazenar as preferências dos usuários no aplicativo Big Compras. Temos usuários que gravarão apenas os dados obrigatórios, enquanto outros poderão gravar inúmeras outras informações, como preferência de roupas, cosméticos, sapatos e livros.

Imagine ter de reestruturar todos os registros já armazenados na tabela para cada inclusão de um novo campo. E se a quantidade de dados armazenados chegar à escala de terabytes? Mesmo se você realizar uma consulta para buscar um único campo da tabela, o banco de dados relacional precisará passar por todos os registros de todas as linhas para trazer os resultados, impactando o desempenho da consulta.

Mas como o banco de dados orientado a colunas se diferencia do banco de dados relacional nesse cenário? Esse tipo de banco busca resolver principalmente o problema de escalabilidade e flexibilidade no armazenamento de dados.

No que se refere à flexibilidade, ao invés de definir antecipadamente as colunas necessárias para armazenar um registro, o responsável pela modelagem de dados define o que é chamado de "famílias de colunas". As famílias de colunas são organizadas em grupos de itens de dados que são frequentemente usados em conjunto em uma aplicação.

Por exemplo, no cenário anteriormente descrito, poderíamos definir ao menos três famílias de colunas: `dados_cadastrais` , `preferencia_roupas` e `preferencia_livros` . A partir delas, o desenvolvedor possui a flexibilidade de inserir as colunas que considerar necessárias em cada registro armazenado, sem precisar alterar a estrutura dos dados já armazenados.

Conforme a estrutura apresentada na figura a seguir, perceba que o cliente "João" (ID_1) tem informações gravadas nas famílias de colunas `dados_cadastrais` , `preferencia_roupas` e `preferencia_livros` . No entanto, o cliente "José" (ID_2) possui informações gravadas somente nas famílias de colunas `dados_cadastrais` e `preferencia_livros` .

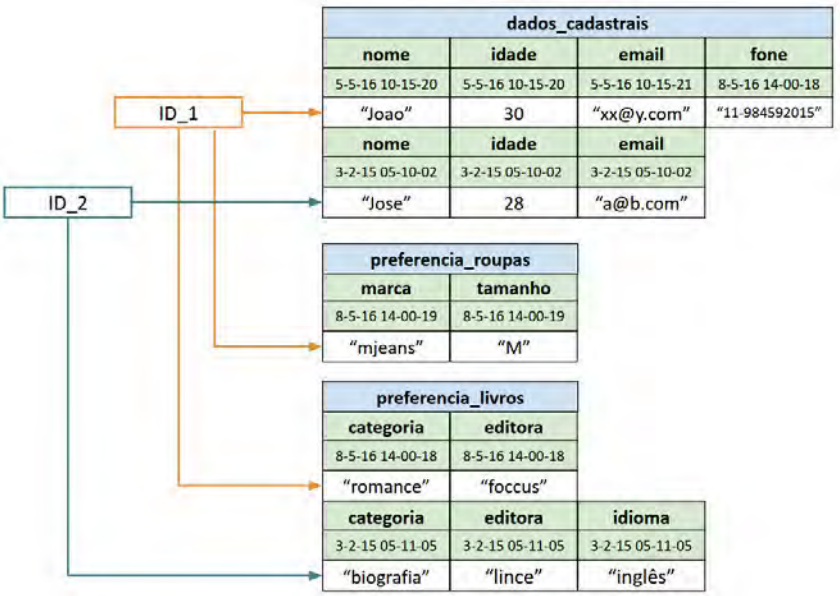

Figura 2.3: Exemplo de família de colunas

Além desse fator, note que o número de colunas pode ser diferente para cada registro. Outra informação armazenada no banco de dados orientado a colunas é o timestamp de cada item gravado. Essa abordagem é utilizada para prover o versionamento das colunas.

Com essa estratégia de armazenamento por famílias de colunas, além de fornecer flexibilidade, esse modelo oferece também grande escalabilidade. O registro de um item pode ter informações gravadas em diversas famílias de colunas, que podem estar armazenadas em diferentes servidores. Isso é possível pelo fato de que os dados são armazenados fisicamente em uma sequência orientada a colunas e não por linhas.

Por exemplo, enquanto no banco de dados relacional o registro seria armazenado na sequência: `"João", 30, xx@y.com, ...`, no banco de dados orientado a colunas a sequência seria: `"João",`

`"José", 30, 28, xx@y.com, a@b.com, ...` . Para esse último cenário, utilizam-se identificadores de linhas e colunas como chave para consultar os dados.

Dessa forma, mesmo em um ambiente distribuído, com milhões de colunas, o banco de dados orientado a colunas pode distribuir as consultas em um grande número de nós de processamento sem realizar operações de *join*.

Se sua aplicação trabalha com volumes imensos de dados, se ela necessita de alto desempenho e de alta disponibilidade na leitura e escrita dos dados, ou se você necessita de inclusão de campos dinâmicos e sua solução tolera eventuais inconsistências, provavelmente o banco de dados orientado a colunas é a solução mais adequada. Por atender tais necessidades, esse modelo é muito utilizado por aplicações de larga escala, como ocorre com o serviço de mensagens do Facebook.

Verifique a seguir exemplos de bancos de dados orientados a colunas. Muitos deles foram inspirados na solução BigTable, introduzida pelo Google (https://cloud.google.com/bigtable/).

- Accumulo — https://accumulo.apache.org/
- Cassandra — http://cassandra.apache.org/
- HBase — https://hbase.apache.org/
- Hypertable — http://www.hypertable.org/

Banco de dados orientado a grafos

Existem casos em que a descoberta de como os dados estão relacionados é mais importante do que os dados em si. Observe o grafo apresentado na próxima figura que ilustra um exemplo dos relacionamentos da rede de usuários do aplicativo Big Compras.

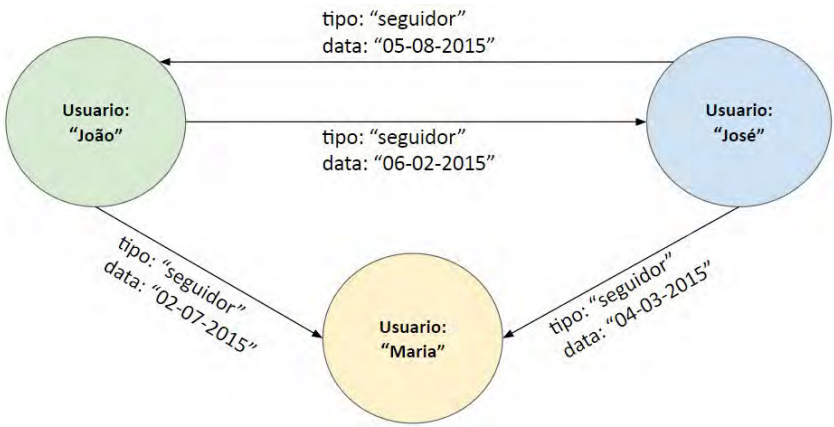

Figura 2.4: Exemplo de banco de dados orientado a grafos

Além das informações armazenadas sobre cada usuário, são também armazenadas informações sobre a ligação entre eles. Podemos identificar no exemplo que o usuário "João" é um seguidor do usuário "José", que também é seu seguidor.

Esse mesmo tipo de informação pode ser usado em toda a rede de usuários, possibilitando a criação de soluções baseada nessa análise, tais como a recomendação de amigos com base na rede de relacionamento. Em situações como essa, com foco no relacionamento dos dados, é que o banco de dados orientado a grafos é recomendado.

Entre os quatro tipos de armazenamento NoSQL apresentados, o orientado a grafos é provavelmente o mais especializado. Diferente dos outros modelos, em vez dos dados serem modelados utilizando um formato de linhas e colunas, eles possuem uma estrutura definida na teoria dos grafos, usando vértices e arestas para armazenar os dados dos itens coletados (como pessoas, cidades, produtos e dispositivos) e os relacionamentos entre esses dados, respectivamente.

Esse modelo oferece maior desempenho nas aplicações que precisam traçar os caminhos existentes nos relacionamentos entre os dados, como por exemplo, as que precisam identificar como um conjunto de amigos está conectado em uma rede, ou descobrir a melhor rota para se chegar a um local em menor tempo.

Um outro modelo de armazenamento, até mesmo o relacional, também é capaz de realizar consultas sobre os relacionamentos entre os itens armazenados. Entretanto, em soluções com milhões de relacionamentos, essa consulta se tornaria muito complexa, resultando em um baixo desempenho.

São exemplos de bancos de dados orientados a grafos:

- AllegroGraph — http://franz.com/agraph/allegrograph/
- ArangoDB — https://www.arangodb.com/
- InfoGrid — http://infogrid.org/trac/
- Neo4J — https://neo4j.com/
- Titan — http://titan.thinkaurelius.com/

Resumo dos modelos de armazenamento NoSQL

Diferentes aplicações necessitam de diferentes tipos de bancos de dados. É exatamente esse fato que impulsionou a criação dos sistemas de gerenciamento de bancos de dados relacionais, e agora, dos bancos de dados NoSQL.

É importante enfatizar que as novas soluções NoSQL não estão sendo construídas para substituir os SGBDRs. Essas são soluções complementares, com características distintas para necessidades não suportadas por um SGBDR.

A tendência é que empresas adotem soluções híbridas, com diferentes modelos de bancos de dados, relacionais e NoSQL, para as diversas necessidades de gerenciamento. Para exemplificar,

confira na figura seguinte uma proposta de armazenamento de dados para o aplicativo Big Compras. Cada serviço pode utilizar um banco de dados específico, para assim garantir um bom funcionamento do aplicativo.

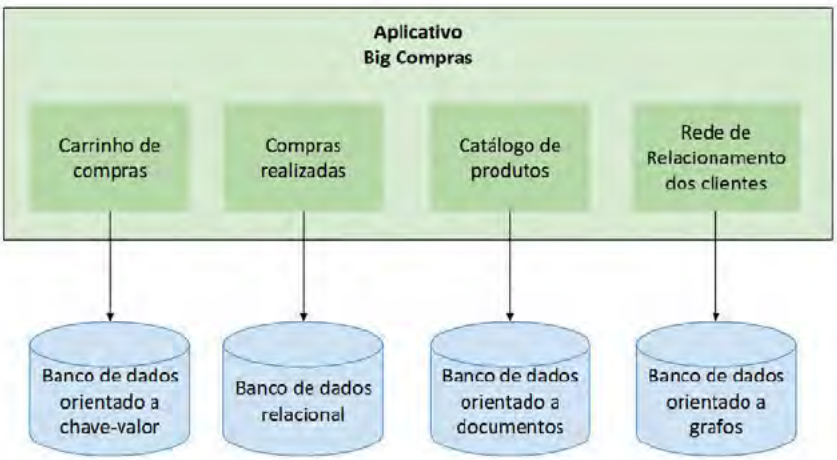

Figura 2.5: Exemplo de solução híbrida de armazenamento de dados

Mas como decidir qual o melhor banco de dados para cada serviço? Isso ainda é um desafio. Entretanto, fazer um estudo de comparação é uma ótima estratégia para garantir que sua solução seja um sucesso.

Embora cada banco de dados NoSQL seja único, com características específicas para atender um determinado requisito de leitura e escrita dos dados, é possível observarmos os seguintes aspectos comuns entre eles:

- **Não relacional:** não seguem as características existentes em um banco de dados relacional, como as garantias da propriedade ACID;
- **Ausência de esquema:** não exigem um esquema rígido e previamente definido como nos bancos de dados

relacionais, oferecendo maior flexibilidade em relação aos tipos de dados armazenados;

- **Projetadas para cluster:** são projetadas desde o início para serem usadas em infraestrutura de cluster, oferecendo maior escalabilidade;
- **Predominância de software livre:** a maioria das soluções existentes em bancos de dados NoSQL seguem a tendência das tecnologias de Big Data, sendo de software livre.

Outro aspecto referente aos bancos de dados NoSQL é o teorema CAP, proposto em 2000 pelo pesquisador Eric Brewer no artigo *Towards robust distributed systems*. O teorema consiste no seguinte conjunto de requisitos para sistemas distribuídos: consistência (*Consistency*), disponibilidade (*Availability*) e tolerância à partição (*Partition tolerance*), conforme apresentado na figura a seguir.

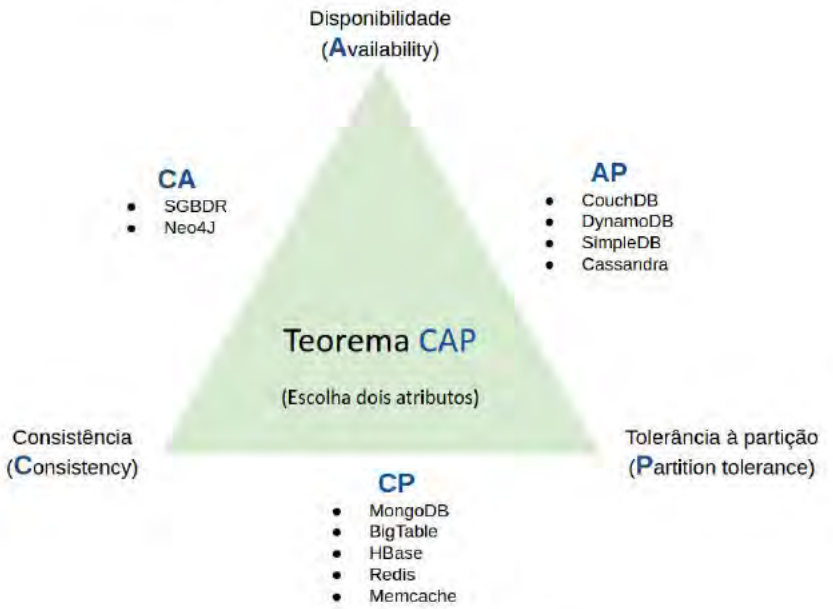

Figura 2.6: Teorema CAP

A consistência refere-se ao aspecto que todos os nós do sistema devem conter os mesmos dados, garantindo que diferentes usuários terão a mesma visão do estado dos dados. Ou seja, é preciso garantir que todos os servidores de um cluster terão cópias consistentes dos dados.

A consistência aqui descrita não tem o mesmo significado que a existente no termo ACID, em que consistência se refere ao fato de que operações que violam alguma regra do banco de dados não serão aceitas. Para o requisito de disponibilidade, o sistema deverá sempre responder a uma requisição, mesmo que não esteja consistente.

Por fim, a tolerância à partição deve garantir que o sistema continuará em operação mesmo que algum servidor do cluster venha a falhar. Ainda segundo Brewer, é teoricamente impossível obter um sistema que atenda os 3 requisitos.

Segundo o teorema CAP, se você precisar garantir consistência e disponibilidade para uma aplicação, você precisará abrir mão da tolerância à partição, pois ele não oferece garantias em relação à alta consistência dos dados se precisar manter a aplicação sempre disponível. Cada um dos exemplos de banco de dados NoSQL citados anteriormente são projetados para atender uma dupla específica dessas características, sendo necessária uma avaliação sobre quais dessas são as mais adequadas para a solução de Big Data projetada.

2.4 A IMPORTÂNCIA DA GOVERNANÇA DOS DADOS

Se o objetivo da empresa em que você atua é, além de criar um projeto isolado de Big Data, criar também uma cultura guiada por dados — na qual eles possuem papel chave para os negócios —, é

crucial que ela tenha uma estratégia eficaz de governança de dados. Sem essa governança, não é possível controlar, gerenciar e monitorar como os dados estão sendo utilizados, nem mensurar o custo e o retorno que eles estão oferecendo.

A implantação de uma estratégia de governança de dados inclui uma série de práticas que deve ser adotada dentro da organização, em todos os departamentos, não somente na equipe de TI, como ainda é comum se pensar. Políticas, padrões, regras, processos, métricas e relatórios são utilizados para comunicar, monitorar e gerenciar os ativos de dados.

A lista a seguir apresenta um resumo dos principais tópicos existentes na governança de dados.

- **Arquitetura dos dados:** é a partir da arquitetura de dados que se define onde os dados ficarão dispostos em toda a organização e como eles poderão ser integrados. O gerenciamento é necessário para identificar as transformações necessárias conforme novas tecnologias são utilizadas e novas soluções são criadas. Nesse sentido, são criadas políticas para padronizar os elementos dos conjuntos de dados, são definidos protocolos e boas práticas para a modelagem de dados, bem como a execução de processos para garantir que os padrões definidos estejam sendo adotados.

- **Auditoria:** uma governança efetiva dos dados deve permitir que profissionais tenham a habilidade de rastrear e compreender quando os dados foram criados, como estão sendo utilizados e o impacto que eles possuem na organização. No contexto de Big Data, esse controle ainda é um desafio, dado que muitas tecnologias e plataformas utilizadas para Big Data ainda não oferecem as funcionalidades necessárias para

soluções de auditoria dos dados.

- **Gerenciamento de metadados:** são esses tipos de dados que servirão de base para as diversas outras áreas de controle na governança de dados, como a segurança e auditoria. Os metadados são importantes para fornecer a contextualização e padronização dos dados. Sejam metadados técnicos, de negócios ou operacionais, é importante que eles sejam gerenciados corretamente para dar o suporte necessário na utilização dos demais dados da empresa.

- **Gerenciamento de dados mestre (*Master Data Management* — MDM):** sabemos que, no contexto de Big Data, os dados não estruturados são coletados e armazenados em seu formato original, tais como os dados de mídias sociais e de sensores. No entanto, as iniciativas de MDM são propostas para criar uma fonte confiável de dados estruturados. Embora ainda seja um desafio, as empresas estão buscando estratégias que utilizem os processos MDM como um papel chave para extrair informações úteis do contexto de Big Data com outros sistemas transacionais da organização.

- **Modelagem dos dados:** a variedade de dados disponíveis e suas diferentes utilizações têm aumentado as formas de modelagem dos dados em uma organização. Um mesmo conjunto de dados pode ser usado em um formato de armazenamento chave-valor, em grafo ou em coluna, por exemplo, necessitando de uma modelagem específica para cada tipo. É importante que se ofereça políticas de modelagem de dados para que se possa estabelecer um padrão entre tantas alternativas disponíveis.

- **Qualidade dos dados:** por mais que quando falamos sobre Big Data, muitos dados coletados podem conter erros ou estar incompletos, o objetivo de uma organização é sempre aperfeiçoar a qualidade e utilidade dos dados. É comum que esses esforços sejam inicialmente aplicados aos dados mestres, porém políticas para criação de *profile* dos dados, bem como estratégias de limpeza, filtragem e agrupamento de dados estão pouco a pouco sendo aplicadas aos demais tipos de dados coletados pela organização.

- **Segurança:** essa prática está relacionada à criação de políticas e ao monitoramento contínuo para uma gestão de risco relacionado a coleta, armazenamento, processamento e análise dos dados. Nesse aspecto, são criadas estratégias de criptografia dos dados, definição e proteção a dados sensíveis, políticas de proteção da integridade, disponibilidade, confiabilidade e autenticidade dos dados. Essas estratégias incluem tanto meios físicos quanto técnicos e administrativos.

Se a governança de dados na era pré-Big Data já era difícil, imagine como essa tarefa se tornou mais desafiadora com a inclusão de dados de inúmeras fontes, grande volume e utilizados para diferentes propósitos? Dado esse desafio e a crescente importância dos dados dentro em uma organização, um novo cargo está sendo criado, principalmente nas grandes organizações: o *Chief Data Officer* (CDO), nome em inglês para o diretor executivo de dados.

A governança de dados é uma das principais responsabilidades do CDO, que deverá também gerenciar e controlar a criação de produtos e serviços guiados por dados em toda a esfera da empresa. Além do conhecimento técnico, esse profissional também deve ter visão de negócios, sendo capaz de criar produtos e serviços a partir

dos dados.

É muito importante que esse profissional consiga conscientizar os colaboradores da empresa sobre a importância de uma governança efetiva, para que eles entendam o porquê precisam seguir determinados processos e padrões. De fato, entre as tantas tarefas atribuídas a esse profissional, a conscientização dessa mudança cultural é provavelmente a mais desafiadora, pois a governança somente será efetiva se todos estiverem dispostos a colaborar.

Embora exista uma estimativa de que somente 25% das grandes organizações possuem um CDO atualmente, a Gartner prevê que esse número será de 90% até 2019. Essa estimativa nos evidencia a tendência de organizações atuarem cada vez mais guiada por dados, e como a monetização de dados será um aspecto essencial para que elas obtenham vantagem competitiva.

2.5 PRATICANDO: ARMAZENANDO TWEETS COM MONGODB

Chegou a hora de colocar em prática um pouco do conhecimento sobre coleta e armazenamento de dados utilizando um banco de dados NoSQL. Para a atividade prática deste capítulo, usaremos o MongoDB, um dos bancos de dados NoSQL orientado a documentos mais populares, para armazenar dados da rede social Twitter. O objetivo será armazenar o conteúdo mais relevante que o Twitter oferece: os tweets, nome dado às mensagens de até 140 caracteres publicadas pelos usuários da rede.

Pense em um banco de dados simples, sem a necessidade de criação de tabelas, esquemas, chaves primárias e chaves estrangeiras, mas que ainda assim permite a criação de consultas complexas sobre os dados. Esse é o MongoDB.

Criado em 2009, MongoDB foi desde o início projetado para ser um banco de dados escalável, de alto desempenho e de fácil acesso aos dados. Os documentos no MongoDB são armazenados em formato BSON, uma representação binária de um documento no formato JSON. Os documentos são agrupados nesse banco de dados em formato de coleções, que podemos pensar como sendo as tabelas de um banco de dados relacional.

Embora não seja adequado para todas as soluções, como por exemplo as que necessitam da garantia ACID, MongoDB é muitas vezes o candidato ideal para soluções de Big Data. Foursquare e Sourceforge são exemplos de soluções que adotaram esse banco.

Essa atividade representa bem um exemplo de como a Big Compras pode compreender melhor seus clientes e assim oferecer uma melhor experiência na utilização do aplicativo. A equipe pode, por exemplo, armazenar todos os tweets relacionados à hashtag *#BigCompras*, para posteriormente analisar esses dados e identificar o que os clientes estão elogiando, sugerindo e/ou reclamando sobre a empresa.

A seguir, é apresentada a descrição do código utilizado nessa atividade. Porém o código também está disponível no repositório git do livro pelo seguinte link:

https://github.com/rosangelapereira/livrobigdata

Para capturarmos os dados que serão armazenados no MongoDB, usaremos a biblioteca open source Twitter4J. Com ela, podemos fazer chamadas à API do Twitter utilizando a linguagem Java. Ela oferece métodos para obtermos autorização de acesso à API, bem como realizarmos operações de captura e inclusão de dados no Twitter.

Em resumo, para realizar essa atividade utilizaremos as

seguintes ferramentas:

- Biblioteca Twitter4J — http://twitter4j.org/
- IDE NetBeans — https://netbeans.org/
- Java — https://www.java.com/pt_BR/
- MongoDB — https://www.mongodb.com/

O objetivo final dessa atividade é que tenhamos um serviço de fluxo de mensagens capaz de coletar tweets e armazená-los em uma coleção no MongoDB. Para alcançarmos esse objetivo, essa atividade é composta de 4 passos principais, sendo eles:

- **Passo 1:** obter credenciais de acesso à API do Twitter;
- **Passo 2:** criar uma classe Java para capturar tweets e armazená-los no MongoDB;
- **Passo 3:** iniciar o serviço de fluxo de mensagens do Twitter;
- **Passo 4:** visualizar dados salvos no MongoDB.

Passo 1: obter credenciais de acesso à API do Twitter

Para acessar a API do Twitter, é necessário que você gere credenciais para ter autorização de acesso. Você deve gerar essas credenciais por meio da página de aplicativos do Twitter (https://apps.twitter.com/). Caso você já possua alguma conta no Twitter, faça o login com essa conta. Caso contrário, crie uma nova conta para realizar o acesso.

Após realizado o login, selecione a opção *Create new app* da página. Como nossa aplicação é somente para teste, você pode preencher somente os campos obrigatórios: `Name`, `Description` e `Website` com informações fictícias. O campo `Callback URL` pode ser deixado em branco, pois não o utilizaremos nessa atividade.

Tendo preenchido os campos necessários, clique no botão *Create your Twitter application*. Após realizar essa operação, acesse a aba *Keys and Access Tokens* e gere um *AccessToken*. Pronto, você agora já tem todas as credenciais necessárias para a aplicação. Copie e cole em algum editor de texto as seguintes informações:

- *Consumer Key (API Key)*
- *Consumer Secret (API Secret)*
- *Access Token*
- *Access Token Secret*

Pronto! Assim completamos a fase de obtenção de credenciais de acesso. Guarde essas informações por enquanto, pois as usaremos em seguida para que nossa aplicação faça a captura dos tweets.

Passo 2: criar uma classe Java para capturar tweets e armazená-los no MongoDB

Neste passo, vamos criar nossa classe com o código necessário para capturar os dados do Twitter e salvá-lo em nossa coleção no MongoDB. Nesta atividade foi utilizada a IDE NetBeans, porém você pode escolher a IDE de sua preferência.

Para dar início à implementação, abra a IDE NetBeans e crie um novo projeto chamado `TwitterApp`, com uma classe chamada `TwitterApp.java`. Para que possamos utilizar as classes do MongoDB e do Twitter4J, devemos importar as seguintes bibliotecas para nosso projeto:

- `mongo-java-driver-2.11.3.jar`
- `twitter4j-core-3.0.4.jar`
- `twitter4j-stream-3.0.4.jar`

Tendo importado as bibliotecas necessárias, abra a classe

`TwitterApp.java`. Aqui será necessário referenciar as seguintes bibliotecas:

```
import com.mongodb.BasicDBObject;
import com.mongodb.DB;
import com.mongodb.DBCollection;
import com.mongodb.Mongo;
import com.mongodb.MongoClient;
import com.mongodb.MongoException;
import java.net.UnknownHostException;
import twitter4j.*;
import twitter4j.conf.ConfigurationBuilder;
```

Na classe `TwitterApp` precisaremos criar objetos das seguintes classes: `ConfigurationBuilder`, `DB` e `DBCollection`. Esses objetos são necessários para conectarmos ao MongoDB e inserirmos informações em uma coleção. Além disso, também serão implementados 4 métodos: `main`, `configuraCredenciais`, `conectaMongo` e `capturaTweets`, descritos na sequência. Confira o esqueleto da classe a seguir.

```
public class TwitterApp {

    private ConfigurationBuilder cb;
    private DB banco;
    private DBCollection collection;

    public void capturaTweets()
            throws InterruptedException {
        //implementação do método
    }

    public void configuraCredenciais(){
        //implementação do método
    }

    public  void conectaMongoDB(){
        //implementação do método
    }

    public static void main(String[] args)
           throws InterruptedException {
        //implementação do método
    }
```

```
}
```

Para o método `capturaTweets`, devemos instanciar um objeto `TwitterStream` e um objeto `StatusListener`. Ambos fazem parte da biblioteca Twitter4J e são utilizados para captura de *streams* do Twitter.

Sempre que o objeto listener capturar um *stream* do Twitter, utilizaremos o método `onStatus` para salvar esse stream em nossa collection no MongoDB. Nesse exemplo, estamos salvando somente os campos `tweet_ID`, `usuario` e `tweet` oferecidos pela API, porém temos a possibilidade de capturar inúmeras outras informações, tais como a localização do tweet e o contador de retweets.

Como queremos capturar somente tweets que possuem a palavra "BigCompras" no corpo da mensagem, precisamos criar um objeto `FilterQuery` e informar no método `track` qual é a palavra que estamos buscando. Esse método recebe como parâmetro um vetor de Strings e, por isso, você pode indicar não somente uma, mas um conjunto de palavras para serem pesquisadas no Twitter.

```
public void capturaTweets() throws InterruptedException {

    TwitterStream twitterStream =
           new TwitterStreamFactory(cb.build()).getInstance();
    StatusListener listener = new StatusListener() {
        @Override
        public void onStatus(Status status) {
            BasicDBObject obj = new BasicDBObject();
            obj.put("tweet_ID", status.getId());
            obj.put("usuario", status.getUser().getScreenName());
            obj.put("tweet", status.getText());

            try {
                collection.insert(obj);
            } catch (Exception e) {
                System.out.println("Erro de conexão: "
                  + e.getMessage());
```

```
            }
        }
    };

    String palavras[] = {"BigCompras"};
    FilterQuery fq = new FilterQuery();
    fq.track(palavras);
    twitterStream.addListener(listener);
    twitterStream.filter(fq);
}
```

O próximo método a ser implementado é o `configuraCredenciais`, no qual devemos inserir as credenciais obtidas no *Passo 1*. Substitua o valor entre parênteses com as suas respectivas credenciais.

```
    public void configuraCredenciais(){
        cb = new ConfigurationBuilder();
        cb.setDebugEnabled(true);
        cb.setOAuthConsumerKey("xxxxxxxxxxxxxxxx");
        cb.setOAuthConsumerSecret("xxxxxxxxxxxxxxxx");
        cb.setOAuthAccessToken("xxxxxxxxxxxxxxxx");
        cb.setOAuthAccessTokenSecret("xxxxxxxxxxxxxxxx");
    }
```

Agora implementaremos o método de conexão com o MongoDB, `conectaMongoDB`. Nesse exemplo a aplicação cliente do MongoDB está sendo executada localmente, mas caso você já tenha o MongoDB instalado em outro endereço, basta substituir o localhost ("127.0.0.1") pelo endereço IP adequado.

```
    public  void conectaMongoDB(){
        try {
            Mongo mongoCli;
            mongoCli = new MongoClient("127.0.0.1");
            banco = mongoCli.getDB("twDB");
            collection = banco.getCollection("tweets");
            BasicDBObject index = new BasicDBObject("tweet_ID",1);
            collection.ensureIndex(index,
                new BasicDBObject("unique", true));
        } catch (UnknownHostException | Exception ex) {
            System.out.println("MongoException :" + ex.getMessage(
));
```

```
    }
}
```

Por fim, no método `main` faremos a chamada dos métodos anteriormente implementados, conforme o código apresentado a seguir.

```
public static void main(String[] args)
        throws InterruptedException {
    TwitterApp tw = new TwitterApp();
    tw.conectaMongoDB();
    tw.configuraCredenciais();
    tw.capturaTweets();
}
```

Após todos os métodos estarem implementados, devemos salvar as alterações realizadas e gerar um `jar` da aplicação. Este conterá as bibliotecas e o código binário necessário para a execução. Para gerá-lo, no NetBeans clique com o botão direito do mouse no nome do projeto e selecione a opção *Construir*. Ao final, deverá ser criado um `jar` com o nome `TwitterApp.jar`.

Passo 3: iniciar o serviço de fluxo de mensagens do Twitter

Tendo criado o `TwitterApp.jar`, acessaremos um terminal para executar a aplicação criada. Para essa operação, estamos partindo do princípio que já existe um serviço do MongoDB em execução no ambiente que vamos executar a aplicação. Você pode verificar como instalar e executar o MongoDB no seguinte site: https://docs.mongodb.com/manual/installation/.

Para executar a aplicação, acesse a pasta onde está localizado o `jar` e execute o seguinte comando no terminal:

```
$ java -jar TwitterApp.jar
```

Esse comando deverá iniciar a aplicação que desenvolvemos.

Caso a conexão com o MongoDB seja estabelecida com sucesso, deverá aparecer no terminal informações similares às apresentadas a seguir:

```
[Mon Jul 25 10:23:23 BRT 2016]Establishing connection.
[Mon Jul 25 10:23:32 BRT 2016]Connection established.
[Mon Jul 25 10:23:32 BRT 2016]Receiving status stream.
```

A partir desse momento são iniciados a captura e o armazenamento de tweets no MongoDB.

Passo 4: visualizar dados salvos no MongoDB

Enquanto a aplicação está capturando os tweets, você pode acessar a coleção no MongoDB e verificar quais tweets já foram armazenados. Para isso, abra um novo terminal e acesse o shell do MongoDB por meio do seguinte comando:

```
$ mongo
MongoDB shell version: 3.0.6
connecting to: test
```

Feito isso, você deverá acessar a coleção que criamos no nosso código. De acordo com os comandos a seguir, primeiramente executamos o comando `use twDB` para acessarmos o banco de dados da nossa aplicação. Para verificar se de fato foi criada a coleção que definimos no código TwitterApp, utilizamos o comando `show collections`.

```
> use twDB
switched to db twDB

> show collections
system.indexes
tweets
```

Agora você pode utilizar os comandos oferecidos pelo MongoDB para consultar os dados armazenados. Como a hashtag *#BigCompras* pode não ter sido citada em nenhum tweet, você

mesmo pode fazer essa postagem no Twitter, para fins de verificação.

No exemplo a seguir, são apresentados dois registros armazenados. Para essa consulta, foram usados o comando `count()` para contar a quantidade de tweets armazenados, e o `find()` para trazer o conteúdo de todos os tweets.

```
> db.tweets.count()
2

> db.tweets.find()
{ "_id" : ObjectId("579612d88f4a37fa6393d541"),
  "tweet_ID" : NumberLong("757566972238278656"),
  "usuario" : "hadoop_girl",
  "tweet" : "Melhor preço só na #BigCompras"
}

{ "_id" : ObjectId("5796130c8f4a37fa6393d548"),
  "tweet_ID" : NumberLong("757567193957691392"),
  "usuario" : "usrtw",
  "tweet" : "#BigCompras é sinal de economia"
}
```

Essas são apenas duas das inúmeras consultas que podem ser realizadas no MongoDB. Você pode, por exemplo, filtrar as mensagens de acordo com um campo, um ID ou uma String, bem como pode ordenar os valores e exportá-los para diversos formatos de arquivo, como CSV e JSON. Aproveite a aplicação criada e explore essas possibilidades!

2.6 CONSIDERAÇÕES

Neste capítulo foram apresentadas as principais questões relacionadas ao processo de captura e armazenamento de dados em um projeto de Big Data. Vimos que atualmente é possível capturar dados de diferentes fontes. Vimos também que, embora o banco de dados relacional tenha sido a solução padrão nos últimos 40 anos, ele se tornou inadequado para suportar grandes volumes e

variedades de dados das aplicações atuais.

A partir da necessidade de um novo modelo de armazenamento de dados, surgiram os bancos de dados NoSQL, oferecendo desempenho e flexibilidade para armazenar diversas estruturas de dados. No capítulo também foi abordada a necessidade da governança de dados para o controle, monitoramento e gerenciamento efetivo dos ativos de dados de uma empresa.

Este capítulo apresentou informações com foco em auxiliar a resposta das seguintes perguntas em um projeto de Big Data:

- Onde estão os dados que preciso coletar?
- De que maneira posso coletar os dados?
- Qual a melhor estrutura para armazenar os dados da minha aplicação?
- Em quais aplicações o uso de um banco de dados NoSQL é mais adequado?
- O que devo controlar e monitorar na utilização dos dados?
- Como posso dar início à governança de dados?

No próximo capítulo, veremos como esses dados podem ser processados por meio de tecnologias de Big Data.

Para saber mais

1. BOAGLIO, Fernando. *MongoDB:* construa novas aplicações com novas tecnologias. São Paulo: Editora Casa do Código, 2015.

2. BREWER, Eric A. *Towards robust distributed systems.* Proceedings of the nineteenth annual ACM symposium on Principles of distributed computing, v. 7, p. 7, 2000.

3. DATE, C. J. *What Is Database Design, Anyway?* O'Reilly

Media, Inc., 2016.

4. MCCREARY, Dan; KELLY, Ann. *Making sense of NoSQL.* Shelter Island: Manning, 2014.

5. PANIZ, David. *NoSQL: como armazenar os dados de uma aplicação moderna*. São Paulo: Editora Casa do Código, 2016.

6. REDMOND, Eric; WILSON, Jim R. *Seven databases in seven weeks:* a guide to modern databases and the NoSQL movement._ Pragmatic Bookshelf, 2012.

7. STEELE, J. *Understanding the Chief Data Officer.* O'Reilly Media, Inc., 2015.

8. SULLIVAN, Dan. *NoSQL for Mere Mortals.* Addison-Wesley Professional, 2015.

9. TIWARI, Shashank. *Professional NoSQL.* John Wiley & Sons, 2011.

10. VAISH, G. *Getting started with NoSQL.* Packt Publishing, 2013.

Capítulo 3

PROCESSANDO OS DADOS

"A Ciência não estuda ferramentas. Ela estuda como nós as utilizamos, e o que descobrimos com elas." — Edsger Dijkstra

Após a fase de captura e armazenamento de dados, podemos iniciar a fase de processamento. Essa etapa também é um desafio em Big Data, visto que precisamos desenvolver algoritmos capazes de processar terabytes ou até petabytes de dados.

Uma vez que esse algoritmo deve ser executado em um ambiente distribuído, além da lógica do problema, precisamos avaliar diversas outras questões relacionadas ao processamento, como alocação de recursos, escalabilidade, disponibilidade, desempenho e tipo de processamento. Pensando nessas questões, neste capítulo são apresentados fatores que devem ser considerados em um projeto de Big Data na etapa de processamento dos dados.

3.1 O DESAFIO DA ESCALABILIDADE

Quando falamos especificamente no processamento de grande volume de dados, um dos maiores desafios em um projeto de Big Data é a escalabilidade da solução. Mas o que isso significa de fato?

Um sistema é considerado escalável se ele permanece com desempenho adequado, mesmo com um aumento significativo do número de usuários, de dados e/ou de recursos. Para garantir essa escalabilidade, portanto, torna-se necessário gerir adequadamente

os recursos computacionais utilizados, bem como monitorar a execução continuamente para identificar quedas de desempenho e criar mecanismos que impeça que a solução esgote algum recurso, levando à sua interrupção. Caso esses controles não sejam devidamente realizados, até mesmo um projeto de Big Data com um ótimo propósito pode ser invalidado.

Imagine se os engenheiros do Facebook não tivessem projetado uma solução que suportasse o acesso de bilhões de usuários à sua plataforma no decorrer dos últimos anos. Imagine se, a cada aumento significativo do número de mensagens recebidas, ou de usuários interagindo na rede social, fosse necessário refatorar todo o código e/ou substituir os recursos computacionais da infraestrutura existente. Certamente a insatisfação dos usuários, o custo e o risco envolvido nessas questões levariam a empresa ao colapso.

Em projetos de Big Data, é crucial um planejamento que permita escalar a plataforma de acordo com o aumento de demanda. Também é necessário que essa plataforma ofereça alta disponibilidade, conseguindo se manter ativa mesmo na ocorrência de falhas, que certamente ocorrerão com o uso de inúmeras máquinas.

Mas como obter um ambiente escalável? Existem duas abordagens: a escalabilidade vertical e a escalabilidade horizontal, conforme veremos a seguir.

Escalabilidade vertical

Para compreendermos o significado de escalabilidade vertical (também conhecido como *scale up*), bem como suas vantagens e desvantagens, imaginemos nossa infraestrutura computacional como sendo o modelo de prédio ilustrado na figura a seguir. Nesse caso, o prédio representa um nó computacional.

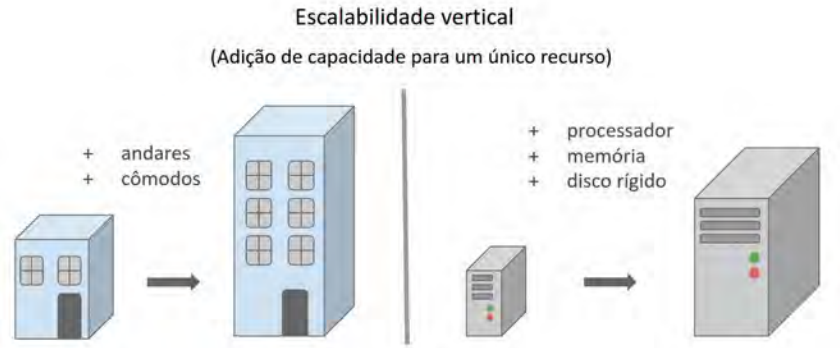

Figura 3.1: Analogia de escalabilidade vertical

Pensando na construção desse prédio, quando for necessário adicionar mais espaço para suportar uma maior demanda de usuários, um novo andar com novos apartamentos é construído. Podemos também substituir alguns materiais utilizados, para que eles ofereçam melhor capacidade para suportar a carga atual. Essa estratégia pode ser usada sempre que houver a necessidade de otimizar a infraestrutura.

Uma das desvantagens desse modelo é que há um limite para a capacidade de expansão, tanto pelo custo, que tende a se tornar cada vez mais caro conforme o aumento da infraestrutura, quanto pelo tamanho, que possui um limite máximo suportado após atingir uma determinada capacidade. Porém, o modelo oferece vantagens significativas para sua aquisição.

A construção de um prédio é uma alternativa atraente pelo fato de que os recursos comuns da infraestrutura podem ser compartilhados entre os usuários. Além disso, o espaço usado para a alocação dos recursos é muito menor quando comparado ao espaço utilizado na construção de casas térreas com a mesma quantidade de cômodos.

Porém, embora o acoplamento dos cômodos traga benefícios,

cada mudança realizada no prédio tende a afetar todos os envolvidos, fazendo com que todos tenham de se adaptar à mudança realizada - o que nem sempre é algo desejado.

De forma similar à do prédio, temos essa estratégia de escalabilidade em um ambiente computacional. Nesse caso, a escalabilidade envolve a adição de processadores, pentes de memória e discos rígidos em um único servidor.

Ou seja, se tenho um processador Intel core i3, por exemplo, posso trocá-lo por um Intel core i7. Se tenho 24 GB de memória RAM disponível, aumento sua capacidade para 32 GB ou para uma capacidade superior, até conseguir o desempenho adequado. O problema nesse cenário é que a substituição ou adição de tais recursos não é transparente, sendo necessária a interrupção dos serviços durante a implantação, podendo causar quedas de serviços temporariamente.

Por outro lado, a escalabilidade vertical permite que os serviços em execução sejam otimizados com a adição de recursos, sem requerer mudanças internas no código das tecnologias utilizadas na manipulação dos dados. Por esse motivo, essa estratégia é frequentemente utilizada, uma vez que as aplicações desenvolvidas são projetadas para escalar dessa forma.

Entretanto, com o surgimento de Big Data, a escalabilidade vertical não foi capaz de suportar a grande demanda de processamento e recursos computacionais impostas pelos dados. Por mais que recursos fossem inseridos a um único servidor, o desempenho continuava insuficiente. Como resultado, muitas aplicações precisaram se adaptar à escalabilidade horizontal para se manterem adequadas às necessidades de Big Data, conforme veremos a seguir.

Escalabilidade horizontal

Seguindo a mesma analogia da escalabilidade vertical, pensemos na escalabilidade horizontal (também conhecido como *scale out*) como sendo um condomínio de casas, conforme apresentado na figura a seguir. A primeira mudança notável nesse tipo de escalabilidade é a necessidade de um maior espaço físico para alocar os recursos, quando comparado com a escalabilidade vertical.

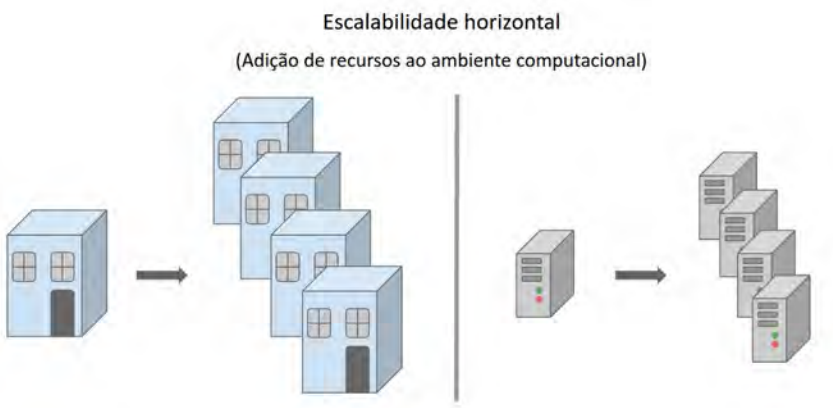

Figura 3.2: Analogia de escalabilidade horizontal

Essa característica pode ser um entrave caso o espaço para alocação dos componentes seja limitado. Esse espaço físico é maior devido ao fato de que a escalabilidade resulta na adição de novos servidores para aumentar o poder computacional, formando assim um cluster.

Outro ponto a se destacar na escalabilidade horizontal é que não há mais o compartilhamento explícito de recursos entre os servidores. Cada um possui um conjunto de recursos independentes. Dessa forma, essa característica evita que os serviços fiquem indisponíveis durante a adição de recursos.

Ainda sobre a escalabilidade de uma aplicação, um cluster computacional tem como objetivo dividir uma carga de trabalho de uma aplicação em um conjunto de tarefas menores, para executá-la

de forma distribuída. Caso seja detectada uma queda de desempenho, ou o aumento do número de usuários ou dos recursos da aplicação, é possível adicionar servidores ao cluster e redistribuir a carga de trabalho até que se alcance o desempenho desejado.

A escalabilidade horizontal oferece inúmeras vantagens à execução de aplicações de Big Data, tais como: permitir que o desempenho da aplicação seja aperfeiçoado de acordo com a demanda, redução de custos para fazer um *upgrade* da infraestrutura, quando comparado à escalabilidade vertical, e por fim, oferece a possibilidade de escalabilidade ilimitada, principalmente em ambientes de computação em nuvem.

Mas, se a escalabilidade horizontal oferece tais benefícios, porque ela não é adotada em todas as aplicações que possuem essa necessidade? Isso acontece pelo fato de que as tecnologias tradicionais para processamento dos dados, ainda muito usadas pelas empresas, não foram originalmente projetadas para a escalabilidade horizontal, sendo necessária uma adaptação da aplicação para esse cenário.

A escalabilidade horizontal exige que o software gerencie a distribuição de dados e as complexidades existentes no processamento paralelo. Caso isso não seja realizado de forma eficiente, o desempenho pode não ser aperfeiçoado (ou em alguns casos pode até ser reduzido) com a adição de novas máquinas.

A partir desse aspecto nasceu a necessidade de novas tecnologias capazes de se adaptar à escalabilidade horizontal de forma eficiente e com complexidade reduzida. Essas são as comumente chamadas "tecnologias de Big Data".

3.2 PROCESSAMENTO DE DADOS COM HADOOP

Uma das primeiras tecnologias de Big Data e que até hoje continua sendo amplamente utilizada é o Hadoop, também conhecido como o famoso elefantinho amarelo. Embora seja usado atualmente para uma infinidade de aplicações de Big Data, esse framework foi inicialmente projetado para um propósito específico: uma engine de busca da Web, tal como o serviço do Google, porém open source.

Criado por Doug Cutting e Mike Cafarella, o framework, que antes era parte integrante do projeto Apache Nutch, foi lançado oficialmente em 2006, passando a se chamar Hadoop.

> Apenas a título de curiosidade, o nome Hadoop surgiu do nome que o filho do Doug Cutting deu ao seu elefante de pelúcia amarelo. Atualmente, o filho está com 13 anos e diz em tons de brincadeira que vai processar o pai por direitos autorais.

O framework Hadoop teve como inspiração a publicação de duas soluções proprietárias da Google: o sistema de arquivos distribuído Google File System (GFS) e o modelo de programação distribuída MapReduce. Ambas as soluções eram utilizadas para dar suporte ao armazenamento e processamento do grande volume de dados que a Google manipulava.

Com base na descrição dos artigos, os criadores do Hadoop desenvolveram uma versão open source baseada nessas soluções, nascendo assim o Hadoop Distributed File System (HDFS) e o Hadoop MapReduce, considerados os principais componentes do framework.

Embora tenha sido desenvolvido para um propósito específico,

desde o seu lançamento grandes empresas passaram a usar o Hadoop para suprir necessidades de diversas aplicações de Big Data. O Yahoo! é até hoje um dos principais utilizadores e colaboradores do framework. Mas houve também contribuições significantes de empresas como o Twitter e Facebook, bem como de universidades e comunidades de usuários open source.

Entre as principais características que tornaram o Hadoop tão interessante para aplicações que envolvem o grande volume de dados estão:

- **Baixo custo:** diferente de muitas aplicações de alto desempenho que requerem hardware específico para o processamento, Hadoop foi desde o início projetado para o armazenamento e processamento de dados em servidores tradicionais, reduzindo consideravelmente custos com infraestrutura. Além dessa característica, o baixo custo também está relacionado ao fato de que Hadoop é open source, podendo ser utilizado gratuitamente.
- **Escalabilidade:** Hadoop oferece escalabilidade linear para as aplicações, além de permitir a execução de aplicações em ambientes de cluster com centenas, ou até mesmo milhares de servidores, sem ser necessário a refatoração de código.
- **Tolerância a falhas:** sabemos que, em ambientes com grandes conjuntos de servidores, é comum a ocorrência de falhas nos componentes de hardware. Por esse motivo, Hadoop possui mecanismos em nível de software que garantem a disponibilidade dos dados e a execução de tarefas, mesmo na ocorrência de falhas.
- **Novas análises:** em um estudo realizado em 2014, a

> Forrester Research estimou que as organizações analisavam apenas 12% dos seus dados, enquanto os 88% restantes não eram utilizados na tomada de decisão. A flexibilidade oferecida pelo Hadoop, tanto no armazenamento quanto no processamento de diferentes tipos de dados, somada à capacidade de escalar a solução, permitiram a exploração de novas análises, até então, inviáveis.

Vimos anteriormente que a escalabilidade horizontal apresenta muitas vantagens, porém muitas tecnologias não estão adaptadas para atuarem nesse cenário. Isso ocorre porque, para processar bases de dados que excedem a capacidade de uma única máquina, torna-se necessário a implementação de soluções distribuídas, capazes de dividir uma grande tarefa em outras menores, executando-as paralelamente em um conjunto de nós.

Entretanto, o desenvolvimento dessas soluções é complexo. Além da lógica do problema, deve-se implementar mecanismos relacionados à distribuição dos dados e das tarefas, como alocação de máquinas, escalonamento de tarefas, balanceamento de carga, comunicação entre máquinas, tolerância a falhas, entre outros aspectos.

A implementação incorreta de algum desses mecanismos pode impactar diretamente o desempenho da solução. É exatamente nesse ponto que o Hadoop se destaca.

Conforme apresentado na figura a seguir, Hadoop oferece uma abstração dos mecanismos existentes em ambiente distribuído, permitindo que o desenvolvedor se concentre unicamente na lógica do problema. Por exemplo, caso a equipe do Big Compras precise executar uma aplicação de detecção de fraude utilizando Hadoop, ela pode se concentrar somente na lógica necessária para realizar a

detecção. Todas as outras questões relativas à execução da aplicação são tratadas pelo próprio framework, reduzindo consideravelmente o tempo gasto no desenvolvimento.

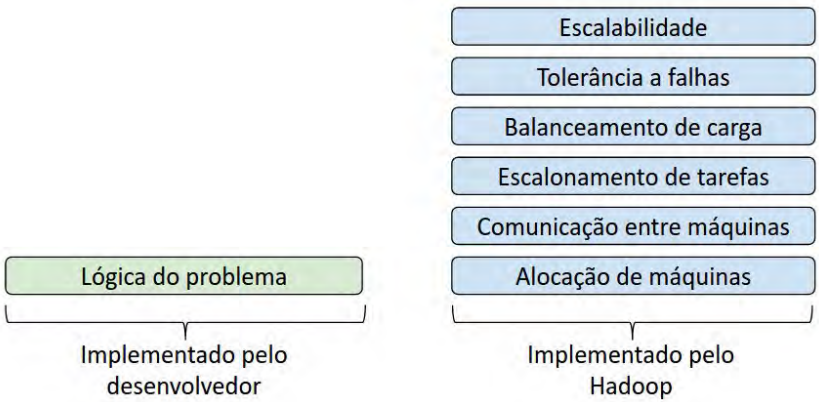

Figura 3.3: Abstração oferecida pelo Hadoop

Nas proximas seções falaremos com mais detalhes sobre o HDFS e o MapReduce, componentes-chave do Hadoop para o armazenamento e processamento de grande volume de dados.

HDFS

Nas atividades relacionadas ao armazenamento dos dados, Hadoop oferece o Hadoop Distributed File System (HDFS), um sistema de arquivos distribuído que permite o armazenamento de grande volume de dados de maneira tolerante a falhas. Por meio desse sistema de arquivos, a distribuição dos dados é feita através dos servidores de um cluster Hadoop.

O HDFS possui uma arquitetura mestre-escravo, na qual um servidor (chamado *NameNode*) é responsável por fazer todo o gerenciamento de metadados do sistema, e um conjunto de servidores (chamados *DataNodes*) são utilizados para fazer o armazenamento dos dados dos usuários. No quesito tolerância a

falhas, o HDFS possui uma estratégia de replicação que garante a recuperação dos dados, mesmo na ocorrência de falhas em um servidor escravo.

Conforme apresentado na figura seguinte, no momento que um usuário submete um arquivo para ser armazenado no HDFS, este é dividido em blocos de tamanhos fixos (128 megabytes por padrão, porém pode ser alterado), que são distribuídos entre os DataNodes do cluster.

Para oferecer tolerância a falhas, para cada bloco são armazenadas outras duas réplicas (valor também configurável) em diferentes servidores do cluster. Essa estratégia garante que, mesmo que um servidor fique indisponível, os blocos que ali estavam podem ser recuperados por meio de suas réplicas em outros servidores.

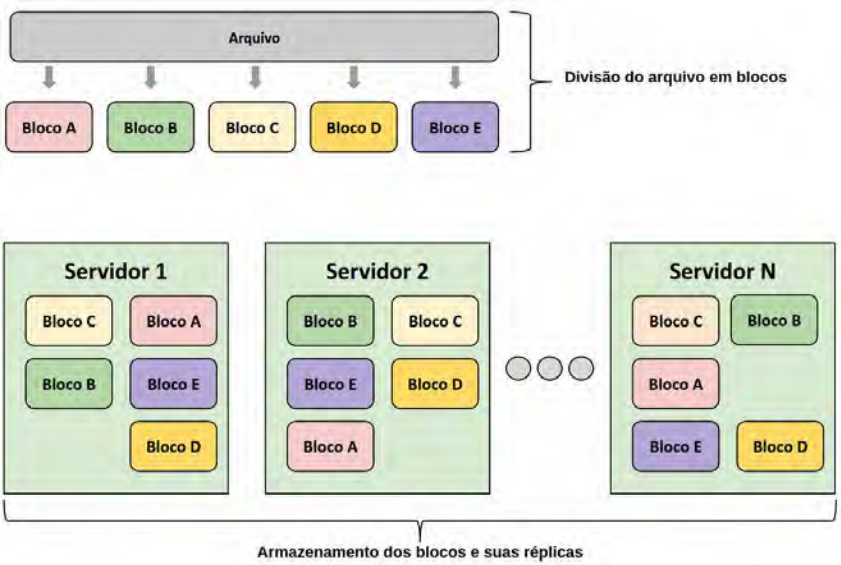

Figura 3.4: Estratégia de armazenamento de dados no HDFS

Outro grande ponto a se destacar no HDFS é a capacidade de

tornar transparentes ao usuário questões complexas de distribuição dos dados. Ou seja, no momento que solicitamos a escrita ou leitura dos dados, não precisamos saber como essas operações serão realizadas internamente, o que torna bem mais simples sua utilização.

Podemos, por exemplo, fazer o envio de um arquivo local para o HDFS usando os comandos a seguir:

```
$ hadoop fs -mkdir MeuDiretorio
$ hadoop fs -put $HOME/MinhaBase.csv MeuDiretorio
```

Nesse exemplo, primeiramente utilizamos o comando `mkdir` da API do HDFS para criarmos um diretório chamado `MeuDiretorio` no HDFS. Na sequência, utilizamos o comando `put` para fazermos uma cópia de um arquivo local no formato CSV para dentro do diretório que acabamos de criar.

Perceba que em nenhum momento foi necessário inserir comandos relativos à distribuição dos dados. Tais operações são realizadas internamente pelo HDFS.

MapReduce

Para o processamento dos dados armazenados no HDFS, Hadoop oferece o modelo de programação MapReduce. Como o objetivo do processamento de dados em um cluster é melhorar o desempenho da aplicação por meio do processamento distribuído, esse modelo permite que grandes volumes de dados sejam processados por meio da divisão de uma aplicação em tarefas independentes, executadas em paralelo nos servidores do cluster.

Como o próprio nome indica, o modelo, inspirado em programação funcional, é composto por duas fases principais: map e reduce. A fase map é a primeira a ser executada em uma aplicação MapReduce. O objetivo dessa fase é processar um conjunto de

dados de entrada, que devem ser obtidos no formato de pares chave-valor.

Dessa forma, cada tarefa map processa cada par chave-valor individualmente, gerando como resultado uma saída, também no formato chave-valor. O resultado dessa saída dependerá da lógica do problema implementada pelo desenvolvedor da aplicação.

Antes de iniciar a fase reduce, ocorre um outro processo de ordenação dos dados. Conforme dados de saída são gerados pelas tarefas map, esses são movidos para serem utilizados nas tarefas reduce.

Cada tarefa reduce deverá receber uma lista contendo todos os valores associados a uma tarefa map. Para isso, é executada uma operação que captura todos os pares chave-valor gerados, e envia para uma tarefa reduce uma lista com a chave e todos os valores correspondentes a ela, de forma ordenada.

Após realizar essa operação para todas as chaves geradas na fase map, inicia-se então a fase reduce. Cada tarefa reduce recebe como parâmetro uma chave e sua respectiva lista de valores. Uma operação sobre essa lista de valores é então executada, gerando assim o resultado da aplicação, também no formato chave-valor.

Na figura adiante, é apresentado um exemplo do fluxo de execução de um algoritmo de contagem de palavras no modelo de programação MapReduce. O objetivo do algoritmo é verificar cada palavra existente no arquivo e contar a frequência de cada uma delas.

Para facilitar o entendimento, o conjunto de dados de entrada foi simplificado, contendo um arquivo com as palavras "Cloud", "IoT" e "Java". Lembre-se de que, em um cenário real, essa entrada pode ser um ou mais arquivos com milhões de palavras.

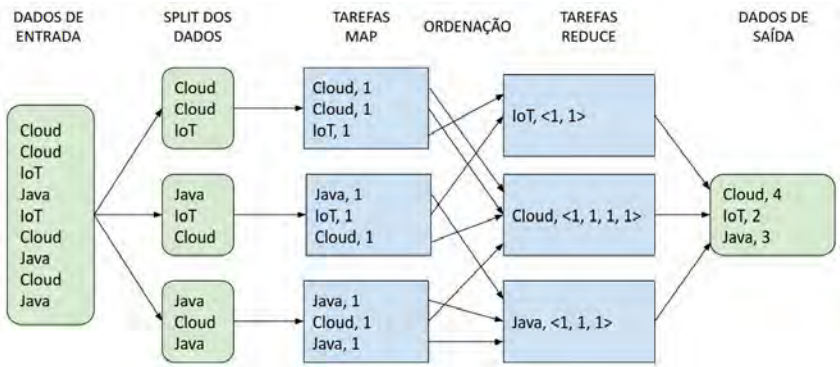

Figura 3.5: Fluxo de execução do algoritmo de contagem de palavras

Para fazer a divisão e distribuição das tarefas, o framework MapReduce realiza um processo chamado split dos dados, que faz uma divisão lógica dos blocos de dados utilizados na aplicação. A quantidade de tarefas map é então determinada pela quantidade de splits.

A partir dessa divisão, um processo do MapReduce inicia a distribuição das tarefas map entre servidores do cluster. O algoritmo da classe map é então executado pelas tarefas, que deverão gerar como resultado um par chave-valor, sendo a chave uma palavra e o valor o número 1.

Perceba que somente com o resultado da fase map não sabemos ainda a quantidade total de ocorrências das palavras, pois elas estão distribuídas entre as tarefas. Para chegarmos a esse valor, no processo de ordenação é gerada uma lista de todos os valores para cada chave. Nesse caso, para cada palavra.

Por fim, as tarefas reduce iteram sobre as listas de valores, gerando como resultado final um arquivo contendo em cada linha uma palavra (chave) e a quantidade de vezes que a palavra foi encontrada no texto (valor). No nosso exemplo, a palavra "Cloud" foi encontrada 4 vezes, a palavra "IoT" 2 vezes e a palavra "Java" 3

vezes.

O algoritmo de contagem de palavras é apenas um exemplo básico de aplicação no formato MapReduce. Esse modelo permite desde a implementação de operações básicas até a construção de soluções complexas, como por exemplo, as que utilizam mecanismos de aprendizado de máquina. O maior desafio está na capacidade do desenvolvedor se adaptar às regras de negócio para serem executadas no estilo map e reduce.

Um dos diferenciais do MapReduce que o tornou adequado para soluções de Big Data foi a estratégia de localidade dos dados. Como a transferência de dados de grande volume resultam em congestionamento da rede e, consequentemente, em baixo desempenho da aplicação, em vez de transferir os dados para onde as tarefas map e reduce deverão ser executadas (como ocorre na maioria das aplicações distribuídas), no MapReduce as tarefas é que são transferidas para onde os dados estão armazenados. Isso reduz drasticamente o consumo de dados transferidos pela rede.

Ecossistema Hadoop

A utilização do Hadoop por grandes organizações contribuiu para sua rápida evolução, tanto em aperfeiçoamento quanto em adição de novas funcionalidades. Como resultado, novos subprojetos foram criados no topo dos componentes principais do Hadoop, criando um ecossistema com diversas soluções de manipulação de dados.

A figura a seguir apresenta uma lista não exaustiva de subprojetos, representando a evolução do ecossistema Hadoop no decorrer dos anos:

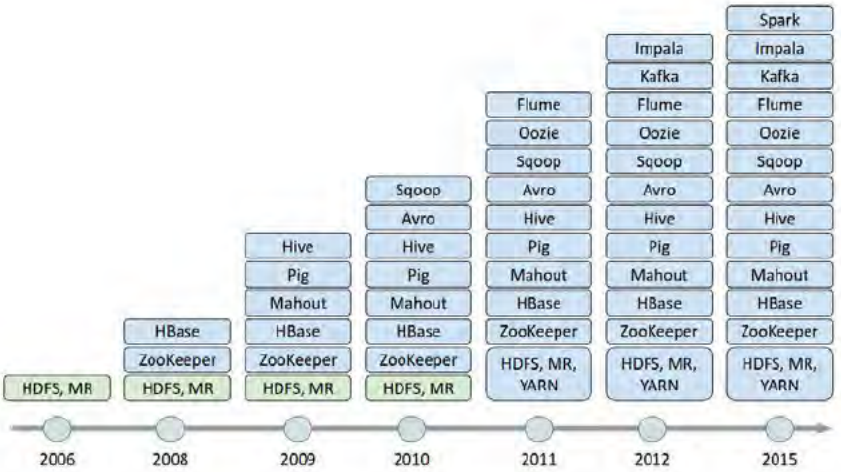

Figura 3.6: Evolução do Hadoop no decorrer do tempo

Diante de tamanha evolução, pesquisadores e desenvolvedores do Hadoop costumam brincar fazendo a seguinte pergunta: "ainda não encontrou uma API que você necessita para trabalhar com o Hadoop? Então espere mais uns cinco minutos que logo ela será desenvolvida". Essa brincadeira faz referência às inúmeras ferramentas criadas a cada ano.

Veja alguns exemplos das funcionalidades que você já pode encontrar do ecossistema Hadoop:

- Se precisar de uma API similar ao SQL para gerar consultar no Hadoop... Apache Hive;
- Se tiver mais facilidade com linguagem de script do que MapReduce... Apache Pig;
- Se precisar monitorar workflows de aplicações em um cluster Hadoop... Apache Oozie;
- Se precisar transferir os dados do HDFS para um banco de dados relacional e vice-versa... Apache Sqoop;
- Se houver a necessidade de executar algoritmos de aprendizado de máquina em modo distribuído...

Apache Mahout;
- Se precisar de uma solução para captura e envio de registros de log para o HDFS assim que estes são gerados... Apache Flume.

Além dessas funcionalidades, a partir da versão 2.0 do Hadoop, novos modelos de programação além do MapReduce puderam ser utilizados na plataforma, tais como processamento em tempo real, iterativo e em memória. Isso foi possível com a criação do YARN (*Yet Another Resource Negotiator*), que trouxe uma nova forma de gerenciamento de recursos e de tarefas executadas no cluster Hadoop.

Com essa mudança, tornou-se possível uma empresa obter uma infraestrutura Hadoop e, a partir dela, executar diferentes modelos de processamento. Assim, Hadoop tem sido considerado cada vez mais a plataforma ideal para soluções de Big Data.

Processamento em lote

Vimos que o Hadoop MapReduce oferece mecanismos para desenvolver aplicações que processam grande volume de dados, oferecendo escalabilidade e desempenho às soluções. Entretanto, Hadoop MapReduce não é uma solução adequada para todas as aplicações de Big Data.

Isso ocorre pelo fato de que, desde sua criação, Hadoop MapReduce foi projetado para uma categoria específica de processamento: o processamento em lote. Mas o que isso quer dizer exatamente?

O processamento em lote refere-se ao processamento em conjunto de um grupo de dados (lotes). Um grupo é formado por dados coletados em um período de tempo e que foram agregados para serem processados por um job. Essa abordagem é comumente

chamada *um-para-muitos*, pois em uma única requisição é processado um grupo inteiro de dados, e não apenas um único registro.

Temos, por exemplo, o processamento em lote de registros de log de um site coletados nos últimos 6 meses, o processamento de transações diárias de cartões de créditos e de tarifas de ligações telefônicas. Ou seja, o foco não é processar um único ponto de dado, mas sim um conjunto deles.

Conforme ilustrado na figura a seguir, as fases de coleta, armazenamento e processamento dos dados ocorrem em momentos distintos no fluxo de processamento em lote. O processo de captura e armazenamento de dados pode ter sido realizado dias, meses ou anos antes dos dados serem processados.

Isso é possível porque o fluxo de processamento não é contínuo, de forma que o processamento se encerra após processar todo o conjunto de dados definido no início do processamento. Por se tratar de um conjunto histórico de dados, é comum processar lotes de tamanhos massivos, podendo demorar minutos, horas, ou até dias, para o processamento ser completamente concluído.

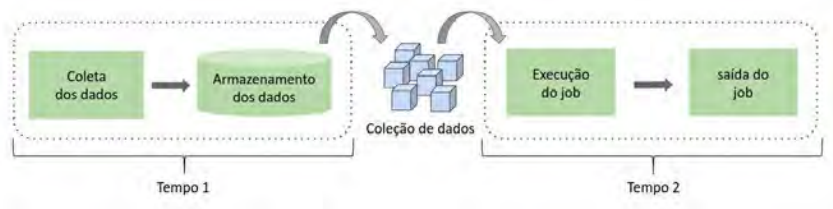

Figura 3.7: Exemplo do fluxo do processamento em lote

Sabemos que, além do volume e variedade dos dados, Big Data está também relacionado à velocidade dos dados. Não somente a velocidade com que os dados são gerados, mas também a rapidez com que os dados são analisados e consumidos.

Ou seja, existem situações nas quais o processamento de dados deve ser realizado no momento em que o dado chega à aplicação.

O grande fator em jogo nesse caso é a latência, ou seja, o tempo que um sistema demora para responder a um evento. Quando falamos de processamento em tempo real, ou próximo ao tempo real, estamos falando de processamento de baixa latência, na ordem de milissegundos ou segundos. Essa é a necessidade de diversas aplicações de Big Data.

3.3 PROCESSAMENTO EM TEMPO REAL

Diferente do processamento em lote, em que as etapas de coleta, armazenamento e processamento ocorrem separadamente, no processamento em tempo real os dados são analisados assim que são gerados, criando a oportunidade de extrair informações imediatas sobre eles. Na figura seguinte, podemos perceber que o fluxo do processamento em tempo real é contínuo, sendo o dado processado à medida que ele chega à aplicação.

Nesse cenário, o processo não é feito sobre um conjunto de dados, mas sobre um item de dado apenas. É o caso, por exemplo, do processamento de dados oriundos de um sensor de temperatura. A cada informação recebida, deve-se imediatamente analisar se o item de dado obtido excede o valor da temperatura considerado aceitável. Esse tipo de processamento é chamado *um-para-um*, pois a requisição do processamento é individual para cada item de dado.

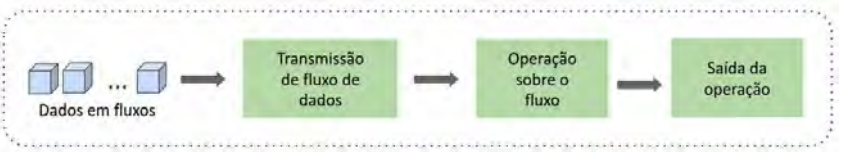

Figura 3.8: Exemplo do fluxo do processamento em tempo real

Como o processamento deve ocorrer rapidamente, realizam-se normalmente somente pequenos cálculos, como uma simples contagem. Para que não haja o gargalo de latência de I/O dos discos rígidos, uma estratégia muito adotada é manter esses dados em memória até que sejam processados, sendo persistidos em um banco somente após essa etapa. Esse tipo de processamento também permite realizar uma análise do dado recém-adquirido com os dados já persistidos no banco, porém, isso pode adicionar uma maior latência.

Características do processamento em tempo real

Um requisito importante no processamento em tempo real é a habilidade de processar os dados seguindo um fluxo, sem precisar armazená-lo para executar uma operação sobre ele. Porém, manter tal processamento em um ambiente que recebe fluxos massivos de dados não é uma tarefa fácil. Para isso, é necessário que a aplicação tenha as seguintes características:

- **Baixa latência:** se o processamento de um item de dado oriundo de um fluxo dura 2 minutos e novos dados chegam nesse fluxo a cada 10 segundos, após um determinado tempo o processamento não ocorrerá mais em tempo real, pois haverá uma latência muito alta para processar cada item de dado recebido. Esse atraso no processamento pode ocorrer tanto por problemas relacionados à projeção da aplicação quanto por ocorrência de falhas que causaram o acúmulo da quantidade de dados na etapa de coleta. Nesse caso, a escalabilidade horizontal do sistema pode ser uma ótima alternativa para manter o desempenho necessário da aplicação.
- **Consistência:** uma solução em tempo real requer a

capacidade para lidar com imperfeições. Uma vez que os dados são processados logo após a coleta, é comum que ocorra alguns problemas, tais como atraso, inversão de sequência e, até mesmo, a perda de alguma parte do dado. Por esse motivo, a aplicação desenvolvida deve ser capaz de manipular inconsistências.

- **Alta disponibilidade:** uma aplicação que realiza o processamento em tempo real pode sofrer grande impacto caso as etapas de coleta, transmissão e processamento dos dados fiquem indisponíveis por um determinado período. Diferente das aplicações que utilizam processamento em lote, o tempo de indisponibilidade no processamento em tempo real pode resultar na perda de dados significantes para a aplicação.

Embora inúmeros fatores possam causar a indisponibilidade dos dados e não seja possível prever todos os problemas, existem mecanismos que buscam amenizar essas ocorrências. A maioria das aplicações adotam recursos de distribuição e replicação, na qual o serviço de processamento é dividido em um conjunto de máquinas, para que, caso uma venha falhar, outra máquina possa realizar o processamento.

Devido a esses requisitos, muitos dos recursos utilizados vão em direção ao uso de sistemas distribuídos. Isso ocorre principalmente no contexto de Big Data, em que o fluxo de dados ocorre em uma escala massiva. Para construir um ambiente computacional adequado para esse tipo de processamento, porém, é necessário verificar as necessidades específicas da aplicação.

Assim como temos o teorema CAP para as tecnologias de

armazenamento de dados NoSQL, os autores Thomas Erl, Wajid Khattak e Paul Buhler apresentam no livro *Big Data Fundamentals: Concepts, Drivers & Techniques* o conceito SVC para sistemas distribuídos de processamento de dados em tempo real. São apresentadas 3 características desses sistemas: velocidade (S — *Speed*), consistência (C — *Consistency*) e volume de dados (V — *Volume*).

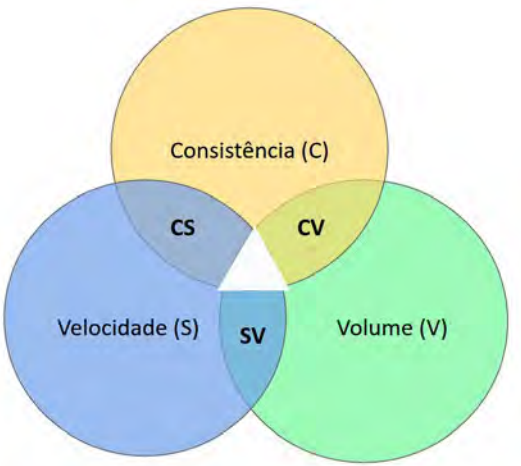

Figura 3.9: Teorema SCV

Conforme ilustrado na figura, assim como no teorema CAP, no teorema SCV somente duas das três características podem ser atendidas em conjunto. Um sistema que requer velocidade e consistência não consegue atender ao requisito volume de dados, visto que não terá tempo hábil para processar em poucos segundos um grande volume de dados.

Entretanto, se o volume de dados a ser processado é grande e a consistência deve ser mantida, então não é possível garantir a velocidade do processamento. Esse teorema pode ser utilizado como um guia para se definir as prioridades e limitações da aplicação a ser desenvolvida.

É comum que os mesmos dados processados em tempo real também sejam posteriormente persistidos em um disco rígido. Assim, permite-se a execução de análises mais complexas sobre eles, agora no modelo de processamento em lote.

Essa abordagem é muito utilizada em soluções que trabalham com o uso de aprendizado de máquina, pois na maioria dos casos essas soluções necessitam de um grande volume de dados histórico para a construção de modelos. Para isso, torna-se necessário uma infraestrutura que permita tanto a baixa latência do processamento das informações quanto tecnologias que ofereçam uma alta vazão para acesso aos dados.

Cenários de utilização do processamento em tempo real

Imagino que você esteja interessado em saber quando o processamento em tempo real é utilizado. A seguir, são apresentados alguns exemplos.

Dados da Web

Vivemos em um mundo cada vez mais digital, onde assistimos a um filme, ouvimos uma música, estudamos, jogamos e compramos pela Web. Por esse motivo, surge cada vez mais a necessidade de empresas que oferecem esses serviços rastrearem os eventos ocorridos nesse contexto.

Em que momento o cliente desistiu de uma compra? Quais produtos ele avaliou antes de decidir qual produto comprar? Qual o número de visitas no site? O serviço está funcionando corretamente?

Para responder perguntas como essas, os provedores desses serviços utilizam técnicas de processamento de registros de log.

Com o advento de Big Data, esse processamento tende a ser cada vez mais automatizado, dada a imensa quantidade de registros a serem analisados.

Essa análise de dados já ocorre há muitos anos, todavia, anteriormente ocorria de forma mais lenta. Porém, diante desse cenário atual cada vez mais competitivo, as empresas estão buscando cada vez mais agilizar o processo de extração de conhecimento dos dados. Por esse motivo, elas estão adotando tecnologias que permitam o processamento em tempo real, para assim agir imediatamente de acordo com o que foi observado.

Detecção de fraude

Com o processamento em tempo real, decisões podem ser tomadas no momento em que uma fraude é identificada ou até mesmo em momentos que a antecede, por meio de modelos preditivos. O processamento em tempo real tem sido então aplicado em inúmeras aplicações para identificação de fraude, tais como em ligações de números de emergência, em transações de cartão de crédito e no mercado de seguros.

Modelos são gerados por meio do rastreamento de inúmeras variáveis, como geolocalização, informações de crédito e perfil em redes sociais, que fornecem insights para identificar uma possível ação fraudulenta assim que um evento é gerado e um novo dado é recebido.

Redes sociais

Sabemos que as redes sociais têm sido uma fonte inesgotável de informações. O processamento em tempo real aplicado aos dados gerados dessas redes sociais tem sido cada vez mais significante. Um exemplo é a identificação atualizada de tendências.

A análise imediata das informações compartilhadas pelas redes

permite gerar descobertas como pandemias de doenças em determinadas regiões. Ou seja, médicos e pacientes podem ter um panorama praticamente imediato da saúde da população em determinadas regiões.

Isso é um grande avanço, visto que em muitos casos se gastava dias para realizar essa verificação, fazendo com que os dados analisados já estivessem desatualizados. Esse é apenas um dos exemplos do benefício do processamento em tempo real de dados oriundos de redes sociais. Os dados gerados por humanos fornecem uma valiosa fonte de informação que, quando avaliados em tempo real, podem gerar conhecimentos imensuráveis.

Internet das Coisas — IoT

No contexto da IoT, temos atualmente milhares de objetos gerando, transmitindo, recebendo e interagindo com os dados. Diferentes tipos de sensores, atuadores, vestíveis e smartphones existentes nesse contexto estão permitindo a inovação em diversas áreas.

Temos como exemplo as casas, o trânsito e as cidades inteligentes. O processamento em tempo real é um dos fatores cruciais para alcançar o potencial oferecido por esses objetos.

Um dos exemplos de uso do processamento em tempo real para IoT é a identificação imediata de acidentes em uma via pública por meio de sensores e câmeras de vigilância. Com essa capacidade de processamento, torna-se possível acionar imediatamente entidades para prestar o socorro necessário. O mesmo pode ser aplicado para a identificação de infrações, desabamentos e assaltos.

Ainda com uso de sensores, um carro conectado pode também identificar um problema interno e emitir um alerta ao motorista sobre a necessidade de manutenção. Além disso, um sensor de

deslizamento pode emitir mensagens aos moradores de locais de risco para que eles saiam de suas casas. Em casos de cuidados de saúde, monitores cardíacos em um paciente podem informar imediatamente um médico sobre a necessidade de socorro. Perceba como o tempo do processamento da informação obtida é crucial para a eficácia dessas aplicações.

Conforme já foi abordado, um fator limitante do processamento de dados inseridos no contexto de IoT ainda é a conectividade. Pois normalmente os dados precisam ser transferidos do dispositivo para o servidor, para somente então serem processados. O tempo gasto durante essa transferência pode impactar diretamente a solução.

Os meios de comunicação têm evoluído rapidamente, estando cada vez mais aperfeiçoados, porém ainda há muita instabilidade, comprometendo a consistência e disponibilidade dessas aplicações. O conceito de *Fog Computing* também tem auxiliado a resolver essa questão, trazendo recursos computacionais para permitir o processamento dos dados de IoT mais próximos dos locais onde os dados são gerados.

Diante dessas possibilidades, como criar uma aplicação que realize o processamento em tempo real? Uma das primeiras decisões a ser tomada é em relação à tecnologia de processamento. Por exemplo, não conseguiremos utilizar o Hadoop MapReduce para esse cenário, visto que ele foi projetado para o processamento em lote dos dados. Precisamos de uma tecnologia que realize o processamento de fluxo de dados, conforme veremos a seguir.

Tecnologias de Big Data para processamento em tempo real

Assim como surgiu o Hadoop e o modelo MapReduce para permitir o processamento em lote de grande volume de dados, surgiram também tecnologias com suporte ao processamento em

tempo real de Big Data. Uma dessas tecnologias que tem se destacado é o Apache Storm.

Desenvolvido originalmente no Twitter, o Storm se enquadra na categoria de tecnologias para o processamento de fluxo de dados, oferecendo baixa latência. Além desses benefícios, uma das características que tornam o Storm atrativo é a garantia de processamento mesmo na ocorrência de falhas, aumentando a consistência da aplicação em tempo real. Esse framework possui o seu próprio gerenciador de recursos do cluster, porém pode também ser utilizado em uma infraestrutura Hadoop, por meio do YARN.

Além do Storm, outro framework que tem se destacado no processamento em tempo real e próximo ao tempo real é o Apache Spark. Considerado uma evolução do Apache MapReduce, esse framework oferece mecanismos que otimizam o processamento em memória dos dados.

Por meio de um cache de resultados intermediários mantidos em memória, ele otimiza os processos que executam diversas vezes sobre o mesmo conjunto de dados. Isso torna-o mais rápido do que o Hadoop MapReduce.

Outra diferença em relação ao modelo MapReduce é a adoção de um modelo conhecido como conjuntos de dados distribuídos e resilientes (*Resilient Distributed Datasets* — RDDs), que são distribuídos em um cluster para serem executados em paralelo. O Spark também é indicado para o processamento em tempo real de fluxos de dados por meio da biblioteca Spark Streaming.

Storm e Spark são apenas dois exemplos de frameworks indicados para o processamento em tempo real para Big Data. Além de tecnologias voltadas à fase de processamento, há também as que oferecem funcionalidades para as etapas de coleta, transmissão,

armazenamento e análise dos dados.

Por exemplo, um framework muito usado para soluções de tempo real é o Apache Kafka. Desenvolvido por engenheiros do LinkedIn, esse framework é considerado um sistema de distribuição de mensagens e oferece mecanismos de gerenciamento da distribuição dos dados, permitindo garantir que os dados coletados serão armazenados em uma fila e continuarão disponíveis mesmo em casos de falhas.

Aqui cabe também citar o framework Apache Samza. Também desenvolvido no LinkedIn, Samza utiliza o Apache Kafka para a distribuição de mensagens e o YARN para o gerenciamento de recursos do cluster.

Conforme a velocidade de processamento se tornou um fator cada vez mais significativo no mundo dos negócios, novas tecnologias têm sido criadas para atender os requisitos específicos das aplicações. A tabela apresentada a seguir apresenta uma lista de sugestões adicionais de tecnologias existentes para esse fim. Perceba que, mesmo havendo a predominância de criação de tecnologias de Big Data open source, ainda existem diversas soluções proprietárias adequadas para esse contexto.

Categoria	Framework
Open source	Apache Flink; Apache Samza; Apache Spark; Apache Storm; S4
Proprietária	Amazon Kinesis; IBM InfoSphere Streams; Data Torrent; Informatica Vibe Data Stream; Microsoft Streaminsight; ParStream's Analytics; SAS Event Stream Processing

Com diversas opções disponíveis, você deve estar se perguntando: qual desses frameworks devo utilizar? A resposta para essa pergunta depende de vários fatores, como a necessidade de integração com outros sistemas da empresa, o conhecimento da

equipe sobre os modelos de programação de cada framework e a política da empresa para atuar com software open source ou proprietário. Uma decisão pode também ser a escolha de mais de um framework, seja atuando em conjunto ou utilizados para propósitos distintos em cada aplicação.

3.4 BIG DATA E COMPUTAÇÃO EM NUVEM

Atualmente existe um conceito chamado *Internet of Everything* (IoE), que podemos traduzir como "a internet de todas as coisas". Esse conceito faz referência às soluções que estão reinventando o modo com que os negócios são operados.

Por exemplo, além de Big Data, temos outras soluções digitais que estão sendo amplamente utilizadas pelas empresas para conduzir os negócios de forma inovadora. A computação em nuvem é certamente uma dessas.

Por meio da oferta de recursos computacionais (como processador, armazenamento e rede) sob demanda, a computação em nuvem tem sido uma grande aliada para a criação de soluções de Big Data. Sendo um paradigma que oferece benefícios como a elasticidade de recursos, a escalabilidade e qualidade de serviço, somada à redução de custo e ao aumento da eficiência, diversas empresas estão aproveitando o potencial da computação em nuvem para hospedar suas soluções de Big Data.

Conforme apresentado na figura seguinte, a computação em nuvem pode ser usada em diversos modelos de entrega de serviços, sendo os mais tradicionais o modelo de Infraestrutura como serviço (*Infrastructure-as-a-Service* — IaaS), Plataforma como serviço (*Platform-as-a-Service* — PaaS) e Software como serviço (*Software-as-a-Service* — SaaS).

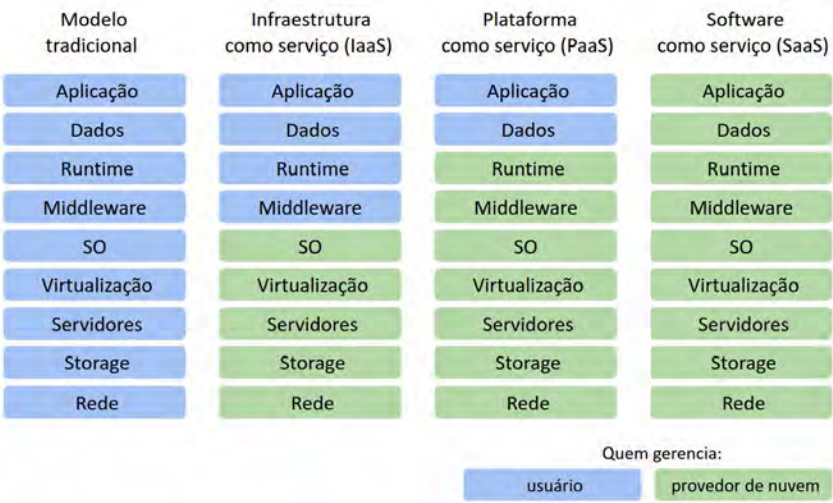

Figura 3.10: Modelos de serviços de computação em nuvem

Uma das principais diferenças em cada modelo ofertado na computação em nuvem é o nível de abstração oferecido ao usuário. Por exemplo, ao utilizar um IaaS, como é o caso do serviço Amazon Web Services, o usuário poderá controlar questões referentes à sua aplicação, como o gerenciamento dos dados e a plataforma de execução da aplicação. Entretanto, ele tem a responsabilidade de construir o ambiente necessário para o processamento dos dados.

O controle sobre os recursos oferecidos já é menor em um modelo PaaS, na qual o usuário deve obrigatoriamente utilizar a *runtime* oferecida pela plataforma, como é o caso da Microsoft Azure e do Google App Engine. No último modelo, o gerenciamento é realizado totalmente pelo provedor da nuvem, como é o caso do Google Gmail e o serviço de CRM SalesForce.com. Ou seja, o usuário não tem controle sobre a execução dos serviços.

Além desses, você também pode encontrar modelos de serviços com outros nomes, tais como *Analytics-as-a-Service* e *Database-as-a-Service*. Tais nomes são adotados comercialmente para especificar

o propósito do serviço oferecido aos usuários finais da nuvem.

A computação em nuvem oferece uma série de vantagens referentes ao armazenamento e processamento de dados. A flexibilidade oferecida pela alocação de recursos sob demanda facilita o gerenciamento da capacidade de armazenamento de uma aplicação.

Dessa forma, uma vez identificada a necessidade de maior ou menor capacidade de armazenamento, esse requisito pode ser imediatamente atendido, oferecendo ao provedor da aplicação maior controle de custos com a infraestrutura. Entretanto, mesmo com pontos positivos na adoção de computação em nuvem para o armazenamento de dados, ainda existem percalços que impactam essa adoção.

Um dos fatores ainda não completamente resolvidos na computação em nuvem é a segurança e privacidade dos dados. Mecanismos de criptografia, autenticação e restrições de acesso estão sendo implementados para evitar o acesso indevido aos dados. Esse é um dos principais entraves para empresas nas quais a segurança dos dados é extremamente crítica, como é o caso das instituições financeiras.

Quando falamos em Big Data, outro grande desafio no armazenamento em computação em nuvem é em relação a transferência de dados. Esses dados precisam ser transferidos por meio de uma conexão com a internet, porém, como é o caso do Brasil, grande parte de pequenas e médias empresas possuem baixas taxas de upload de dados.

Por exemplo, para fazer o upload de um arquivo de 100 GB com uma taxa de upload de 10 Mbps (Megabits por segundo), seria necessário praticamente um dia inteiro para completá-lo, isso desconsiderando a ocorrência de falhas e atrasos. Se considerarmos

1 terabyte de dados, seriam necessários aproximadamente 13 dias para completar o upload, tornando inviável para uma empresa que precisa trafegar centenas de terabytes de dados.

Esse cenário é um desafio para milhares de empresas de todo o mundo. É por esse motivo que até mesmo a Amazon, uma das maiores empresas de serviços de computação em nuvem, oferece como opção o transporte manual dos dados, por meio do envio de um dispositivo portável de armazenamento de dados via correio. É o popularmente chamado "protocolo Kombi". Esse cenário torna evidente como as tecnologias precisam caminhar juntas para garantir o sucesso das aplicações de Big Data que utilizam a internet.

3.5 PRATICANDO: CONTAGEM DE HASHTAGS EM MAPREDUCE

Nessa atividade, veremos como implementar uma aplicação MapReduce e executá-la em um ambiente Hadoop. O foco da aplicação será contar a frequência de cada hashtag encontrada em uma base de dados de mensagens da empresa Big Compras. A base é composta por mensagens de texto oriundas das redes sociais e do e-commerce da empresa.

O objetivo será, a partir das mensagens obtidas, identificar quais hashtags são mais comentadas pelos usuários, fornecendo uma visão inicial da experiência dos clientes. Para que possamos compreender facilmente o funcionamento da aplicação, utilizaremos uma base de dados pequena. Entretanto, o mesmo código que implementaremos pode ser usado para uma imensa base de dados, na escala de gigabytes até petabytes.

A base de dados está disponível no repositório git do livro, na pasta `cap3 >> input >> tweets.txt` . Você pode fazer o

download desse arquivo pelo seguinte link:

https://github.com/rosangelapereira/livrobigdata/tree/master/cap3/

Assim como a base de dados, o código da atividade também está disponível para download no diretório `cap3 >> ContaHashtags`. O arquivo `tweets.txt` contém em cada linha uma mensagem que faz referência à empresa Big Compras. Confira a seguir as 8 primeiras linhas do arquivo:

```
#BigCompras em liquidação. #sucesso #livros #queropresente
O produto é barato mas o frete é muito caro #BigCompras...
Ficando louca com as promoções de perfumes da #BigCompras!...
#BigCompras aceitando pagamento com cartão de débito! #sucesso
#BigCompras é top! Comprei o produto ontem e hoje já ...
Comprei um fone de ouvido na #BigCompras a 3 semanas ...
Só compro tênis da #BigCompras, lá tem #PrecoJusto
Só volto a comprar na #BigCompras quando tiver #PrecoJusto
```

Para executarmos a aplicação MapReduce em um ambiente Hadoop, a atividade será composta de 4 passos principais:

- **Passo 1:** implementação da aplicação MapReduce;
- **Passo 2:** envio da base de dados para o HDFS;
- **Passo 3:** execução da aplicação em ambiente Hadoop;
- **Passo 4:** verificação dos resultados.

Para isso, serão utilizadas as seguintes ferramentas:

- Apache Hadoop — http://hadoop.apache.org/
- IDE NetBeans — https://netbeans.org/
- Java — https://www.java.com/pt_BR/

Passo 1: implementação da aplicação MapReduce

O algoritmo de contagem de hashtags tem como objetivo contar a frequência com que cada hashtag aparece em uma base. Sabemos que, caso a base de dados seja pequena, podemos fazer essa

contagem utilizando ferramentas tradicionais, como o Excel. Entretanto, se a base contiver milhões ou bilhões de linhas, provavelmente o processamento será muito mais eficiente se for executado em um cluster de servidores.

Por meio do modelo de programação MapReduce, você conseguirá fazer com que algoritmo use os servidores de um cluster para processar a aplicação, melhorando o desempenho do processamento. Para isso, durante a execução do algoritmo, as tarefas map são criadas e distribuídas para os servidores, para que cada uma execute o algoritmo sobre uma parte dos dados. Após toda a base de dados for processada nessa fase, inicia-se então a fase reduce, responsável por agregar os resultados intermediários da fase map, gerando o relatório final com a contagem de hashtags.

Para dar início à implementação do código, abra a IDE NetBeans e crie um novo projeto chamado `ContaHashtags`. Dentro desse projeto, crie as seguintes classes:

- `ContaHashtagsMapper`: recebe como entrada uma sentença de texto, converte a sentença em uma lista de palavras, verifica quais palavras são hashtags e, para cada hashtag encontrada, emite como saída um par chave-valor no formato `<hashtag, 1>`.
- `ContaHashtagsReducer`: recebe como entrada os pares chave-valor emitidos pela classe map, itera sobre a lista de valores de cada chave, calculando a quantidade de vezes em que cada hashtag foi encontrada. Para cada hashtag, emite como saída um par chave-valor no formato `<hashtag, quantidade>`.
- `ContaHashtagsDriver`: faz a configuração e inicialização do job, indicando as classes que deverão ser usadas, bem como os arquivos de entrada e de saída.

Antes de implementar tais classes, é preciso importar as bibliotecas do Hadoop que utilizaremos em nosso código. Você pode encontrar essas bibliotecas na pasta `lib` do código `ContaHashtags` no repositório do livro.

Nesse exemplo, usamos a versão 2.6 do Apache Hadoop, sendo importadas as seguintes bibliotecas:

- `hadoop-common-2.6.0.jar`
- `hadoop-mapreduce-client-app-2.6.0.jar`
- `hadoop-mapreduce-client-common-2.6.0.jar`
- `hadoop-mapreduce-client-core-2.6.0.jar`

Para fazer a importação no NetBeans, clique com o botão direito do mouse sobre o nome do projeto e selecione a opção *Propriedades*. Será aberta uma caixa de diálogo, na qual você deverá selecionar a opção *Bibliotecas*, e depois clicar no botão *Adicionar JAR/Pasta*. Com essa ação, será aberta uma nova caixa de diálogo, na qual você deverá selecionar as bibliotecas descritas anteriormente.

Após a importação das bibliotecas, iniciaremos a implementação da classe `ContaHashtagsMapper`. Devemos primeiramente fazer referência às classes da biblioteca do Java e do Hadoop que serão utilizadas. Isso é feito usando o comando `import`, conforme descrito a seguir:

```
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
```

A classe `ContaHashtagsMapper` deve obrigatoriamente herdar a classe `Mapper`, disponibilizada pela biblioteca do Hadoop. Conforme o código a seguir, na extensão da classe devemos indicar quais objetos serão usados para os seguintes campos da função

`map`: chave de entrada, valor de entrada, chave de saída e valor de saída, respectivamente.

Nesse exemplo, os seguintes objetos foram definidos: `Object`, `Text`, `Text` e `IntWritable`. Estes são tipos especiais de dados oferecidos pela biblioteca MapReduce, que substituem os tipos de dados primitivos do Java, como `int` e `String`. Essa medida foi adotada para facilitar a serialização dos dados realizada pelo framework.

```
public class ContaHashtagsMapper
    extends Mapper<Object, Text, Text, IntWritable>{

}
```

No corpo da classe `ContaHashtagsMapper` serão declaradas duas variáveis: `palavra` e `numeroUm`. Elas também são objetos oferecidos pelo Hadoop (`Text` e `IntWritable`) e serão usadas para armazenar os valores referentes ao campo chave e o campo valor da saída do map, respectivamente.

```
private final static IntWritable numeroUm = new IntWritable(1);
private final Text palavra = new Text();
```

A partir da herança da classe `Mapper`, o desenvolvedor deve obrigatoriamente implementar o método `map`, conforme o código a seguir. Esse é o método principal de uma classe `Mapper`, sendo o local em que a lógica do problema de contagem de hashtags deverá ser implementada.

Perceba que nos argumentos desse método estão as variáveis `key` e `value`, que deverão corresponder aos dados de entrada de cada tarefa, no formato chave-valor. Nesse exemplo, cada tarefa map receberá como entrada no campo `value` uma linha de texto da base `tweets.csv`.

O objeto `Context` é utilizado para a interação com o ambiente Hadoop. Podemos usá-lo para diversas abordagens, como captura

de parâmetros, relato de progresso e escrita de dados das fases map e reduce.

```
@Override
public void map(Object key, Text value, Context context)
              throws IOException, InterruptedException {

}
```

O código a seguir descreve o algoritmo do método `map`. Para essa aplicação, foi criada uma variável chamada `tk` do tipo `StringTokenizer`. Essa variável será utilizada para converter nossa linha de texto em uma lista de tokens, correspondentes a cada palavra encontrada no texto.

```
StringTokenizer tk = new StringTokenizer(value.toString());
```

Após essa implementação, usamos um laço de repetição `while` para iterar sobre cada palavra encontrada na variável `tk`. No corpo desse laço é inicialmente criada uma variável `token`, que receberá a palavra emitida pelo método `nextToken`.

Como nosso objetivo é contar somente as hashtags, descartando as demais palavras do arquivo, verificamos se a palavra em questão inicia com uma cerquilha (`#`). Caso o resultado da verificação seja verdadeiro, a variável `palavra` recebe o conteúdo da variável `token`.

Perceba que, nesse código, foi realizado um tratamento da variável antes de armazená-la. Primeiramente convertemos a hashtag para minúsculo com o método `toLowerCase`, e utilizamos o método `replaceAll` com uma expressão regular para remover todos os caracteres não correspondentes às letras e à cerquilha. Esse tratamento fará com que nossa aplicação não faça distinção das hashtags *#BigCompras*, *#bigcompras* e *#BigCompras!*, por exemplo.

Por fim, nesse laço criamos um par chave-valor que será submetido para a função `reduce` por meio do método

`context.write`, no qual o campo chave será composto pela variável `palavra` e o campo valor pela variável `numeroUm`.

```
while (tk.hasMoreTokens()) {
    String token = tk.nextToken();
    if(token.startsWith("#")){
        palavra.set(token.toLowerCase()
               .replaceAll("[^a-zA-Z# ]", ""));
        context.write(palavra, numeroUm);
    }
}
```

Confira como deve ficar o código final da classe `ContaHashtagsMapper`:

```
public class ContaHashtagsMapper
  extends Mapper<Object, Text, Text, IntWritable>{

  private final static IntWritable numeroUm = new IntWritable(1);
  private final Text palavra = new Text();

  @Override
  public void map(Object key, Text value, Context context)
                throws IOException, InterruptedException {

      StringTokenizer tk = new StringTokenizer(value.toString());
      while (tk.hasMoreTokens()) {
          String token = tk.nextToken();
          if(token.startsWith("#")){
              palavra.set(token.toLowerCase()
                     .replaceAll("[^a-zA-Z# ]", ""));
              context.write(palavra, numeroUm);
          }
      }
  }
}
```

A segunda classe que devemos criar para o algoritmo de contagem de hashtags é a `ContaHashtagsReducer`. Aqui devemos fazer referência às seguintes classes:

```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
```

A classe `ContaHashtagsReducer` deverá herdar a classe `Reducer` oferecida pela biblioteca do Hadoop. Conforme mencionado anteriormente, as tarefas reduce recebem como entrada todos os dados intermediários gerados nas tarefas map. Portanto, é essencial que os tipos de dados da entrada da função reduce coincidam com os tipos de dados da saída da função `map`.

Podemos ver que isso ocorreu nesse exemplo, em que ambas as classes possuem os tipos `Tex` (para a variável `palavra`) e `IntWritable` (para a variável `numeroUm`):

```
public class ContaHashtagsReducer
    extends Reducer<Text,IntWritable,Text,IntWritable> {

}
```

A seguir, criaremos no corpo da classe uma variável do tipo `IntWritable`, que usaremos para armazenar a frequência de hashtags, conforme o código a seguir:

```
private IntWritable resultado = new IntWritable();
```

O método a ser implementado nessa classe será o `reduce`. Esse método recebe como parâmetro um objeto relacionado ao campo chave de entrada e um objeto do tipo `Iterable` que contém a lista de valores da chave recebida da fase map.

```
@Override
public void reduce(Text key, Iterable<IntWritable> values,
        Context context) throws IOException, InterruptedException
{

}
```

Com essa lista completa de todos os valores relacionados a uma mesma chave, podemos implementar uma lógica que faça a agregação desses dados. Nesse algoritmo, por exemplo, cada hashtag corresponde a uma chave enviada à tarefa reduce.

O conjunto de valores dessa chave é uma lista de números "uns",

que deverão ser somados, calculando assim a quantidade de vezes que a hashtag apareceu no texto. O resultado dessa soma é então atribuído à variável `resultado`. O método `context.write()` descrito a seguir é o responsável por escrever no arquivo de saída o resultado final da aplicação.

```
int soma = 0;
for (IntWritable val : values) {
    soma += val.get();
}
resultado.set(soma);
context.write(key, resultado);
```

Confira como deve ficar o código final da classe `ContaHashtagsReducer`:

```
public class ContaHashtagsReducer
    extends Reducer<Text,IntWritable,Text,IntWritable> {

    private IntWritable resultado = new IntWritable();

    @Override
    public void reduce(Text key, Iterable<IntWritable> values,
            Context context )
            throws IOException, InterruptedException {
        int soma = 0;
        for (IntWritable val : values) {
            soma += val.get();
        }
        resultado.set(soma);
        context.write(key, resultado);
    }
}
```

Além da classe `ContaHashtagsMapper` e `ContaHashtagsReducer`, devemos também implementar a classe `ContaHashtagsDriver`. Para isso, serão necessárias as seguintes referências:

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
```

```
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
```

A classe `ContaHashtagsDriver` é responsável pela configuração do job (uma aplicação MapReduce) que será executado no Hadoop. Um objeto do tipo `Job` é instanciado por meio da função `getInstance()`.

Feito isso, é necessário passar algumas informações adicionais a esse objeto, para que ele receba as configurações da nossa aplicação MapReduce. Por exemplo, devemos indicar qual será a classe `Map` e classe `Reduce` utilizadas, por meio dos métodos `setMapperClass` e `setReducerClass`, respectivamente.

Com o método `setOutputKeyClass`, indicamos o tipo de dado referente ao campo saída das fases map e reduce, assim como utilizamos o método `setOutputValueClass` para indicar tipo de dado do campo valor. Além desses parâmetros, é necessário indicar o caminho para o job encontrar os dados de entrada, com o método `addInputPath`, e indicar um caminho para salvar os dados de saída, com o método `addOutputPath`.

Por fim, usamos o método `waitForCompletion` para dar início à execução do job. O parâmetro `true` indica que desejamos visualizar as informações do processamento durante a execução do job.

```
public class ContaHashtagsDriver {

    public static void main(String[] args)
            throws Exception {

        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "conta hashtags");

        job.setJarByClass(ContaHashtagsDriver.class);
        job.setMapperClass(ContaHashtagsMapper.class);
        job.setReducerClass(ContaHashtagsReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
```

```
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));

        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```

Após todas as classes estarem devidamente implementadas, devemos salvar as alterações realizadas e gerar um JAR da aplicação. Para isso, clique com o botão direito do mouse no nome do projeto e selecione a opção *Construir*.

A partir da janela apresentada, crie um JAR com o nome `ContaHashtags.jar`. Caso tenha algum problema para executar essa operação, você pode utilizar o JAR disponível no repositório do livro.

Passo 2: envio da base de dados para o HDFS

Após termos implementado a aplicação MapReduce, devemos executá-la em um ambiente Hadoop. Porém, antes de executarmos, é preciso enviar nossa base de dados de entrada para o HDFS.

Nessa atividade, estamos partindo do princípio que você já tenha um ambiente Linux com o framework Hadoop funcionando corretamente. Caso ainda não tenha, você pode instalar o Hadoop seguindo o tutorial disponível em:

https://hadoop.apache.org/docs/r2.6.0/hadoop-project-dist/hadoop-common/SingleCluster.html

Para enviar a base `tweets.csv` do nosso sistema de arquivos local para o HDFS, devemos abrir um terminal Linux e digitar o seguinte comando:

```
$ hadoop fs -mkdir bases
$ hadoop fs -put $HOME/tweets.csv bases
$ hadoop fs -ls bases
```

```
Found 1 items
-rw-r--r-- 1 hadoopuser supergroup ... bases/tweets.csv
```

Na primeira linha, utilizamos o comando `-mkdir` da API Hadoop FS Shell para criarmos um diretório dentro do HDFS com o nome `bases`. Na segunda, usamos o `-put` para enviar uma cópia do arquivo de uma base local (`$HOME/tweets.csv`) para o diretório que acabamos de criar no HDFS (bases).

Para verificar se os dados foram devidamente copiados, na terceira linha utilizamos o comando `-ls` para listar os arquivos do diretório `bases`. Esse comando deverá retornar a lista de arquivos dentro do diretório.

Passo 3: execução da aplicação em ambiente Hadoop

Tendo enviado a base de dados de entrada para o HDFS, podemos enfim executar nossa aplicação. Para isso, digite o seguinte comando em um terminal:

```
$ hadoop jar $HOME/ContaHashtags.jar ContaHashtagsDriver bases sai
da
16/07/25 16:48:10 INFO input.FileInputFormat: Total input ...
16/07/25 16:48:10 WARN snappy.LoadSnappy: Snappy native ...
16/07/25 16:48:10 INFO util.NativeCodeLoader: Loaded the ...
16/07/25 16:48:10 INFO snappy.LoadSnappy: Snappy native ...
16/07/25 16:48:11 INFO mapred.JobClient: Running job: ...
16/07/25 16:48:12 INFO mapred.JobClient:  map 0% reduce 0%
16/07/25 16:48:15 INFO mapred.JobClient:  map 100% reduce 0%
...
```

Para fazer a chamada da nossa aplicação, usamos o comando `hadoop jar`, passando como parâmetro o JAR da aplicação que desenvolvemos. Além disso, indicamos na sequência a classe `ContaHashtagsDriver`, para que a JVM saiba por onde iniciar a aplicação.

Os últimos dois parâmetros indicam o caminho do diretório de entrada e o caminho do diretório de saída dos dados,

respectivamente. Após iniciar a execução do comando, você poderá acompanhar no terminal informações sobre o andamento da execução das tarefas map e reduce, até o momento em que todas as tarefas forem concluídas.

Passo 4: verificação dos resultados

Tendo finalizada a aplicação, podemos acessar o diretório de saída para verificar se o arquivo de saída foi gerado corretamente. Para isso, liste os arquivos do diretório de saída pelo seguinte comando:

```
$ hadoop fs -ls saida
Found 2 items
-rw-r--r--   1 hdpuser ... /saida/_SUCCESS
drwxr-xr-x   - hdpuser ... /saida/_logs
-rw-r--r--   1 hdpuser ... /saida/part-r-00000
```

O diretório `_SUCCESS` é um diretório gerado pelo Hadoop que indica que a aplicação foi executada com sucesso. O diretório `_logs` contém registros de log do job executado, no qual você pode encontrar informações adicionais sobre o processamento.

O arquivo `saida/part-r-00000` é o que contém de fato o resultado da aplicação. Podemos verificar o resultado desse arquivo por meio do seguinte comando:

```
$ hadoop fs -cat saida/part-r-00000

#bigcompras    42
#demora    1
#descontos    10
#desisti    4
#fretecaro    6
#indignada    1
#irritada    2
#lento    2
#livros    6
#medo    3
#nuncamais    1
```

```
#precojusto    7
#queropresente    7
#revoltada    7
#satisfeita    3
#sucesso    8
```

Pronto! Conseguimos implementar e executar uma aplicação MapReduce no Hadoop. O arquivo final gerou para cada linha um par chave/valor, contendo a hashtag e o número de vezes em que ela foi encontrada na base de dados. Por exemplo, a hashtag *#BigCompras* foi a que teve maior frequência, com 42 aparições, seguida da hashtag *#descontos*, com 10 aparições.

É importante ressaltar que, com o mesmo código implementado, podemos executar essa aplicação em cluster com dezenas, centenas ou até mesmo milhares de máquinas. O código não precisa ser alterado e, mesmo assim, o MapReduce saberá utilizar as máquinas apropriadamente para garantir o desempenho da aplicação. A transparência e escalabilidade são uns dos grandes benefícios do Hadoop.

Caso você queira se aventurar ainda mais com Hadoop, deixo aqui dois desafios:

1. Desenvolver uma aplicação MapReduce que faça a contagem apenas das top **n** hashtags mais encontradas, sendo que **n** deverá ser um parâmetro indicado pelo usuário;
2. Desenvolver uma aplicação MapReduce que conte somente quantas vezes apareceu a hashtag *#BigData*.

Ficarei muito feliz se você conseguir. Para facilitar o desenvolvimento, utilize o esqueleto do código que implementamos e altere somente o conteúdo dos métodos. Bom trabalho!

3.6 CONSIDERAÇÕES

Neste capítulo, podemos identificar as mudanças impostas por Big Data em relação ao processamento de dados. Podemos perceber que as tecnologias tradicionais não oferecem suporte para os requisitos atuais de manipulação de dados. Por esse motivo, é crucial a utilização de novas tecnologias para se ter uma solução escalável, ágil e de bom desempenho.

Além disso, a computação em nuvem pode ser uma grande aliada para o armazenamento e processamento de soluções de Big Data, oferecendo diferentes cenários de aquisição. Porém, principalmente em soluções de Big Data, a transferência de dados para os servidores da nuvem ainda é um desafio.

Este capítulo teve como objetivo auxiliar nas seguintes perguntas de um projeto de Big Data:

- Como obter uma solução escalável?
- Quais tecnologias utilizar para processar os dados?
- Quais os benefícios em usar o Hadoop?
- Como agir no processamento em tempo real?
- Que tipo de processamento preciso para a minha aplicação?
- Quais benefícios posso obter com a computação em nuvem?

No próximo capítulo, veremos que, além do surgimento de novas tecnologias para o processamento de dados, existem novas técnicas e mecanismos para a análise de dados, permitindo extrair informações valiosas para o negócio.

Para saber mais

1. BARLOW, Mike. *Real-Time Big Data Analytics:* Emerging Architecture. O'Reilly Media, Inc., 2013.

2. DEAN, Jeffrey; GHEMAWAT, Sanjay. *MapReduce:* simplified data processing on large clusters. Sixth Symposium on Operating System Design and Implementation, Dez. 2004.

3. ELLIS, Byron. *Real-time analytics:* techniques to analyze and visualize streaming data. John Wiley & Sons, 2014.

4. GHEMAWAT, Sanjay; GOBIOFF, Howard; LEUNG, Shun-Tak. *The Google file system.* ACM SIGOPS operating systems review, v. 37, n. 5, Out. 2003.

5. GOLDMAN, Alfredo; KON, Fabio; JUNIOR, Francisco Pereira; POLATO, Ivanilton; PEREIRA, Rosangela de Fátima. *Apache Hadoop:* conceitos teóricos e práticos, evoluçao e novas possibilidades. XXXI Jornadas de atualizaçoes em informatica, 2012.

6. KARAU, Holden; ZAHARIA, Matei; WENDELL, Patrick; KONWINSKI, Andy. *Learning spark:* lightning-fast big data analysis. O'Reilly Media, Inc., 2015.

7. LEIBIUSKY, Jonathan; EISBRUCH, Gabriel; SIMONASSI, Dario. *Getting started with storm:* Continuous streaming computation with Twitter's cluster technology. O'Reilly Media, Inc., 2012.

8. LUBLINSKY, Boris; YAKUBOVICH, Alexey; SMITH, Kevin. *Professional Hadoop Solutions.* John Wiley & Sons, 2013.

9. NABI, Zubair. *Pro Spark Streaming:* The Zen of Real-Time Analytics Using Apache Spark. Apress, 2016.

10. WHITE, Tom. *Hadoop:* The definitive guide._ O'Reilly Media, Inc., 2015.

Capítulo 4

ANALISANDO OS DADOS

"Estamos nos afogando em informações e famintos por conhecimento." — John Naisbitt

Após a captura, armazenamento e processamento dos dados, iniciamos a fase de análise. Atrevo-me a dizer que essa talvez seja a fase mais empolgante em Big Data, na qual temos inúmeras possibilidades de converter dados brutos em conhecimento.

Sabemos que analisar dados não é uma prática recente. Entretanto, somente nos últimos anos estamos presenciando a adesão por essa prática de forma ampla, sendo utilizada como fator chave para alavancar negócios, reduzir custos e aperfeiçoar produtos e serviços.

Costumo dizer em sala de aula que o processo de análise de dados é similar ao de relacionamento a dois. No início da relação, é importante que haja um processo de descoberta, de conhecimento.

Devemos compreender qual o objetivo nosso e do nosso "parceiro" na relação. Por esse motivo, para conhecer os dados, é importante fazer perguntas como: qual sua origem? A quem pertencem? Em que contexto estão inseridos? Terei benefícios ao utilizar esses dados? Eles vão agregar em algo?

Responder tais perguntas nos ajuda a decidir se vale a pena ou não investir na relação. Haverá casos em que o relacionamento será rompido, pois os dados não estão condizentes com o que foi

esperado.

Entretanto, caso se perceba que os dados são valiosos, mas que precisam de ajustes, é possível iniciar uma fase de preparação. Nela, modificações serão realizadas para que eles fiquem de acordo com o desejado.

Feito isso, o relacionamento pode ser aprofundado. A investigação sobre os dados passa a ser mais próxima, gerando assim novas descobertas e, consequentemente, novos frutos. Mas lembre-se de que, assim como em um relacionamento, caso você não tenha agido corretamente, os frutos podem não aparecer.

Analisar dados é um processo que envolve tanto ciência quanto arte. Infelizmente, os analistas não possuem uma bola de cristal capaz de fazer predições. Torna-se necessário adquirir a arte de identificar quais dados utilizar, como integrá-los e quais perguntas serão úteis na tomada de decisão.

Aliado a essas habilidades, é essencial que se aplique uma abordagem científica a esse processo, exigindo um domínio de conhecimento do analista para usar as ferramentas e técnicas de forma apropriada e confiável. O domínio dessas duas habilidades é que faz o processo de análise de dados ser um sucesso. Falaremos sobre essa abordagem científica e criativa na análise de dados neste capítulo.

4.1 CARACTERÍSTICAS DA ANÁLISE DE DADOS

Caso você nunca tenha se aventurado a realizar a análise de dados com o objetivo de extrair informações úteis, você pode não saber que existem algumas particularidades nessa prática. Para dar início a essa jornada, verifique a seguir premissas que devem ser

consideradas.

Os dados utilizados estão normalmente "sujos"

Embora seja comum encontrarmos em livros de análises de dados exemplos que utilizam bases de dados estruturadas e prontas para serem analisadas, no cenário real é muito raro isso acontecer.

Provavelmente a base de dados que você deseja analisar terá dados incompletos, inconsistentes, corrompidos, duplicados, em formatos inadequados, com caracteres indesejados, entre tantas outras questões. Por esse motivo, é necessário um profissional com habilidades para realizar o tratamento dos dados, antes de a análise ser efetivamente realizada.

Gasta-se mais tempo preparando do que analisando os dados

Há uma estimativa de que, no processo de análise de dados, 80% do tempo é gasto para limpar e preparar os dados. Parece muito tempo, não? Mas é o que acontece na maioria das análises.

Como cada base de dados possui sua peculiaridade, muitas tarefas de tratamento precisam ser avaliadas e definidas manualmente, não existindo muitos meios para automatizar completamente esse processo. Então, não se espante se você demorar muito tempo nessa etapa. Embora seja oneroso, o tratamento dos dados evita inconsistências nos resultados das análises.

Procura de uma agulha em um palheiro

Analisar uma grande base de dados em busca de padrões pode significar muitas vezes um processo análogo ao de procurar uma agulha em um palheiro. Essa analogia existe pelo fato de que

encontrar um padrão diante de uma infinidade de dados é uma tarefa muitas vezes complexa e demorada.

Entretanto, no contexto de Big Data, em que se trabalha com uma avalanche de dados, alguns pesquisadores dizem que o desafio da análise de dados não é somente encontrar a agulha em um palheiro, mas encontrar o que de fato é a agulha. Ou seja, identificar qual pergunta é possível se responder a partir dos dados.

Garbage in, garbage out

Aqui voltamos à importância da qualidade dos dados durante o processo de análise. No contexto de Big Data, é muito comum a utilização de dados em sua forma bruta, que não passaram por um processo de refinamento. O problema é que, sem um processo de inspeção, pode ocorrer que dados incorretos não sejam descartados ou corrigidos.

Uma vez que esses dados sejam usados na construção de um modelo analítico, o resultado obtido pode não representar a realidade dos fatos. Se uma organização faz a tomada de decisão orientada por esses resultados, ela pode desencadear uma série de ações baseadas em fatos inconsistentes.

Correlação não implica causalidade

Esse é um dos principais fundamentos da estatística: correlação não implica em causalidade! Enquanto, na causalidade, você prova que "o acontecimento A causa o acontecimento B", a correlação apenas indica que "A" e "B" tendem a ser observados no mesmo tempo, mas não há necessariamente uma causalidade entre eles. Pode ser que a correlação seja apenas uma coincidência.

Para inferir uma causalidade, é preciso a realização de testes estatísticos e experimentos controlados que façam essa validação. Se

a correlação sempre implicasse causalidade, poderíamos identificar algumas tendências um tanto quanto estranhas, como por exemplo de que, sempre que a venda de sorvetes aumenta, aumenta também o número de afogamentos. Por esse motivo, tenha sempre muito cuidado na interpretação dos dados.

É fácil fazer a análise de dados de forma errada

Isso é um perigo alertado por muitos pesquisadores. As ferramentas de análise de dados disponíveis atualmente facilitou a construção de inúmeros algoritmos utilizando uma diversidade de dados. Entretanto, um erro cometido ou uma interpretação errada dos dados durante esse processo pode gerar resultados que nos deixam animados, mas que na verdade não condizem com a realidade.

Por esse motivo, é extremamente necessária a validação das respostas obtidas, principalmente quando utilizamos bancos de dados de grande volume, em que as incoerências podem não ser claramente perceptíveis.

4.2 O PROCESSO DE ANÁLISE DE DADOS

Quando falamos em Big Data e em análise de dados, é comum ouvirmos palavras como identificação de padrões, modelagem dos dados, detecção de grupos, classificação de dados. Essas atividades são possíveis por meio da utilização de técnicas há muito tempo desenvolvidas, como técnicas estatísticas, matemáticas, de aprendizado de máquina e de mineração de dados.

Pense em uma solução em que um sistema computacional receba informações de sensores instalados em uma fábrica e consiga identificar automaticamente que uma das máquinas usadas está prestes a falhar, mesmo antes de ela ter apresentado problemas

perceptíveis ao olhar humano. Imagine também em um sistema computacional que tenha a capacidade de diagnosticar uma doença de forma automatizada, com base na análise dos dados coletados sobre o paciente.

O que podemos perceber de comum nessas duas soluções? Ambas utilizavam meios para fazer predições de forma automatizada. É para soluções como estas que se aplicam as técnicas de aprendizado de máquina.

O foco principal dessa área de estudo é permitir que o computador aprenda, ou seja, que ele seja capaz de organizar seu conhecimento, sem que isso seja explicitamente programado. Quando aplicado à análise de dados, o aprendizado de máquina é utilizado para automatizar a construção de um modelo analítico.

Embora essa técnica tenha sido adotada com maior ênfase somente nos últimos anos pelas organizações, já temos exemplos inspiradores resultantes dessa adoção, tais como:

- Detecção de fraude em transações com cartão de crédito;
- Diagnóstico de doenças;
- Identificação de atividades criminosas;
- Segmentação de clientes;
- Descoberta de genes.

Voltando ao exemplo da varejista Big Compras, imagine que a equipe de analistas deseja identificar se existe um padrão na compra dos clientes em relação a escolha dos produtos. Será que existem produtos que sempre (ou na maioria das vezes) são adquiridos na mesma compra?

Saber isso é importante, pois gera insights para a criação de campanhas e definição de ofertas. Mas eis o problema. Para realizar

essa análise, será necessário observar dados históricos de 5 milhões de registros de compras.

Uma alternativa para esse problema é a adoção de técnicas de mineração de dados. Utilizando técnicas estatísticas, matemáticas e de aprendizado de máquina, a mineração de dados é um campo de estudo com foco na extração de informações úteis e padrões ocultos em conjuntos massivos de dados.

Embora seja similar a uma relação, para se obter sucesso na análise de dados, é preciso estabelecer e seguir um processo sistemático. Existem diversas definições de processos de análise de dados na literatura, tais como o SEMMA (*Sample, Explore, Modify, Model, and Assess*) e CRISP-DM (*Cross Industry Standard Process for Data Mining*).

Embora cada processo tenha definições distintas, em geral, eles envolvem as seguintes etapas:

1. **Entendimento do negócio:** aqui são definidas as perguntas, o objetivo da análise de dados e o plano a ser seguido;
2. **Compreensão dos dados:** etapa utilizada para coletar e explorar os dados, aumentando a compreensão sobre sua estrutura, atributos e contexto;
3. **Preparação dos dados:** após a análise exploratória, inicia-se o processo de limpeza, filtragem, estruturação, redução e integração dos dados;
4. **Modelagem dos dados:** envolve as tarefas de seleção dos dados, definição e construção do modelo;
5. **Validação do modelo:** os resultados gerados pelo modelo são avaliados, para verificar se a precisão obtida está satisfatória e coesa;
6. **Utilização do modelo:** após serem validados, os resultados dos modelos são utilizados e monitorados.

Nas próximas seções, abordaremos aspectos técnicos das etapas de compreensão e preparação dos dados, modelagem dos dados e validação do modelo. O foco é apenas apresentar aspectos dessas etapas, fornecendo uma visão geral sobre elas. Tenha em mente que essas são áreas com conceitos muito amplos, que excedem o escopo do livro.

4.3 PREPARANDO OS DADOS

Sabe aquele mundo ideal, no qual acessamos um software de análise de dados, inserirmos nossa base, pressionarmos um botão e rapidamente nosso modelo é gerado e os padrões ocultos são revelados? Pois é, infelizmente esse mundo ainda não existe.

Conforme já descrito, a fase de preparação, tratamento ou pré-processamento dos dados é essencial na análise de dados, sendo a tarefa que demanda maior tempo e trabalho. Quando falamos de análise dos dados no contexto de Big Data, essa fase se tornou ainda mais importante, uma vez que muitas vezes os dados usados estão em seu formato original, sem nenhuma "lapidação" realizada sobre eles.

Mas por que será que preparar os dados é algo tão demorado? Confira a seguir algumas das atividades realizadas nessa fase e a resposta para essa pergunta.

Limpeza dos dados

Está lembrado do termo "*garbage in, garbage out*"? O processo de limpeza de dados é necessário exatamente para minimizar essa ocorrência, de gerar resultados incorretos devido às "sujeiras" existentes nos dados de entrada.

O processo de limpeza requer uma inspeção minuciosa dos

dados, bem como a realização de operações de correção e remoção, conforme a necessidade. Para exemplificar, considere os registros a seguir, referentes aos dados cadastrais dos clientes da Big Compras.

id	nome	idade	sexo	cidade
500	"pedro"	32	"M"	"São Paulo"
501	"maria"	41	"F"	"Curitiba"
502	"jonas"	25	"1"	"05360-152"
503	"lucia"	38	"2"	"Londrina"
504	"lucas"	29	"masc"	"Aracaju"
505	"lucas"	29	"masc"	"Aracaju"

Consegue perceber alguns problemas nesses registros? Avaliando a coluna `sexo`, por exemplo, percebemos que ela está registrada de diferentes formas: com siglas `F` e `M`, com números `1` e `2` (provavelmente para indicar uma categoria), e com o texto `masc`, para representar o sexo masculino.

Também podemos perceber que alguma coisa está errada na coluna cidade do cliente 502, pois ela contém o código postal em vez do nome da cidade. Além disso, os registros da quinta e sexta linha parecem ser referentes ao mesmo cliente, estando possivelmente duplicados.

Uma vez que esses problemas são encontrados, decisões devem ser tomadas para padronizar e ajustar as informações. Por exemplo, no caso da coluna `sexo`, é possível criar uma função que ajuste todos os registros de acordo com uma regra estabelecida, deixando todos os registros preenchidos com `F` ou `M`.

Como essa transformação pode afetar inúmeros registros da base de dados, é preciso ter cuidado para não aplicar uma regra que realize a transformação incorretamente. Para evitar essa situação, é

indicado testar a operação de limpeza em uma pequena parte dos dados primeiramente, para somente depois aplicá-la em toda a base de dados.

As seguintes perguntas podem auxiliar na identificação de quais operações devem ser realizadas:

- Existem dados duplicados?
- Existem dados com informações incompletas?
- Existem dados com erros de digitação?
- Existem dados iguais representados de diferentes formas?
- Existem dados que violam as regras de negócio?

Embora em alguns casos seja possível realizar uma inspeção manual desses dados, isso pode ser muito custoso, principalmente no contexto de Big Data. Linguagens como R e Python podem ajudar nessas operações, dado que elas possuem pacotes com funções específicas para tratamento de dados, facilitando consideravelmente esse processo.

Manipulação de dados ausentes

Ao avaliarmos a base de dados que será utilizada na análise, outra situação que podemos identificar é a ausência de dados, ou seja, registros com informações incompletas. Para exemplificar, verifique a seguir um arquivo simplificado de registros de compras de clientes da Big Compras:

id	data	valor	frete	pagamento
100	2016-03-02	250,00	22,00	boleto
101	2016-03-02	500,00	30,00	boleto
102	2016-03-03	420,00	-	cartão
103	2016-03-03	108,00	15,50	boleto

104	2016-03-04	100,00	5,85	-
105	2016-03-04	216,00	12,00	cartao

Podemos verificar que dois registros estão com campos sem informações. O registro `102` não possui informação sobre o campo `frete`, e o registro `104` não possui informação sobre o campo `pagamento`. O que fazemos com esses registros em nossa análise?

A opção mais simples nesse caso é eliminar os registros com informações incompletas da análise, pois assim não teremos problemas, correto? Não exatamente.

Imagine se, além desses dois registros, a quantidade de registros com informações incompletas fosse superior a 20% do conjunto total de dados? Parece um desperdício não utilizá-los, não?

Embora seja comum descartar registros com dados ausentes, a adoção dessa prática oferece riscos de gerar estimativas viesadas e inconsistentes, uma vez que os registros descartados podem conter padrões significativos para a análise. Para não descartar os registros com dados ausentes em nossa análise, Daniel T. Larose e Chantal D. Larose, autores do livro *Data Mining and Predictive Analytics*, indicam as seguintes abordagens:

- Substituir o dado ausente com alguma constante, especificada pelo analista;
- Substituir o dado ausente pela média ou moda do campo;
- Substituir o dado ausente com um valor gerado aleatoriamente a partir de uma distribuição observada;
- Substituir o dado ausente a partir de valores baseados em outras características do registro.

Embora essas abordagens sejam indicadas, é preciso muito cuidado para selecionar qual a mais apropriada, evitando que a

substituição gere informações inapropriadas ao conjunto de dados e, consequentemente, à análise.

Identificação de anomalias

Para darmos início à explicação de identificação de anomalias, considere os seguintes registros de compras de clientes da Big Compras:

id	data	valor	frete	pagamento
106	2016-03-05	120,00	10,00	boleto
107	2016-03-05	350,00	14,00	cartão
108	2016-03-06	400,00	22,50	boleto
109	2016-03-06	310,00	40,00	cartao
110	2016-03-06	250,00	15,00	cartao
111	2016-03-06	135,00	20,00	cartao
112	2016-03-06	280,00	15,00	cartao
113	2016-03-06	350,00	18,00	cartao
114	2016-03-06	310,00	50,00	cartao
115	2016-03-06	120,00	10,00	cartao
116	2016-03-06	5000,00	65,00	cartao

Opa, parece que o registro `116` possui um valor de compra bem diferente de todos os outros registros restantes. Enquanto que os outros registros ficaram com valores entre R$ 100,00 e R$ 400,00, esse teve o valor de compra de R$ 5000,00.

Como esse registro apresenta um valor que desvia significativamente do padrão normal do restante dos dados, ele é considerado uma anomalia (do inglês *outlier*). Mas por que identificar anomalias é uma tarefa importante na preparação de dados?

A detecção de anomalias é importante porque ela permite identificar se existe algum erro na entrada de dados numéricos, bem como nos ajuda a perceber a existência de valores extremos que influenciarão alguns métodos estatísticos, mesmo em casos em que as anomalias correspondam a dados válidos.

A média é um exemplo de cálculo que sofre essa influência. Se considerarmos a média dos 10 primeiros registros (106-115), teremos como resultado o valor médio de compras de R$ 262,50. No entanto, se também considerarmos o último registro (116), essa média aumentaria para R$ 693,20, um valor muito acima do que todos os demais registros.

Quando temos um grande volume de dados, identificar uma anomalia em dados apresentados em formato tabular não é uma tarefa fácil. Como solução, os gráficos podem auxiliar bastante esse processo, como por exemplo, o diagrama de caixa (*boxplot*) e o gráfico de dispersão (*scatterplot*), conforme veremos no capítulo seguinte.

Transformação dos dados

Mesmo em situações nas quais os dados usados para a análise já estejam limpos e sem informações ausentes, pode ser necessário aplicar técnicas de transformação sobre eles. Considere a título de exemplificação o conjunto de dados a seguir, referente aos valores dos produtos da Big Compras:

id	preço
001	20,00
002	180,00
003	30,00
004	65,00
005	52,00

006	23,00

id	preço
007	97,00
008	82,00
009	261,00
010	347.00

Perceba que o campo `preço` apresenta valores bem distintos, utilizando como unidade de medida a moeda Real. Para evitar que essa diferença influencie de forma tendenciosa a construção do modelo, uma transformação muito adotada é a normalização dos dados.

O processo de normalização de variáveis numéricas é aplicado para ajustar a escala dos valores das variáveis. Uma das formas de normalização é a transformação linear, também conhecida como normalização *min-max*, dado que o cálculo é feito com base nos valores mínimo e máximo de cada atributo no ajuste da escala. Aplicando essa normalização, os registros teriam os seguintes valores:

id	preço	preço normalizado
001	20,00	0
002	180,00	0,49
003	30,00	0,03
004	65,00	0,14
005	52,00	0,1
006	23,00	0,01
007	97,00	0,25
008	82,00	0,19

009	261,00	0,74
010	347.00	1

Além da normalização, dependendo do algoritmo utilizado na modelagem, outros ajustes podem ser necessários:

- Transformação de dados numéricos para categóricos;
- Transformação de dados categóricos para numéricos;
- Agregação de dados, por meio da combinação de dados de diferentes conjuntos em uma única fonte, de forma coerente;
- Criação de novos atributos.

Redução dos dados

Mesmo com as possibilidades oferecidas pelas tecnologias de Big Data para processar um grande volume de dados, é possível que o processamento de uma base de dados com centenas de variáveis e milhões de registros seja muito caro computacionalmente, resultando em um gargalo de desempenho em alguns algoritmos. Para casos como esses, são aplicadas técnicas de redução e sintetização de dados em busca de reduzir a dimensionalidade dos dados.

Mas ora, se preciso reduzir a base de dados, não basta apenas selecionar uma parte do conjunto de dados? Não é bem assim. Caso façamos a redução dessa forma, não temos garantia de que registros significativos não foram descartados do modelo.

Na verdade, a técnica de redução de dados tem como objetivo gerar uma representação reduzida do conjunto de dados, porém mantendo os mesmos (ou próximo a isso) resultados da análise. Para isso, essa prática requer uma fase de seleção de atributos, identificando quais são irrelevantes para a análise e podem ser removidos da base. Além de reduzir a complexidade do

processamento, a eliminação dos atributos irrelevantes também evita que eles atrapalhem o resultado final do modelo.

Uma técnica muito conhecida para a prática de redução de dados é a de Análise de Componentes Principais (*Principal Component Analysis* — PCA). Essa técnica tem como objetivo detectar a correlação entre as variáveis. E caso seja detectado uma forte correlação entre elas, cria-se um conjunto menor de combinações lineares dessas variáveis, reduzindo assim a dimensionalidade dos dados.

Conseguiu perceber quantas tarefas são necessárias realizar antes de iniciar a análise de dados de fato? Mesmo estando superansioso para construir o modelo e assim obter os resultados, não há como fugir dessa primeira etapa.

Sem ela, você até pode conseguir seguir adiante, porém, as possibilidades de encontrar problemas na execução do algoritmo ou nos resultados obtidos são muito grandes. Ou seja, preparar os dados para a análise é um "mal necessário".

4.4 CONSTRUINDO O MODELO

Com os dados preparados para a análise, damos início à fase de modelagem dos dados. É nessa etapa que utilizamos um algoritmo para gerar a resposta que estamos procurando.

A figura a seguir apresenta uma lista de tarefas comuns em mineração de dados para obtenção dessas respostas. Em geral, essas tarefas podem ser divididas em duas categorias: descritiva e preditiva.

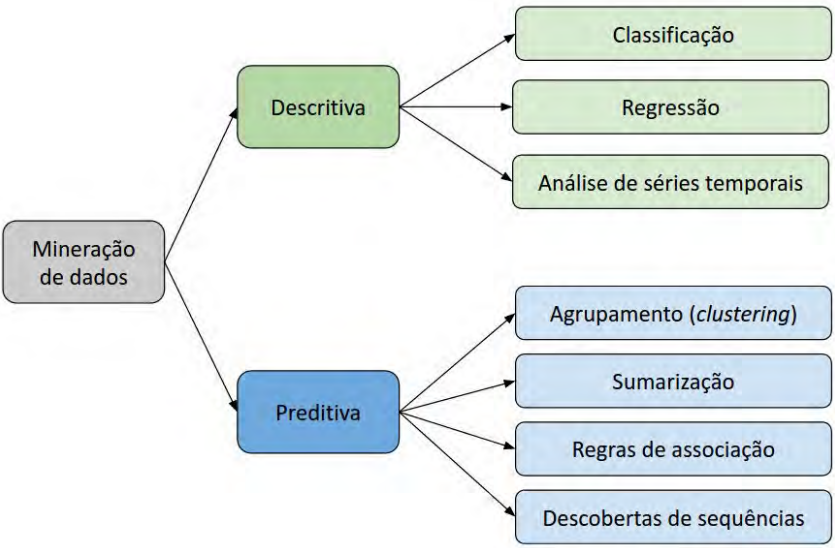

Figura 4.1: Tarefas de mineração de dados

Enquanto que o foco principal das tarefas da categoria descritiva é caracterizar e apresentar as propriedades de um conjunto de dados de maneira concisa e informativa, o objetivo das atividades preditivas é construir um modelo para prever as propriedades e/ou tendências de um conjunto de dados desconhecido. Mas qual a diferença entre cada uma dessas tarefas? Veja um resumo sobre cada uma a seguir.

Classificação

Considerado por muitos pesquisadores a tarefa mais comum em mineração de dados, a classificação tem como objetivo utilizar atributos de um objeto para determinar a qual classe ele pertence. Imagine, por exemplo, que a varejista Big Compras deseja avaliar as transações de compras dos clientes pelo aplicativo e identificar se alguma transação online de cartão de crédito é fraudulenta.

A cada transação é gerado um conjunto de atributos, tais como:

data e horário da transação, valor da transação, localização, lista de produtos comprados. A partir desses atributos, o objetivo é classificar a transação como fraudulenta ou idônea. Esse objetivo pode ser alcançado com uso de algoritmos de classificação.

Os algoritmos de classificação necessitam de um conjunto de dados rotulados para gerar o modelo preditivo. Por exemplo, para o cenário de detecção de fraude, devemos utilizar como entrada do algoritmo um conjunto de dados históricos de transações, tendo para cada transação um conjunto de atributos da transação e um atributo especial, que indique se a transação foi rotulada (classificada) como fraudulenta ou não.

A partir desse conjunto de dados, o algoritmo de classificação vai "aprender" quais combinações dos atributos estão associados com cada rótulo, gerando assim o modelo. Após essa etapa, novos registros de transações, agora não rotulados, são enviados ao modelo, que deverá gerar como resultado a predição do rótulo de cada uma delas.

Algoritmos que utilizam dados rotulados na fase de treinamento do modelo são categorizados como algoritmos de aprendizado supervisionado, conforme ilustrado adiante.

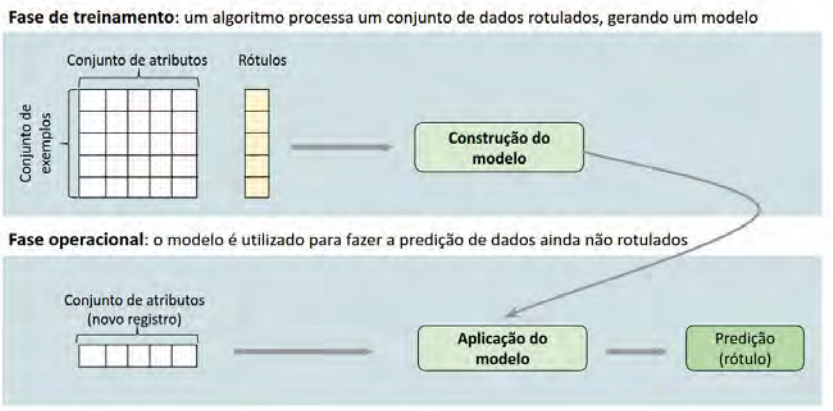

Figura 4.2: Exemplo de aprendizado supervisionado

São exemplos de algoritmos de classificação: árvores de decisão, classificação Bayesiana, classificação baseada em regras, máquinas de vetores suporte (*support vector machines*) e redes neurais.

Regressão

Além da classificação, outra técnica de aprendizado supervisionado é a regressão. A diferença entre essas técnicas é que, enquanto a classificação tenta prever à qual classe pertence uma nova instância, a regressão busca prever um valor numérico contínuo.

Por exemplo, imagine que, em vez de prever a adesão a uma oferta de cartão, a equipe da Big Compras estivesse interessada em prever o total de vendas nos próximos meses. Perceba que aqui a resposta desejada é um valor contínuo, e não um rótulo do tipo "sim/não". Esse valor será obtido com base na análise de valores passados de um conjunto de dados.

São exemplos de algoritmos de regressão: regressão linear simples e múltipla, regressão não linear simples e múltipla.

Análise de séries temporais

Essa tarefa é aplicada a bancos de dados de séries temporais, ou seja, bancos de dados que contenham sequências de valores ou eventos armazenados sucessivamente em função do tempo. Tais valores são normalmente obtidos em um mesmo intervalo de tempo, como a cada dia, hora ou minuto.

Por exemplo, no caso da Big Compras, esse banco poderia ser o histórico de vendas de uma categoria de produtos ao longo do tempo. A partir da análise de série temporal, torna-se possível observar o comportamento desses dados em relação ao tempo, podendo assim fazer estimativas como a previsão de vendas,

controle de estoque, lucro mensal, entre outras.

Sumarização

Essa tarefa descritiva tem como objetivo mapear os dados em subconjuntos, podendo ocorrer em diversos níveis, para fazer uma descrição compacta sobre eles. Aqui são utilizadas desde operações estatísticas básicas (como média, mediana, moda e desvio padrão) até operações mais complexas (como a derivação de regras de sumarização).

Se pensarmos no caso da varejista Big Compras, por exemplo, a sumarização pode ser útil para analisar dados relacionados à navegação dos clientes no aplicativo. Isso gera informações como a média de minutos permanecidos no aplicativo, de produtos pesquisados e produtos comprados em uma escala diária, semanal e anual.

Agrupamento

Lembra-se de que, na tarefa de classificação, é necessário enviar um conjunto de dados rotulados para que o modelo seja treinado? Mas como fazer em situações nas quais não sabemos antecipadamente esse rótulo?

Por exemplo, imagine que a equipe da Big Compras tivesse como objetivo realizar campanhas de marketing e precisasse segmentar seus clientes com base em comportamentos ou características similares. O problema é que a equipe não sabe como "rotular" esses clientes, pois ela não conhece os padrões existentes nos dados para fazer essa inferência. Para situações como essa, em que o objetivo é que um algoritmo seja capaz de detectar padrões ocultos nos dados, utiliza-se a tarefa de agrupamento.

Também conhecido como *clustering* ou segmentação, nessa

tarefa um algoritmo de agrupamento analisa um conjunto de exemplos não rotulados, com foco em determinar se alguns deles podem ser agrupados de acordo com uma medida de similaridade, gerando assim os grupos (ou clusters). Dessa forma, um algoritmo de agrupamento poderia segmentar clientes da Big Compras de acordo com os padrões encontrados, tais como: nível de renda, faixa de idade, preferências de marca etc. Essa mesma estratégia pode ser adotada em inúmeras outras aplicações, tais como o agrupamento de pacientes com sintomas similares e a classificação de documentos.

Conforme apresentado na figura a seguir, os algoritmos que não utilizam conjuntos de dados rotulados no processo de aprendizado são denominados algoritmos de aprendizado não supervisionado. Isso porque eles não recebem nenhuma indicação em relação aos padrões que devem ser detectados.

Durante a fase de treinamento, um modelo é criado para identificar os grupos com base nas similaridades. Estando o modelo construído, na fase operacional novos registros são enviados ao modelo, que deverá identificar a qual grupo esse registro pertence.

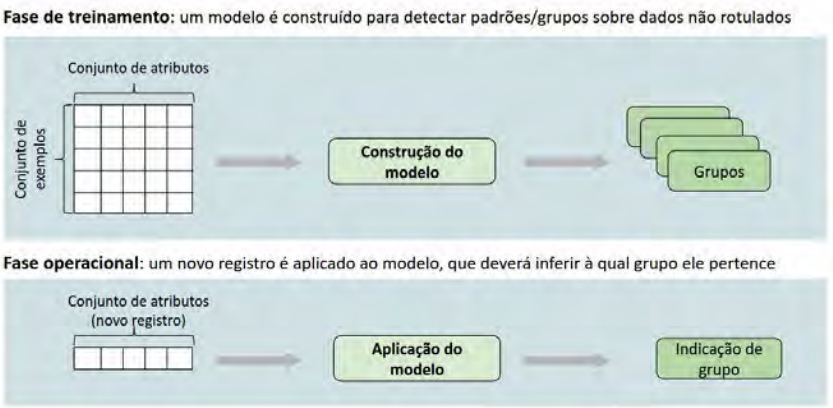

Figura 4.3: Exemplo de fluxo de aprendizado não supervisionado

São exemplos de algoritmos de agrupamento: k-means, fuzzy c-means e redes neurais.

Associações

Essa atividade tem como objetivo identificar afinidades em um conjunto de atributos. Ou seja, avaliar como os atributos estão relacionados, gerando como resultado um conjunto de regras de associação.

Um dos problemas mais conhecido nessa tarefa é a análise do cesto de compras. Como exemplo dessa análise aplicado ao cenário da Big Compras, pode-se citar a análise de itens comprados em uma mesma compra pelos clientes. Como resultado, pode-se obter a seguinte regra de associação: 80% dos clientes que compram leite também compram pão e margarina, sendo o 80% denominado confiança da regra.

Descobrir informações similares a essa pode gerar insights para a organização das prateleiras e definição de itens em promoção, aumentando as chances de vendas casadas.

Descoberta de sequências

Podendo ser utilizado como uma extensão das tarefas de regras de associação, o objetivo das tarefas de descoberta de sequências é também identificar itens frequentes, porém considerando um determinado período de tempo. Ou seja, de acordo com a maneira com que os dados estão alinhados, com essa tarefa pode-se descobrir uma sequência cronológica em que aconteceram os eventos.

Para o cenário da Big Compras, por exemplo, a descoberta de sequências pode revelar que, quando um cliente compra uma cama, ele costuma comprar itens de roupa de cama na sequência. Tal

informação pode ser útil para a realização de campanhas de marketing direcionadas.

Essas são apenas algumas das possibilidades para se obter valor a partir da análise de dados. Empresas brasileiras de diversos setores já estão adotando essas técnicas para obter vantagem competitiva no mercado.

Um exemplo notório é a empresa varejista Magazine Luiza. Com um histórico de dados de clientes desde 1998, a empresa realiza a análise de dados para diversas ações, desde segmentação de clientes, modelagem estatística e ações de comunicação dirigida.

Além dos dados transacionais, a empresa também utiliza informações como dados de navegação, informações sobre presença em loja física e informações de crédito. A análise desses dados permite que a empresa consiga ter maior inferência sobre o comportamento do consumidor, para que assim eles possam ser mais assertivos nas abordagens, realizadas por meio do site da loja, e-mail marketing, mala direta e rede social. Como resultado, a empresa obtém maior satisfação do cliente, maior retorno em vendas e diluição nos investimentos de marketing.

Aplicações específicas de análise de dados

Nessa seção, são abordadas as aplicações específicas atualmente desenvolvidas a partir das técnicas de mineração de dados.

Análise de redes sociais online

Você já utilizou uma rede social online na qual encontrou pessoas conhecidas por meio da sugestão de amigos feitas para você? Imagine se essa funcionalidade não existisse. Certamente seria muito mais difícil encontrar pessoas conhecidas dentro da rede.

Mas você sabe como esse tipo de sugestão é implementada? Isso

é possível por meio de técnicas de análise de redes sociais online.

Esse tipo de análise tem despertado um grande interesse no mundo dos negócios, devido ao crescimento do número de diferentes tipos de interações entre indivíduos e organizações. As técnicas utilizadas nesse tipo de análise são fundamentadas em teoria de grafos, na qual nós representam indivíduos, e vértices representam os inúmeros relacionamentos entre eles.

Por meio dessas técnicas, torna-se possível identificar como grupos foram formados, quais usuários são mais influentes dentro desses grupos, como um usuário seleciona suas conexões, entre inúmeras outras informações.

Um dos desafios relacionados à análise de redes sociais online é que elas costumam ser altamente dinâmicas, na qual mudanças ocorrem a todo o momento. Por isso sua aplicação precisa ter a habilidade para se adaptar e identificar tendências de acordo com essas mudanças.

Processamento de linguagem natural

Imagine que você deseja analisar uma série de vídeos de palestras em inglês, para gerar automaticamente a legenda do áudio do filme em português. Como você implementaria essa funcionalidade?

A área de processamento de linguagem natural (PLN) tem como objetivo analisar e extrair significados de conteúdos de voz e texto. Considerada um subconjunto de *text analytics*, as técnicas de PLN se diferem das demais por terem a habilidade de analisar a linguagem escrita, buscando compreender a estrutura de um texto e o significado de cada palavra dentro de um contexto de outras.

Pense, por exemplo, no Google Tradutor, um serviço do Google que faz a tradução automática de um texto para outro idioma.

Como o computador pode compreender quando a palavra "nada" faz referência a um pronome indefinido ou a flexão do verbo nadar? Para realizar atividades como essa, a área de PLN é composta por um vasto conjunto de técnicas, tais como a análise léxica e extração de palavras-chave.

Essas são algumas aplicações que fazem uso do PLN: tradução de línguas, reconhecimento de voz, classificação de texto em categorias, detecção de plágio, extração de informação, correção gramatical e extração de dados a partir de textos.

Visão computacional

Pense no seguinte desafio: você deseja identificar imagens para identificar se nelas aparece a marca de um determinado produto. Como você faria para fazer essa identificação?

Esse seria um dos problemas inseridos na área de visão computacional. Sabemos que a quantidade de dados a partir de vídeos e imagens cresceu de forma acelerada nos últimos anos. Isso trouxe à tona a possibilidade de descobrir novos insights a partir desses dados.

A área de visão computacional tem como objetivo estudar mecanismos para analisar e compreender a informação visual nesse grande volume de dados de vídeos e imagens. Algoritmos de aprendizado de máquina são utilizados nesse segmento para detecção de objetos, mapeamento de imagens, busca, indexação e recuperação de imagens, compreensão de cenas, entre outras inúmeras pesquisas com esses tipos de dados.

Uma das técnicas que tem ganhado destaque recentemente em visão computacional é a deep learning, um tipo específico de rede neural artificial que tem se destacado nas tarefas de tratamento e reconhecimento de imagens, texto e voz. Deep learning utiliza um

processo de aprendizagem muito similar ao que ocorre no cérebro humano.

Para isso, durante a aprendizagem de reconhecimento de uma imagem, por exemplo, são utilizados um grande volume de informações para que o reconhecimento seja possível. Diversas soluções que usamos atualmente são processadas por meio de deep learning, como por exemplo a funcionalidade de reconhecimento de faces que o Facebook oferece.

São exemplos de aplicações: câmera inteligente, reconhecimento de íris, reconhecimento de sinais de tráfego, análise de conteúdo de vídeos, recuperação de imagens, detecção de pedestre e detecção de objetos.

4.5 VALIDANDO O MODELO

Já ouviu falar que uma informação errada é pior que nenhuma informação? Essa frase também se aplica à análise de dados.

Imagine o caos que pode ser gerado em uma empresa da área médica que utiliza resultados de um modelo que faz diagnósticos errados sobre seus pacientes. Ou então, uma empresa que utiliza um modelo preditivo que reconhece grande parte das transações idôneas como sendo fraudulentas? Ou até mesmo o contrário, que considera muitas transações fraudulentas como sendo idôneas.

Quanto mais serviços forem realizados com base em informações obtidas da análise de dados, maior a importância de se validar os modelos e assim ter resultados mais assertivos. Dessa forma, após ter realizado o tratamento dos dados e construído o modelo de acordo com a análise desejada, deve ser iniciado a fase de validação do modelo.

Essa fase tem como objetivo avaliar o desempenho do modelo

por meio de dados reais, ou seja, dados que não foram utilizados na fase de treinamento. Existem diversas formas para medir a qualidade de um modelo, dependendo da tarefa e do algoritmo adotado. Entre as possibilidades, as mais comuns são:

- Utilização de medidas estatísticas para validar se os dados de treinamento e o modelo foram corretamente utilizados;
- Separação da base de dados em treinamento e teste, permitindo avaliar o desempenho do modelo antes de usá-lo em um ambiente de produção;
- Avaliação perante profissionais especializados em análise de dados e na área de negócio em que o modelo foi aplicado, para que eles possam determinar se a descoberta ou predição foi condizente e significativa.

Para se ter uma medida mais precisa da qualidade do modelo, é muito comum que mais de uma alternativa seja utilizada. Durante esse processo, diversos aspectos sobre os resultados obtidos são validados, sendo mais comuns as verificações da acurácia, confiabilidade e utilidade do modelo.

A medida de acurácia é utilizada para avaliar quão bem o modelo faz a correlação de um resultado com os atributos dos dados de entrada. Já a confiabilidade tem como objetivo avaliar como o modelo é executado em diferentes conjuntos de dados. Caso ele gere a mesma predição ou encontre os mesmos padrões, independente dos dados testados, ele é considerável confiável. Por fim, mas não menos importante, a utilidade do modelo é uma medida que avalia o quanto o modelo oferece informações significativas ao propósito da análise.

Uma técnica existente para validar a acurácia do modelo é a validação cruzada (*cross validation*), muito utilizada em algoritmos

de classificação. Nessa técnica, omite-se uma observação da base de dados durante as iterações, e a função de classificação é realizada com os dados restantes.

Por exemplo, sendo `k = 10` , o classificador será treinado 10 vezes. Na primeira iteração, o grupo 1 é utilizado para teste e o restante para treinamento. Na segunda iteração, o grupo 2 é usado para teste e o restante para treinamento, e assim sucessivamente.

Para cada iteração é calculada a taxa de erro de classificação. E ao fim de todas as iterações, calculam-se a média e o desvio padrão das taxas de erro sobre esses grupos.

4.6 TECNOLOGIAS DE BIG DATA PARA ANÁLISE DE DADOS

Com o advento de Big Data, surgiram novas possibilidades relacionadas à análise de dados. Entretanto, há muitos anos essa prática já é realizada pelas empresas, existindo diversas ferramentas para esse fim.

Por exemplo, *Business Intelligence* (BI) se tornou um termo popular nas áreas de negócios nos anos 90 para referenciar um conjunto de técnicas para o processo de coleta, organização, análise e monitoramento de informações que oferecem suporte a gestão de negócios. Diversas ferramentas analíticas se estabeleceram nessa época, sendo até hoje amplamente utilizadas.

Temos como exemplo o Microsoft Excel, SAS, SPSS, R, Weka e Cognos. Cada uma tinha como objetivo oferecer funcionalidades para aperfeiçoar a inteligência dos negócios.

Com a necessidade de manipular dados referentes aos 3 Vs de Big Data (volume, variedade e velocidade), essas ferramentas passaram a sofrer limitações na análise de dados. Um dos primeiros

desafios foi em relação à variedade dos dados, visto que as ferramentas tradicionais de análise de dados tinham como principal objetivo a análise de dados estruturados.

Os dados eram mantidos em silos em um *data warehouse*, e analisados para identificação de padrões que pudessem auxiliar o processo de tomada de decisão. Entretanto, as empresas que usavam essas ferramentas foram percebendo a necessidade de processar não somente dados estruturados, mas também uma avalanche de dados não estruturados, em formato de textos, vídeos, imagens e outros conteúdos.

O segundo desafio foi em relação ao volume e velocidade dos dados. Ao realizar suas análises com um grande volume de dados, as empresas perceberam que as ferramentas tradicionais não ofereciam o desempenho e escalabilidade necessária para manipular a quantidade de dados desejada. Por consequência, havia uma limitação no tempo de resposta das análises e na quantidade máxima de dados passível de ser utilizada.

Diante dessas limitações, as empresas perceberam a necessidade de se criar uma nova infraestrutura para suportar grandes volumes de dados, e assim gerar as análises de acordo com a necessidade. O que elas passaram a buscar foram soluções que permitissem utilizar as técnicas de análise de dados em ambientes escaláveis, com bom desempenho e baixo custo.

Já vimos no capítulo anterior que existem diversas ferramentas de Big Data que permitem o processamento distribuído de grande volume de dados. Todas as ferramentas citadas são adequadas para o processo de análise de dados. Entretanto, elas estão focadas principalmente em fornecer suporte a questões relativas ao desempenho, escalabilidade, disponibilidade e tolerância a falhas, porém não possuem em seu core um suporte para o desenvolvimento de algoritmos utilizados nas análises.

Para suprir essa necessidade, têm surgido frameworks e bibliotecas de programação específicas para o desenvolvimento de algoritmos de análise de dados que podem ser utilizados em conjunto com as tecnologias de Big Data, facilitando assim a análise dos dados em ambiente escalável. A seguir, apresento algumas dessas ferramentas.

Apache Mahout

O modelo de programação MapReduce pode ser utilizado para desenvolver inúmeras aplicações. Entretanto, a conversão de algoritmos para o modelo chave-valor, com funções map e reduce, pode ser um grande desafio, principalmente se esses algoritmos forem os pertencentes à área de aprendizado de máquina.

Para que a implementação desses algoritmos na plataforma Hadoop não fossem tão complexas, foi criado o Apache Mahout, uma biblioteca Java que oferece a base para diversos algoritmos de mineração de dados e aprendizado de máquina. Com o Mahout, o desenvolvedor tem disponível uma série de implementações prontas para serem usadas em sua análise, devendo apenas configurar o algoritmo com os parâmetros e fluxo desejado.

Tendo feito isso, a execução do algoritmo é feita em uma plataforma Hadoop, permitindo capturar e salvar dados no HDFS, bem como distribuir as tarefas dentro da plataforma. Atualmente, o Apache Mahout já fornece suporte para inúmeros algoritmos de aprendizado supervisionado e não supervisionado.

Spark MLLib

Enquanto que o Apache Mahout oferece algoritmos de mineração de dados e aprendizado de máquina baseados no modelo de programação MapReduce, a biblioteca Spark MLLib oferece esses algoritmos para serem executados no ambiente Spark. Dessa forma,

o desenvolvedor também conta com um conjunto de classes para gerar as análises necessárias para o seu negócio. Essa biblioteca também permite trabalhar com fontes de dados armazenados no HDFS, capturando e gravando novos dados.

Weka

O software Weka foi desenvolvido em 1997, pela Universidade de Waikato (Nova Zelândia), oferecendo à comunidade uma interface gráfica para o desenvolvimento de algoritmos de mineração de dados. Entretanto, o software tem como limitação o fato de utilizar somente o processamento local para suas análises, limitando a capacidade de processamento que usam grandes volumes de dados.

Para se adaptar às necessidades impostas por Big Data, foram desenvolvidas bibliotecas que realizam a integração entre o Weka e os frameworks de Big Data Hadoop e Spark. Dessa forma, o desenvolvedor continua usufruindo da interface amigável que o Weka oferece, mas tem agora a possibilidade de processar sua aplicação em um ambiente distribuído.

R

R é uma linguagem de programação e um ambiente de software gratuito com funcionalidades voltadas à computação estatística e à visualização de dados. Um dos destaques do R é a facilidade na manipulação dos dados e a vasta quantidade de bibliotecas em seu repositório, contendo funções para diferentes análises, como mineração de texto, redes Bayesianas, agrupamento, classificação e análise de séries temporais.

Uma das limitações do R é o gerenciamento de memória. Por manter todos os objetos em memória, você pode ter limitações na quantidade de dados processados durante a análise, sendo isso um

grande problema principalmente na era de Big Data. Para solucionar essa questão, atualmente existe a possibilidade de integrar o software R com outros frameworks de Big Data, como o Hadoop e Spark.

Python

Diferente de R, Python é uma linguagem de programação de alto nível voltada para um propósito geral, não somente análise de dados. Entretanto, algumas bibliotecas Python específicas para esse propósito estão tornando essa linguagem muito atrativa para os analistas de dados, principalmente aos que possuem maior background em programação do que em estatística.

Uma das bibliotecas que tem se destacado é o *scikit-learn*, contendo inúmeras ferramentas voltadas à mineração de dados e ao aprendizado de máquina. Além dessa, temos também o *Pandas*, outra biblioteca open source com funcionalidades para estrutura e análise de dados.

Tanto R quanto Python são poderosos na quantidade de funções que disponibilizam para a análise de dados. Por esse motivo, a escolha entre as duas linguagens pode ser feita de acordo com a que você sentir mais facilidade.

4.7 BIG DATA ANALYTICS

Com a quantidade massiva de dados gerados atualmente, novos desafios foram surgindo à análise de dados. Para possibilitar o avanço das análises realizadas, métodos estatísticos, algoritmos de aprendizado de máquina e técnicas de mineração de dados precisaram ser adaptados para suportar modelos de processamento paralelo e distribuídos. Dessa forma, as tecnologias de processamento de Big Data, somadas à evolução dessas abordagens,

culminaram em formas inovadoras de obtenção de insights sobre dados.

Técnicas tradicionais de análise de dados passaram a ser usadas em soluções adaptadas para o grande volume, variedade e velocidade dos dados. Esse novo cenário também impactou diretamente a área de analytics, um termo utilizado para descrever as técnicas e metodologias focadas em transformar dados em informação, principalmente para apoio no processo de tomada de decisão.

Com a junção das técnicas tradicionais de analytics com as tecnologias de Big Data, surgiu o termo *big data analytics*. Ele tem como foco a extração de informação a partir de uma avalanche de dados.

Atualmente, as técnicas de analytics podem ser divididas em 4 diferentes categorias, que se distinguem pelos resultados que elas produzem e pelos algoritmos e técnicas usados. São elas:

- análise descritiva;
- análise diagnóstica;
- análise preditiva;
- análise prescritiva.

Confira a seguir uma descrição de cada uma dessas categorias.

Análise descritiva

Imagine que você tenha uma base de dados de um e-commerce com 3 milhões de clientes e os dados de compras realizadas por eles nos últimos 5 anos. Para contextualizar esses dados, é necessário responder a perguntas como:

- Quais são os clientes que mais compraram?

- Qual a periodicidade de compra desses clientes?
- Qual produto foi mais vendido em cada trimestre?
- Qual foi o volume de vendas mensal nos últimos anos?

Considerada a técnica mais adotada pelas empresas, a análise descritiva refere-se a forma mais básica de se obter indicadores como esses mencionados para análise da situação de uma empresa. Nesse tipo de análise, o objetivo principal é responder a seguinte pergunta: "o que aconteceu?". Para isso, indicadores são gerados a partir de dados históricos da organização, permitindo ao gestor ter uma visão das ocorrências passadas da empresa.

Na análise descritiva, utiliza-se normalmente *dashboards* e relatórios ad-hoc, que apresentam geralmente as informações de forma estática, obtidas a partir dos dados históricos. Nesse tipo de análise, é comum a utilização de técnicas como agrupamento, regras de associação e detecção de anomalias, para assim obter uma descrição mais detalhada dos dados analisados.

O objetivo principal das informações coletadas na análise descritiva é então sumarizar os dados passados, obtendo informações como a quantidade de vendas por categoria, média de vendas, índice de aumento de novos clientes, índice de cancelamentos e quantidade de produtos em estoque.

Estima-se que mais de 80% das análises realizadas nas empresas são descritivas. Embora ofereça informações valiosas, que permitem aos gestores terem uma percepção mais detalhada da empresa, a análise descritiva não fornece meios para a automatização do processo de tomada de decisão, necessitando da total intercepção humana.

Análise diagnóstica

Uma vez que a análise descritiva tenha fornecido informações que nos permitem entender "o que aconteceu", novas informações podem ser obtidas por meio da análise diagnóstica. Ela tem como foco responder à questão "porque isso aconteceu?".

- Porque tivemos um aumento das vendas no primeiro semestre?
- Porque a filial X teve maior índice de inadimplência que as demais filiais?
- Porque a produção dos equipamentos diminuiu no último trimestre?

Perceba que, na análise diagnóstica, a pergunta a ser respondida ainda está relacionada à análise de dados históricos. Entretanto, diferente da descritiva, a análise diagnóstica busca identificar informações que estão relacionadas aos fenômenos ocorridos na empresa. Para obter esse conhecimento, as duas técnicas de analytics, descritiva e diagnóstica, devem ser utilizadas.

Uma técnica adotada na diagnóstica é o gráfico de controle. Dessa forma, uma empresa que teve um declínio das vendas pode identificar, por exemplo, que a falta do produto X foi o principal responsável por essa ocorrência. Tal informação permite a tomada de decisão bem direcionada à causa raiz do problema.

Diferente da análise descritiva, o resultado da diagnóstica normalmente é apresentado em ferramentas de visualizações interativas, facilitando a identificação de padrões e tendências pelos usuários. Similar à descritiva, ela requer a interferência humana no processo de tomada de decisão, uma vez que essa técnica também não fornece indicativos de ocorrências futuras.

Análise preditiva

Até agora foram apresentadas as características de duas técnicas

de analytics: a descritiva e a diagnóstica. Vimos que cada uma delas visa responder a uma determinada pergunta, porém ambas têm como foco entender o que ocorreu no passado.

Essas informações são muito valiosas para os gestores de uma empresa. Entretanto, e se além de identificar, por exemplo, qual foi o montante de vendas no último trimestre, fosse possível responder às seguintes questões:

- Qual será a demanda necessária para o mês seguinte?
- Qual a probabilidade de um cliente aceitar um cupom de desconto e efetivar uma compra?
- Qual a estimativa de venda para os próximos meses?

Para obter respostas a essas perguntas, utiliza-se a análise preditiva. Podemos considerar essa categoria um divisor de águas entre os 4 tipos de analytics, uma vez que ela permite não somente compreender o passado, mas também oferece a habilidade de obter informações sobre "o que pode acontecer" no futuro, tanto em relação aos riscos como também oportunidades.

Sabemos que não há como saber exatamente o que vai acontecer no futuro. Porém, por meio de mecanismos de aprendizagem de máquina e técnicas estatísticas, podem-se identificar padrões, tendências e exceções existentes nos dados históricos e, a partir daí, criar um modelo que permita fazer predições de eventos futuros.

A análise preditiva é mais complexa do que a descritiva e a diagnóstica. Elas exigem o uso de grandes conjuntos de dados históricos para permitir assim prever a classe de um conjunto de observações, baseando-se na similaridade de observações classificadas no passado.

Análise prescritiva

A evolução das soluções de Big Data trouxe um novo nível de inteligência às aplicações. Atualmente, além de se obter soluções em tempo real para agir prontamente sobre os resultados, têm surgido mecanismos capazes de identificar previamente um problema e sugerir ações estratégicas para resolvê-los.

Essa capacidade de sugerir ações que se beneficiem das predições é a característica da análise prescritiva. Enquanto a preditiva tem como foco responder à questão: "o que vai acontecer?", a prescritiva vai além dessa inteligência, identificando os meios necessários para saber "como fazer acontecer".

Dessa maneira, conseguimos respostas para questões como as demonstradas a seguir:

- Qual procedimento adotar ao perceber uma tendência no aumento das vendas?
- Quais medidas a serem tomadas para produzir os produtos da empresa no tempo e custo desejado?

Para gerar essas respostas, os algoritmos usados na análise prescritiva são programados com um mínimo de intervenção humana nas regras do algoritmo. O propósito aqui é que o algoritmo seja capaz de se adaptar de acordo com os parâmetros recebidos por ele, de forma que sua capacidade de predição e otimização seja feita automaticamente.

O carro autônomo do Google é um grande exemplo de uma solução que utiliza análise prescritiva. O próprio algoritmo é capaz de tomar decisões sobre a direção do veículo baseado nas inúmeras informações recebidas do mundo externo, como geolocalização, identificação de pedestre, informação do semáforo, aspecto do solo etc.

Para ser capaz de tomar essas decisões automaticamente, a

análise prescritiva necessita de enormes bases de dados para o processo de aprendizado. Felizmente, Big Data tem facilitado a aquisição deles.

Com esse exemplo, já é possível identificar que a análise prescritiva é uma solução disruptiva, capaz de transformar negócios, tornando o processo de tomada de decisão cada vez mais eficiente. Entretanto, por ser uma técnica ainda complexa de ser adotada, há uma estimativa de que apenas 3% das empresas usam análise prescritiva em seus negócios. Ou seja, ainda há muitas possibilidades de obter vantagem competitiva por meio dessa técnica de análise.

Resumo das categorias de analytics

Podemos observar na figura a seguir claramente quais questões são respondidas por cada categoria de analytics. Uma empresa que utiliza essas quatro categorias tem a capacidade de tomar decisões apoiada por dados, obtendo percepções claras sobre a real situação de seu negócio.

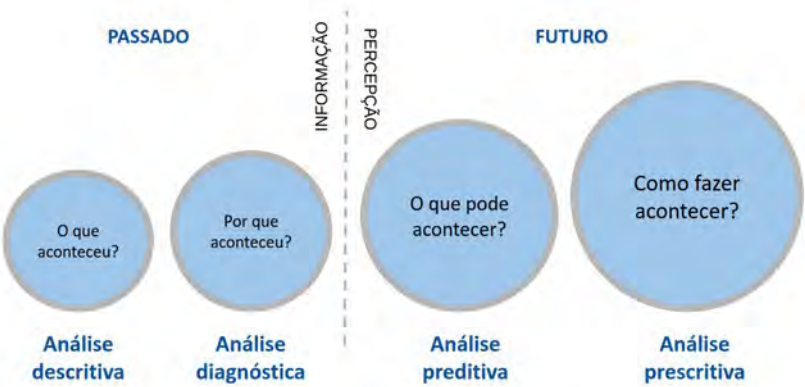

Figura 4.4: Quatro níveis de analytics

Certamente veremos nos próximos anos a iniciativa cada vez

maior de empresas aperfeiçoando suas análises de formas inteligentes e criativas. Aliado a estratégias de Big Data, este mercado está cada vez mais promissor, de tal forma que a IDC prevê que, até 2020, 50% das aplicações de analytics já terão incluído a análise prescritiva. Sua empresa está preparada para essa adoção?

4.8 PRATICANDO: CLASSIFICAÇÃO DE MENSAGENS USANDO R

Nessa atividade, aprenderemos como criar um classificador para filtrar mensagens como sendo "spam" ou "não spam". Para concluir a atividade, devemos realizar a seguinte sequência de passos:

- **Passo 1:** tratamento da base de dados;
- **Passo 2:** construção do modelo;
- **Passo 3:** teste e verificação do modelo.

Para executarmos esses passos, utilizaremos o software R (https://www.r-project.org/). Nessa atividade, foi usada a versão 3.3.1. Estamos partindo do princípio de que você já tenha um ambiente com o R com o ambiente RStudio funcionando corretamente. Caso ainda não tenha, você pode instalar o RStudio seguindo o tutorial disponível em:

https://www.rstudio.com/products/rstudio/download/

Para realizar essa atividade, utilizaremos uma base de dados de mensagens `msg`, disponibilizada pelo seguinte link: https://github.com/rosangelapereira/livrobigdata/tree/master/cap4/input. Essa base é composta por 5.559 mensagens já rotuladas como spam ou ham (nome dado para mensagens não spam).

O classificador que criaremos atribui uma probabilidade de uma nova mensagem estar em uma classe (spam) ou em outra (ham). A partir das palavras que estão e que não estão na mensagem, essa

técnica calcula a probabilidade de spam ou não spam para cada palavra. Essa técnica é baseada na regra de Bayes e da análise de frequência de ocorrências de palavras.

Passo 1: tratamento da base de dados

O primeiro passo é carregar a base de dados para o R, conforme o comando a seguir:

```
> msg <- read.csv(file="mensagens.csv", stringsAsFactors=F)
```

Podemos então verificar o conteúdo da variável `msg` utilizando o comando `str`:

```
> str(msg)
'data.frame': 5559 obs. of 2 variables:
$ type: chr "ham" "ham" "ham" "spam" ...
$ text: chr "Hope you are having a good week.
Just checking in" "K..give back my thanks."
"Am also doing in cbe only. But have to pay."
"complimentary 4 STAR Ibiza Holiday or £10,000
cash needs your URGENT collection. 09066364349
NOW from Landline not to lose out"
__truncated__ ...
```

Perceba que a base foi devidamente carregada, contendo 5.559 observações. Perceba também que o texto das mensagens está em seu formato original, precisando de ajustes para ser utilizado no modelo.

Para aplicarmos os tratamentos necessários à base de dados, usaremos o pacote `tm` do R, específico para técnicas de mineração de texto. Isso pode ser feito por meio do seguinte comando:

```
> install.packages("tm")
> library(tm)
```

O primeiro comando é utilizado para instalar o pacote, e o segundo para carregar o pacote na seção. O próximo passo para usarmos as operações de mineração de texto será criarmos uma

coleção de documentos. Essa coleção é tecnicamente referenciada como um objeto Corpus no software R.

```
> msg_corpus <- Corpus(VectorSource(msg$text))

#verificando o conteúdo da variável
print(msg_corpus)
<<VCorpus>>
Metadata: corpus specific: 0, document level (indexed): 0
Content: documents: 5559
```

Com esse objeto, podemos dar início ao tratamento dos dados. Utilizaremos uma função chamada `content_transformer`, que aplica a transformação desejada em uma base de texto. A primeira atividade que realizaremos será transformar todas as palavras da base em minúsculo, usando a função `tolower`.

```
> ccorpus <- tm_map(msg_corpus, content_transformer(tolower))
```

Também removeremos os números da base de dados, uma vez que os números não auxiliam na classificação de uma mensagem. Nessa transformação, será utilizada a função `removeNumbers`.

```
> ccorpus <- tm_map(ccorpus, content_transformer(removeNumbers))
```

A próxima transformação será remover as pontuações da base de dados, para que elas também não influenciem o classificador. Nesse caso, é usada a função `removePunctuation`.

```
> ccorpus <- tm_map(ccorpus,
content_transformer(removePunctuation))
```

Até agora já conseguimos fazer uma boa lapidação da base de dados. Porém, mesmo agora com a base contendo apenas palavras, sem pontuações e números, precisamos também remover palavras que não são significativas para a classificação de mensagens, como: *me*, *my*, *she* e *he*. Essas palavras são chamadas no R de *stopwords* e podem ser verificadas com o seguinte comando:

```
> stopwords()
[1] "i" "me" "my" [4] "myself" "we" "our"
```

```
[7] "ours" "ourselves" "you"
[10] "your" "yours" "yourself"
[13] "yourselves" "he" "him"
[16] "his" "himself" "she"
```

Para remover essas palavras, utilizamos a função `removeWords`, conforme o código a seguir.

```
> ccorpus <- tm_map(ccorpus,
content_transformer(removeWords), stopwords())
```

Vamos verificar como ficou nosso objeto Corpus após essas transformações?

```
> inspect(ccorpus[1:4])
Metadata: corpus specific: 0, document level (indexed): 0
Content: documents: 4
[[1]]
<<PlainTextDocument>>
Metadata: 7
Content: chars: 33
```

Após a limpeza de dados, estamos prontos para utilizar a base na construção do modelo. Entretanto, ainda precisamos criar um objeto chamado matriz de termos em documentos, que será usado para contar a frequência de cada palavra nas mensagens observadas. Podemos fazer isso com o seguinte comando:

```
> ccorpus <-
Corpus(VectorSource(ccorpus))

> msg_mtd <- DocumentTermMatrix(ccorpus)

#verificando a matriz
> inspect(msg_mtd[1:4, 30:35])
<<DocumentTermMatrix (documents: 4, terms: 6)>>
Non-/sparse entries: 0/24
Sparsity
: 100%
Maximal term length: 8
Weighting
: term frequency (tf)
Terms
Docs abstract abt abta aburo abuse abusers
1    0 0 0 0 0 0
```

```
2    0 0 0 0 0 0
3    0 0 0 0 0 0
4    0 0 0 0 0 0
```

Estamos quase prontos para gerar o classificador, falta apenas separarmos nossa base de dados para a fase de treinamento e para a fase de teste, conforme vimos na explicação sobre a tarefa de classificação. Nessa atividade, deixaremos 75% da base de dados para o treinamento e os 25% restante para o teste.

Para isso, primeiramente vamos separar os índices referentes às mensagens spam e os índices referentes às mensagens ham:

```
> spam_indices <- which(msg$type == "spam")

#verificando o conteúdo dos 3 primeiros registros
> spam_indices[1:3]
[1] 4 5 9

> ham_indices <- which(msg$type == "ham")

#verificando o conteúdo dos 3 primeiros registros
> ham_indices[1:3]
[1] 1 2 3
```

Você se lembra de que, até o momento, temos 3 variáveis criadas: `msg` , referente à base de dados original; `ccorpus` , referente ao nosso objeto Corpus; e `msg_mtd` , referente à matriz de termos de documentos. Pois bem, nossa divisão da base de dados deverá ocorrer para elas.

```
> msg_trei <- msg[1:4169,]
> msg_test <- msg[4170:5559,]
> msg_mtd_trei <- msg_mtd[1:4169,]
> msg_mtd_test <- msg_mtd[4170:5559,]
> msg_cp_trei <- ccorpus[1:4169]
> msg_cp_test <- ccorpus[4170:5559]
```

Agora que temos nossa base de treinamento, podemos dividi-la em outras duas partes: base com mensagens spam e base com mensagens ham.

```
> spam <- subset(msg_trei, type == "spam")
> ham <- subset(msg_trei, type == "ham")
```

Está lembrado de que nossa matriz de termos de documentos armazena a frequência com que cada palavra apareceu na base de dados? Como as palavras que apareceram poucas vezes podem ter pouca influência no modelo, vamos removê-las da matriz.

```
> baixa_freq <- findFreqTerms(msg_mtd_trei,5)

> msg_trein <-
DocumentTermMatrix(msg_cp_trei,
control=list(dictionary = baixa_freq))

> msg_test <- DocumentTermMatrix(msg_cp_test,
      control = list(dictionary = baixa_freq))
```

Um próximo ajuste que precisamos fazer é alterar os valores da frequência de palavras em nossa matriz para um valor booleano. Isso é necessário pelo fato de que o classificador Bayesiano atua de forma binária. Nossa estratégia será criar uma função que deixe a matriz com valores 0 e 1.

```
> conversao <- function(x) {
    y <- ifelse(x > 0, 1,0)
    y <- factor(y, levels=c(0,1), labels=c("nao", "sim"))
    y
}
```

Utilizamos então essa função para atualizar as matrizes:

```
> msg_trein <- apply(msg_trein, 2, conversao)

> msg_test <- apply(msg_test, 2, conversao)
```

Ufa! Quanto trabalho para preparar os dados para gerar o classificador. Lembra-se das premissas de análise de dados? A preparação dos dados costuma ser a fase mais demorada do processo de análise. Mas agora estamos prontos para a construção do modelo.

Passo 2: construção do modelo

Para a criação de um classificador Bayesiano, usaremos a função `naiveBayes`, oferecida pelo pacote `e1071` do R. Para isso, devemos primeiramente carregar esse pacote.

```
> install.packages("e1071")

> library(e1071)
```

Feito isso, podemos criar o classificador pelo seguinte comando:

```
> classificador <- naiveBayes(msg_trein, factor(msg_trei$type))
```

Passo 3: teste e verificação do modelo

Estando o classificador criado, podemos utilizá-lo com nossa base de teste para verificar se ele conseguirá fazer a classificação corretamente. Isso é feito usando a função `predict`. Lembre-se de que, na base de teste, os dados enviados ao modelo não estão mais rotulados.

```
> msg_test_pred <- predict(classificador, newdata = msg_test)
```

Como será que foi o desempenho do classificador? Podemos fazer essa verificação gerando a tabela que apresenta o resultado da classificação.

```
msg_test_pred          ham     spam
                ham    1202    31
                spam    5      152
```

Opa, tivemos um bom resultado do classificador. De todas as mensagens do tipo ham, ele classificou 1202 corretamente, sendo que apenas 31 foram classificadas como spam (chamados de falso negativo). De todas as mensagens do tipo spam, ele classificou 152 mensagens corretamente, sendo que apenas 5 foi classificada como sendo ham (chamados de falso positivos).

É muito difícil termos um modelo que consiga prever corretamente 100% de novas instâncias, mas podemos tentar novas

abordagens no tratamento dos dados para verificar se aumentamos a acurácia do modelo. Também podemos tentar outros algoritmos, para avaliar qual apresenta o desempenho mais adequado.

4.9 CONSIDERAÇÕES

Quando Big Data ainda era usado por pouquíssimas empresas, tinha-se como um pensamento comum que sua utilização era uma oportunidade para se obter vantagem competitiva. Com a evolução desse conceito nos últimos anos, aliado ao fato que mais e mais empresas passaram a adotar Big Data em diferentes cenários, a sentença sobre a adoção de Big Data também mudou.

O que é dito atualmente é que as empresas precisam utilizar Big Data para se manterem competitivas no mercado. Dessa forma, destaca-se a empresa que estiver mais bem preparada para não somente capturar grande volume de dados, mas também utilizá-los para gerar produtos e serviços a partir deles. Por esse motivo, a análise de dados se tornou um processo tão importante para as empresas atualmente.

Vimos neste capítulo as diferentes estratégias para realizar a análise de dados. É importante ter o conhecimento da existência dessas técnicas, saber o objetivo e característica de cada uma delas, para que o analista saiba identificar qual melhor se enquadra em seu contexto.

Os resultados dessas análises são normalmente apresentados em uma estrutura que muitas vezes é legível somente para o analista. Entretanto, para transmitir o resultado a uma audiência, um meio eficaz é o uso de técnicas de visualização de dados, que serão apresentadas no próximo capítulo.

Neste capítulo, foram abordados tópicos para nos ajudar a

responder às seguintes perguntas:

- Quais são as etapas existentes no processo de análise de dados?
- Quais técnicas posso utilizar para analisar diferentes conjuntos de dados?
- Qual nível de informação posso obter a partir das análises realizadas?
- Quais ferramentas usar para a análise de grande volume de dados?

Para saber mais

1. CONWAY, Drew; WHITE, John. *Machine learning for hackers.* O'Reilly Media, Inc., 2012.
2. KRISHNAN, Krish; ROGERS, Shawn P. *Social Data Analytics:* Collaboration for the Enterprise. Newnes, 2014.
3. LAROSE, Daniel T.; LAROSE, Chantal D. *Data mining and predictive analytics.* John Wiley & Sons, 2015.
4. MCCUE, Colleen. *Data mining and predictive analysis:* Intelligence gathering and crime analysis. Butterworth-Heinemann, 2014.
5. SHEIKH, Nauman. *Implementing analytics:* a blueprint for design, development, and adoption. Newnes, 2013.
6. WU, Xindong; KUMAR, Vipin; QUINLAN, J. Ross; et al. *Top 10 algorithms in data mining.* Knowledge and information systems, v. 14, p. 1–37, 2008. Disponível em: http://www.cs.uvm.edu/~icdm/algorithms/10Algorithms-08.pdf.

7. ZACHARSKI, Ron. *A Programmer's Guide to Data Mining.* 2015. Disponível em: http://guidetodatamining.com/.

Capítulo 5

VISUALIZANDO OS DADOS

"Tudo deveria se tornar o mais simples possível, mas não simplificado." — Albert Einsten

Estamos chegando à etapa final do projeto de Big Data. Capturamos os dados, os armazenamos em uma infraestrutura escalável que suporta grande volume de dados não estruturados, utilizamos tecnologias de Big Data para processá-los e técnicas de estatística, mineração de dados e aprendizado de máquina para analisá-los.

Ufa! Quantos empecilhos e questões foram necessários resolver para chegar até aqui no decorrer de um projeto. Entretanto, uma vez que os dados foram analisados, ainda resta uma última etapa a ser cumprida: a utilização dos dados como meio de gerar valor para a empresa.

Um projeto de Big Data que tem como foco o aumento da percepção deve conter uma visualização de dados capaz de alavancar esse objetivo. Assim como ocorre nas etapas anteriores, a etapa de visualização de dados também exige o uso de novas soluções, oferecendo recursos que simplificam o processo de visualização, bem como mecanismos que enriquecem a experiência dos usuários durante a visualização dos dados.

O volume e variedade dos dados trouxe uma maior complexidade para o processo de compreensão dos dados

analisados, tornando a visualização uma abordagem cada vez mais essencial em um projeto de Big Data. Neste capítulo, é apresentada a importância da visualização de dados atualmente, demonstrando suas características, exemplos de soluções e tecnologias habilitadoras.

5.1 O QUE É VISUALIZAÇÃO DE DADOS

Considere o cenário em que uma análise foi realizada em um determinado conjunto de dados, e as observações obtidas a partir dessa análise precisam ser repassadas para outras pessoas. Essa exposição dos resultados pode ocorrer de inúmeras formas.

Por exemplo, é possível apresentar os resultados usando uma planilha eletrônica, com dados em formato tabular. Também é possível gerar um relatório, reportando textualmente os resultados observados. Como uma terceira alternativa, é possível fazer uma apresentação oral dos resultados obtidos.

Mas será que essas alternativas são as mais eficazes? Você consegue perceber algum problema que pode ocorrer em alguma delas?

Um dos possíveis problemas com a apresentação dos dados em formato tabular é a dificuldade que nós, humanos, temos para fazer comparações a partir desse formato. Isso torna a assimilação dos resultados um processo árduo, especialmente se a quantidade de itens for alta.

Com a utilização de um relatório, somente o relato textual dos resultados da análise, é possível que as observações não sejam apresentadas em sua completude. Isso pode gerar dúvidas ao leitor em alguns aspectos dos dados não relatados.

Por fim, uma apresentação oral dos resultados pode fazer com

que as pessoas tenham conclusões distintas do que foi apresentado, de acordo com o seu nível de compreensão da mensagem do orador. Isso nos faz concluir que, embora sejam muito utilizadas, essas alternativas podem gerar problemas referentes à compreensão do leitor sobre os dados apresentados.

Para evitar a ocorrência de problemas como os citados, a visualização de dados pode desempenhar um papel essencial como suporte à transmissão adequada da mensagem. Mas o que vem a ser visualização de dados?

No livro *Interactive data visualization: foundations, techniques, and applications*, Ward, Keim e Grinstein definem visualização como sendo *"a comunicação da informação utilizando representações gráficas"*. Ou seja, utilizamos meios visuais para comunicarmos algo para outras pessoas, assim como acontece em fotografias, pinturas e filmes, por exemplo.

Quando falamos especificamente sobre visualização de dados, representações gráficas são utilizadas como mecanismos para oferecer uma maior compreensão do que os dados representam. Você já deve ter lido ou escutado em algum momento que "*uma imagem vale mais do que mil palavras*".

Essa frase é de fato verdadeira pelo fato de nós, humanos, termos em nosso cérebro uma incrível capacidade para compreender padrões por meio do sentido visual. Esse potencial é tanto que, de toda a informação sensorial que temos, 80% é destinada a esse sentido.

E sabe por que conseguimos assimilar rapidamente uma imagem quando a vemos? Porque nosso cérebro atua igual ao framework Hadoop que vimos anteriormente: de forma paralela. Nossos olhos e memória dividem as tarefas de assimilação de uma imagem em nosso cérebro, o que faz com que consigamos

interpretar algo rapidamente. Por termos essa capacidade, a visualização de dados permite que nós tenhamos uma compreensão aperfeiçoada do que nos é apresentado, por meio de uma perspectiva totalmente diferente de todas as outras estratégias que apresentei anteriormente.

No entanto, é importante lembrar de que, embora uma imagem valha mais do que mil palavras, caso a visualização de dados não esteja apresentada de forma clara, nosso cérebro não é capaz de identificar padrões de forma tão eficaz. Assim, nosso sentido visual perde seu poder.

Por esse motivo, é essencial definir desde o início qual será o propósito da visualização. É essa definição que direcionará os próximos passos para uma visualização de sucesso. Como veremos a seguir, um dos primeiros passos para essa definição é identificar se o propósito da visualização é para fins de exploração ou explanação/explicação dos dados.

Visualização exploratória

Durante o processo de análise, os dados precisam ser avaliados com minuciosidade, de forma que o analista tenha uma visão detalhada sobre eles, e assim possa decidir quais operações realizar com os dados.

A visualização de dados auxilia muito nesse processo, pois nos permite "conhecer" melhor os dados. Ela oferece uma nova perspectiva, facilitando a identificação da estrutura das variáveis, de tendências, de relacionamentos e até mesmo da existência de anomalias, que por vezes passam despercebidas em uma tabela, principalmente se esta possui milhares de linhas.

Perceba que a visualização de dados exploratória é muito utilizada na fase de análise de dados, apresentada no capítulo

anterior. Nesse estágio, o analista dos dados não está preocupado em identificar se o que é apresentado visualmente será compreensível por outras pessoas. Ele precisa apenas de meios que forneçam descobertas para que ele possa explorar a estrutura dos dados que estão sendo analisados.

Diferentes formas de representações gráficas podem ser usadas durante a fase de exploração dos dados. Apenas para exemplificar, na figura a seguir são apresentados dois exemplos de gráficos muito utilizados durante essa fase: o histograma e o diagrama de caixa (*boxplot*).

O histograma é um tipo de gráfico muito utilizado para visualizar como os dados estão distribuídos, pois ele representa a frequência de ocorrências individuais subdivididas em classes. Por outro lado, o diagrama de caixa (boxplot) é útil para a identificação de anomalias e para fazer uma comparação visual entre dois ou mais grupos.

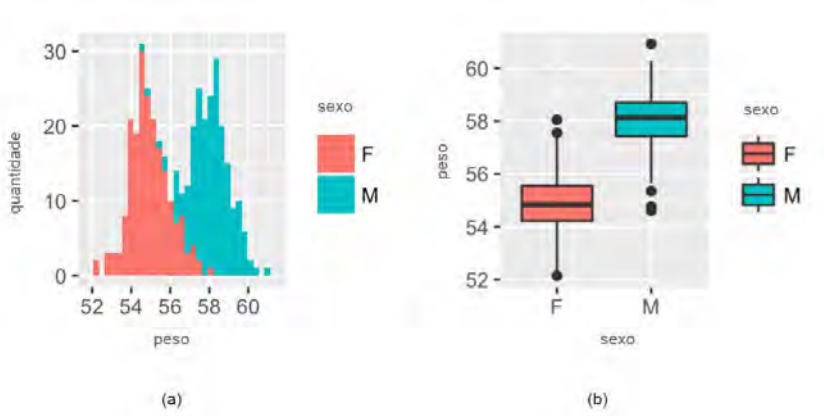

Figura 5.1: Exemplo do gráfico histograma (a) e gráfico de caixa (b)

O histograma e o gráfico de caixa são apenas dois exemplos de gráficos utilizados na fase exploratória. Na verdade, durante essa fase, é comum a geração de inúmeros tipos de gráficos, como o

gráfico de dispersão, de barras e de linhas. Além disso, é comum que o analista gere os gráficos diversas vezes, alterando os valores conforme for estruturando os dados da maneira desejada.

Dado o propósito da visualização de dados na análise exploratória, percebemos que não é necessário nesse estágio um refinamento visual dos gráficos. Isso porque quem está em contato com a visualização gerada é a própria pessoa que a criou e, por esse motivo, espera-se que ela tenha maior facilidade de avaliar o que está sendo apresentado.

Dessa forma, o propósito maior está na rapidez de geração dos gráficos, permitindo acelerar o processo da análise. Por esse motivo, os gráficos gerados nessa fase não requerem o auxílio de um profissional de design gráfico, de forma que o próprio analista pode criar suas próprias visualizações. A necessidade de refinamento surge no momento que a visualização é apresentada para terceiros, ou seja, pessoas que conhecem pouco, ou até mesmo nada, sobre os dados, e precisam de recursos adicionais para interpretarem o que está sendo apresentado.

Visualização explanatória

Uma vez que os resultados obtidos são considerados corretos pelo analista dos dados e ele já compreenda o que os dados estão representando, ele está pronto para demonstrar os resultados obtidos para um grupo de pessoas. Inicia-se nesse momento a fase de explanação dos dados.

Essa seria a fase, por exemplo, em que a equipe da Big Compras apresentaria as percepções obtidas das análises realizadas a partir das inúmeras bases de dados apresentadas no *Capítulo 2 — Capturando e armazenando os dados*. Conforme apresentado na figura a seguir, diferente da visualização exploratória (onde a relação ocorre entre a base de dados e o analista que deseja saber

mais sobre ela), a relação na visualização explanatória ocorre entre o analista e um público diverso, interessado nos resultados obtidos.

Nesse momento, o objetivo não é mais fazer a descoberta dos dados, mas sim enfatizar de forma eficaz o que já foi descoberto, buscando facilitar a compreensão das informações por pessoas que não participaram do processo de análise.

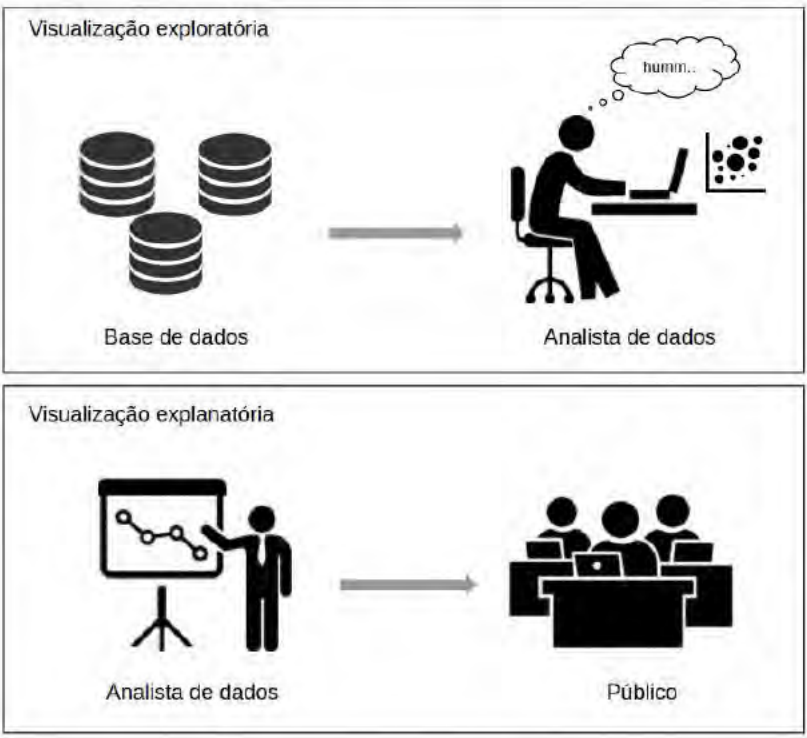

Figura 5.2: Diferença entre a visualização exploratória e explanatória

Para o propósito de comunicação, a visualização de dados oferece os seguintes benefícios:

- Melhor comunicação, pois permite ao analista apresentar de modo efetivo os resultados e percepções

obtidas das análises, e que os usuários identifiquem facilmente os resultados das análises obtidas;
- Melhor monitoramento do desempenho da empresa, pois permite resumir os resultados obtidos de um grande volume de dados em informações mais concisas;
- Apoio no processo de tomada de decisão, pois permite revelar tendências e desvios de tendências que não são facilmente perceptíveis em tabelas e relatórios.

Fica evidente, dessa forma, que a visualização de dados pode resultar em eficiência, otimizando o tempo, facilitando a colaboração e fornecendo suporte ao processo de tomada de decisão. Além disso, se considerarmos o crescente aumento do número de dados que precisam ser analisados ultimamente nos negócios, podemos perceber o quão poderosa uma visualização de dados eficaz pode se tornar. Mas como criar visualizações de dados efetivas?

5.2 CRIANDO AS INTERFACES VISUAIS

Após passar por todo o processo de aquisição e preparação dos dados, é chegada a hora de gerar a interface visual para representá-los. Nesse momento, esteja preparado para lidar com aspectos como: formas, cores, posição, orientação, tamanho, área, saturação, luminosidade, transparência, textura, rótulo e movimento. São esses aspectos que vão nos permitir induzir o leitor a visualizar um ponto específico dos dados, bem como facilitará a apresentação do relacionamento entre diversas variáveis do nosso conjunto de dados.

Para nos auxiliar nesse processo de gerar uma visualização que transmita rapidamente a informação desejada, Colin Ware, diretor do laboratório de pesquisa de visualização de dados da Universidade

de New Hampshire, indicou em seu livro *Information Visualization: Perception for Design*, alguns atributos que imediatamente prendem nosso olhar quando olhamos em uma visualização. Conforme apresentado na figura a seguir, esses atributos estão divididos em três categorias: forma, cor e posição espacial.

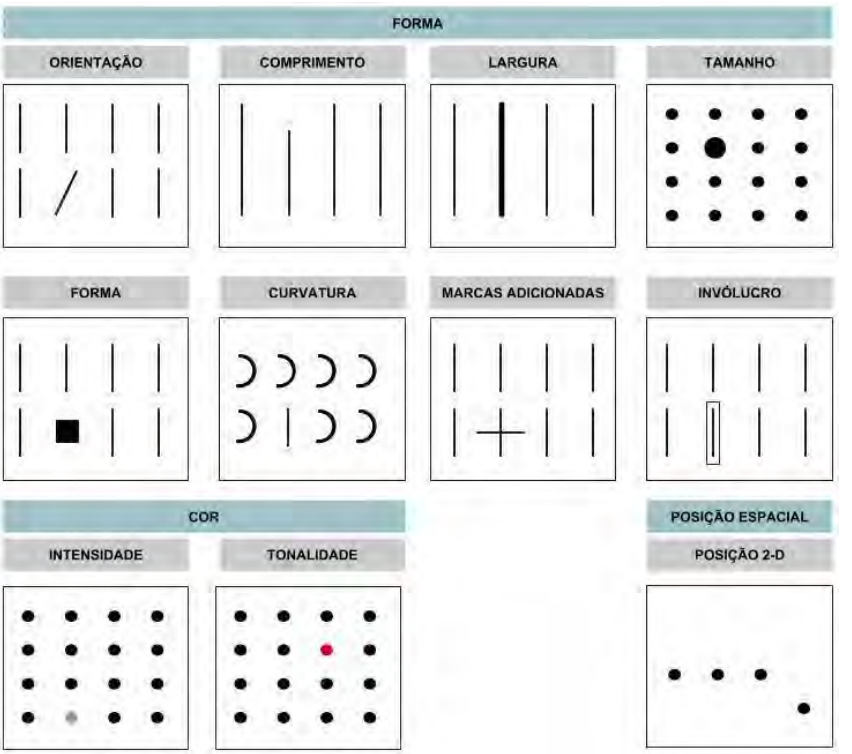

Figura 5.3: Atributos para visualização de dados

A escolha de cada um desses atributos é que definirá se a visualização será eficaz ou não. Por exemplo, imagine se o mapa que apresenta as linhas do metrô não apresentasse distinção de cores em cada linha? Ou então se ele apresentasse inúmeras curvas durante o trajeto de uma linha. Com certeza a identificação visual de uma rota nesse mapa seria prejudicada.

Por esse motivo, é importante que uma equipe envolvida no projeto de visualização de dados conte com profissionais com conhecimento em design gráfico, que tenham a percepção de como esses aspectos podem ser mais bem utilizados para representar uma informação.

Mas além dessas escolhas, temos também de decidir qual gráfico melhor representa a informação que desejamos passar, uma vez que cada gráfico oferece uma perspectiva diferente. Por exemplo, se queremos mostrar uma comparação do volume de vendas de cada filial da empresa Big Compras no decorrer de um ano, o gráfico de linhas talvez seja mais adequado do que um gráfico de pizza.

A seleção correta do gráfico é muito importante para ter uma visualização da informação clara e objetiva. Quando isso não acontece, em vez de ajudar, podemos confundir o leitor ou fazer com que ele interprete os dados de forma errônea, o que pode causar efeitos catastróficos.

Embora seja uma tarefa que exija experiência para fazer escolhas apropriadas, as perguntas a seguir podem guiar essa decisão:

- *Pretendo comparar valores?* — Nesse caso, considere os seguintes gráficos: colunas, barras, áreas circulares, linhas e de dispersão. Exemplos: comparar número de assinantes no decorrer do tempo, comparar quantidade de produtos vendidos por categorias.
- *Pretendo mostrar como os dados estão distribuídos?* — Para esse caso, em que buscamos ressaltar questões como anomalias e tendências, podemos considerar os seguintes gráficos: dispersão, histogramas, gráficos de área 3D. Exemplos: preço de revenda durante um trimestre, custo de frete para diferentes regiões do país.

- *Pretendo apresentar a composição dos dados?* — Nesse caso, considere os seguintes gráficos: pizza, área, barras empilhadas e colunas empilhadas. Exemplos: porcentagem do número de usuários de redes sociais por faixa de idade, porcentagem de gastos por departamento de uma empresa.

- *Pretendo que minha audiência identifique tendências a partir dos dados apresentados?* — Caso seja esse o objetivo, os gráficos sugeridos são: linha, linha de dois eixos e coluna. Exemplos: visualização de uma página Web no decorrer de um mês.

- *Pretendo destacar o relacionamento entre os dados?* — Para isso, considere os seguintes gráficos: bolha, linha ou dispersão. Exemplos: quantidade de produtos vendidos por categorias e faixa de valores, quantidade de horas gastas na internet por idade e gênero.

Para exemplificar, veja na figura seguinte um gráfico gerado a partir do site de tendências do Google (https://www.google.com.br/trends/), que permite visualizar tendências de pesquisa de diferentes tópicos realizadas na engine de busca do Google no decorrer dos anos. No gráfico de linhas apresentado, os tópicos pesquisados foram: "big data", "internet of things" e "cloud computing".

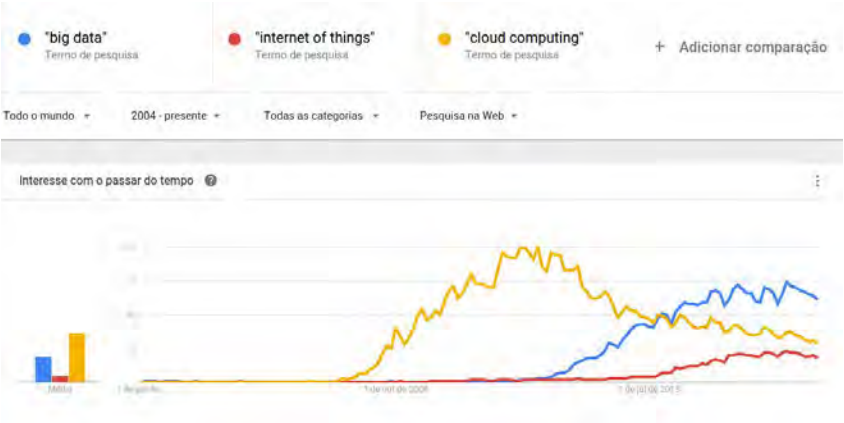

Figura 5.4: Gráfico de linhas para apresentar tendências de pesquisa

Se o objetivo desse gráfico foi apresentar a tendência de pesquisa de cada um dos termos no decorrer dos anos, bem como permitir uma comparação entre os tópicos, podemos perceber que, mesmo sendo um gráfico simples, a informação é passada de maneira clara e objetiva.

Podemos rapidamente observar que o termo "cloud computing" começou a ser pesquisado muito antes do que os termos "big data" e "internet of things". Podemos perceber também que, atualmente, "big data" é o termo mais pesquisado entre os três, provavelmente pelo crescente interesse na adoção do conceito nos últimos anos.

Tipos de gráficos

Algo importante também em relação à visualização dos dados é que, além dos gráficos convencionais que estamos acostumados (como o de barras, de pizza e de linha), há uma variedade de opções para se transmitir uma mensagem, que dependendo do cenário, pode ser a forma mais eficaz de representar visualmente seus dados. Confira a seguir algumas dessas possibilidades.

Mapas

Se o conjunto de dados que você precisa apresentar possui informações geográficas, como nome de cidades, estados, países, códigos postais, latitude e longitude, é provável que uma forma eficiente de apresentar os seus dados seja por meio de mapas.

Sendo provavelmente uma das maiores subseções de tipos de visualização de dados, existe uma variedade de tipos de mapas, cada qual para uma representação diferente da informação. Veja, por exemplo, a figura a seguir que apresenta o gráfico em formato de mapa para mostrar os interesses de pesquisas realizadas no Google por regiões geográficas.

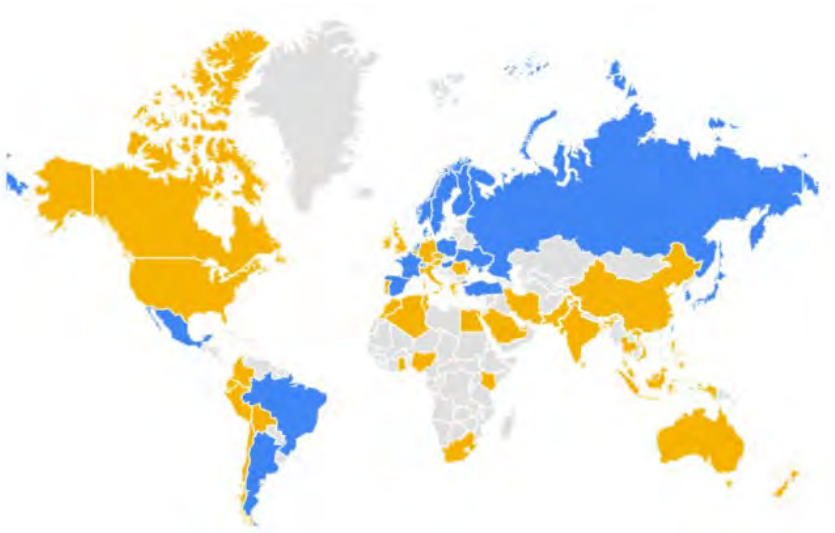

Figura 5.5: Exemplo de visualização de dados em formato de mapa

Esse gráfico em formato de mapa complementa as informações apresentadas no gráfico da penúltima figura. Aqui podemos perceber em quais regiões cada tópico foi mais pesquisado. Podemos perceber, por exemplo, que no Brasil o tópico "big data" (em azul) teve uma média de pesquisa superior ao tópico "cloud computing" (em amarelo). Essa e outras percepções que podemos extrair do gráfico podem servir de base para análises futuras sobre

os principais interesses de pesquisa dos países.

Word clouds

Muitos dados que buscamos representar visualmente são numéricos ou categóricos. Mas e se precisarmos representar visualmente dados baseados puramente em texto? Uma das alternativas é o conceito de *word cloud*.

Também conhecido como *tag clouds*, esse tipo de visualização tem como objetivo representar a frequência de ocorrência de cada palavra existente em uma base de dados textual. Quanto maior a frequência, maior é o tamanho da palavra representada graficamente. Por exemplo, na figura a seguir, é apresentada uma *word cloud* formada pelo conjunto de palavras citadas neste capítulo.

Figura 5.6: Exemplo de visualização no formato Word Cloud

Nesse gráfico, conseguimos perceber rapidamente que as palavras "dados" e "visualização" foram muito mais usadas do que as

demais no texto. Da mesma forma, podemos utilizar essa representação para analisar o feedback dos clientes, para identificar quais palavras foram mais pesquisadas na base de dados da empresa, ou avaliar como os funcionários se sentem a respeito da empresa que trabalham. Tudo isso identificando rapidamente quais foram as palavras mais mencionadas em uma fonte de dados.

Layout circular

Se está trabalhando com grafos direcionados, talvez uma das formas mais eficazes de visualizar seu dado seja em um formato circular. Esse tipo de gráfico ganhou maior popularidade com o pacote de software Circos (http://circos.ca/). Além de apresentar os dados em um formato elegante, o layout circular facilita nossa compreensão sobre como cada elemento está relacionado com os demais, principalmente em casos nos quais existem diversos relacionamentos.

Um exemplo muito interessante de visualização sobre dados da área de saúde utilizando esse gráfico é o Health InfoScape, desenvolvido pela empresa General Eletric (GE) (http://visualization.geblogs.com/visualization/network/). Essa visualização foi criada com base em 7.2 milhões de registros médicos eletrônicos, capturadas pelos dispositivos da GE.

Conforme apresentado na figura a seguir, com base na análise desse grande volume de registros, foi possível criar uma rede que apresenta a associação de uma doença com as demais, oferecendo novas percepções que podem ajudar na saúde dos pacientes.

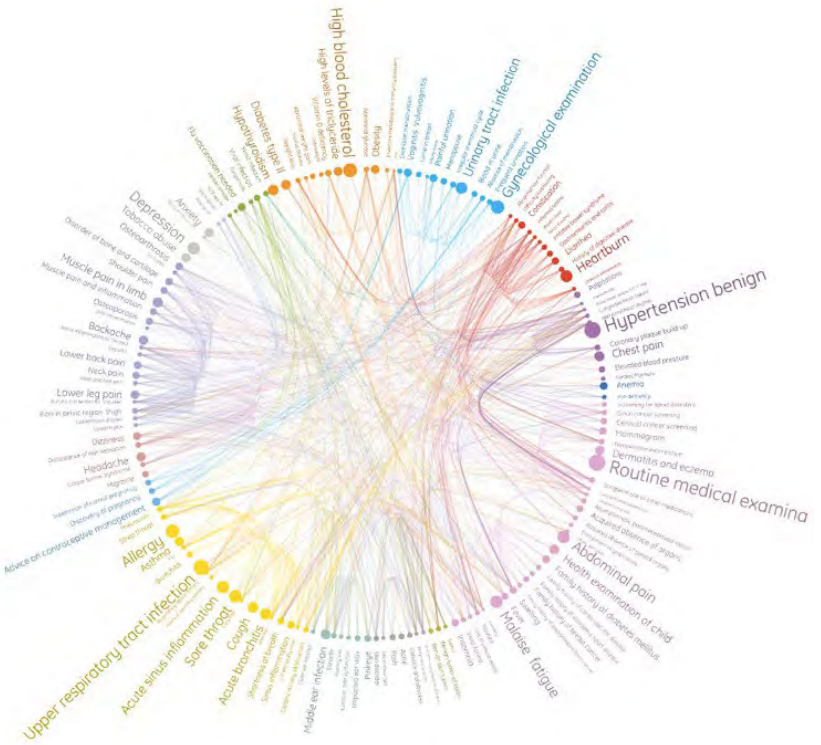

Figura 5.7: Exemplo de gráfico com associação de doenças

A partir desse gráfico, a rápida identificação do relacionamento entre as doenças acelera os processos na área de cuidados de saúde, provendo uma interpretação única aos leitores, bem como uma maior compreensão sobre uma série de sintomas que influenciam o estado de saúde dos pacientes. Provavelmente se o mesmo resultado da análise fosse apresentado somente em formato tabular, as informações obtidas não seriam facilmente compreensíveis e, assim, a utilização de tais dados não teria o mesmo impacto.

5.3 RECURSOS PARA VISUALIZAÇÃO INTERATIVA

Muitos conceitos relacionados a visualização de dados já existem e são utilizados a diversos anos por analistas, engenheiros, designers, entre outros profissionais que trabalham com o processo de visualização. Entretanto, o avanço das tecnologias baseadas na Web, das tecnologias de bancos de dados e de soluções para dispositivos móveis trouxeram novos requisitos e funcionalidades às visualizações de dados.

Como foi apresentado na seção sobre os estágios da visualização de dados, a interação é um dos recursos essenciais para uma visualização eficaz. É por meio da interação que o leitor terá a possibilidade de explorar as informações de acordo com sua necessidade e interesse na representação dos dados.

Mas quais interações podem fornecer essas habilidades? No artigo de 2005 intitulado *Beyond Guidelines: What Can We Learn from the Visual Information Seeking Mantra?*, Brock Craft e Paul Cairns descrevem as seguintes formas de interação na visualização dos dados:

- *Filtragem de itens* — Interação utilizada para permitir que o leitor faça um ajuste na visualização, permitindo controlar quais pontos de dados ficarão visíveis.
- *Detalhes em demanda* — Além de permitir a filtragem, com essa interação o leitor tem a possibilidade de obter informações adicionais de acordo com recursos oferecidos pela visualização, como um clique ou movimento do mouse sobre um ponto de dado. Esse recurso permite obter uma visualização de dados mais "limpa", porém com níveis de detalhes que oferecem maior compreensão dos dados.
- *Relação entre dados* — Nessa interação, é prevista a visualização não somente dos itens de dados, mas

também do relacionamento entre eles.

- *Histórico de ações* — Interação muito importante que oferece recursos ao leitor de retornar a visualização para um determinado estágio de sua interação. Sem essa possibilidade, o leitor fica limitado na quantidade de interações possíveis.

- *Extração de subcoleções e consulta de parâmetros* — Além de permitir que o leitor navegue em diversos cenários da visualização, essa interação faz referência à possibilidade do leitor salvar seu estado atual de visualização para ser utilizado posteriormente.

- *Zoom* — Essa interação está relacionada à redução ou à ampliação da complexidade da representação de dados, por meio de mudanças de escala dos dados. Nessa interação, podemos aplicar dois tipos de efeitos: o *zoom-in* e o *zoom-out*. No efeito *zoom-in* são ampliadas as informações de dados de interesse pelo leitor, reduzindo ou excluindo da área visual os dados de menos interesse. No efeito *zoom-out* são apresentados os dados em mais alto nível, com menos detalhes dos itens dos dados.

Voltando ao gráfico apresentado na figura dos mapas, para que o usuário possa explorar melhor o conteúdo apresentado, sem que a visualização fique sobrecarregada de informações, dois recursos interativos foram acrescentados ao gráfico: o recurso de detalhes em demanda apresentado na figura (a) e recursos de zoom na figura (b).

No primeiro caso, ao passar o ponteiro do mouse sobre um determinado país (no exemplo da figura, o ponteiro do mouse estava sobre o Brasil), é apresentada uma caixa com informações sobre o índice de classificação de cada tópico para o país em

questão. Dessa forma, o usuário pode visualizar essas informações mais detalhadas de acordo com seu interesse. No segundo recurso, é aplicado o efeito *zoom-in*, onde o usuário tem como possibilidade selecionar um país e identificar como a classificação dos tópicos ocorre entre os estados.

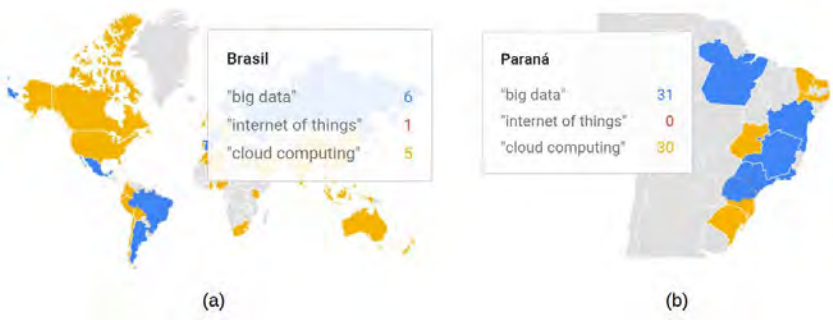

Figura 5.8: Gráficos de mapas com recursos interativos

Os recursos interativos podem oferecer uma experiência rica ao leitor, permitindo que ele navegue pelo dado de acordo com o seu interesse e com a forma que se sente mais agradável de visualizar os dados. Entretanto, para cada ponto de dados que alteramos graficamente na visualização, precisamos reestruturar internamente nossa representação, para que tudo continue condizente.

Isso requer funcionalidades para reduzir ou alterar os dados, modificar os relacionamentos, selecionar novos dados e, até mesmo, alterar a forma de representação gráfica. Para facilitar esse processo, temos como apoio as ferramentas de visualização de dados.

5.4 PROCESSO DE VISUALIZAÇÃO DE DADOS

Uma vez que já compreendemos o que é visualização de dados e como ela pode facilitar a compreensão do significado dos dados, vamos identificar quais as etapas necessárias para gerar uma visualização. Conforme está estruturado este livro, o processo de

visualização de dados é considerado uma das etapas finais de um projeto de Big Data. Ela é realizada durante e após a etapa de análise de dados, por meio da análise exploratória e explanatória.

Entretanto, para maior eficácia do processo de visualização, é importante que ela já seja projetada desde o início do projeto. Isso pode facilitar a estruturação das técnicas visuais utilizadas.

Etapas do processo

No livro *Visualizing Data*, Ben Fry (2008) definiu 7 estágios para a visualização de dados, sendo que muitos deles fazem parte das etapas de armazenamento, processamento e análise de dados. São eles:

- *Aquisição* — Estágio em que os dados para análise são capturados. Processo realizado na etapa de coleta dos dados.
- *Estruturação* — Uma vez que os dados podem ser coletados de diversos formatos, esse estágio é importante para definir uma estrutura padrão para esses dados.
- *Filtragem* — Além da estruturação, deve-se aplicar o estágio de filtragem dos dados, para remover dados incorretos, incompletos ou que não são interessantes para a análise.
- *Mineração* — Pertencente ao estágio de análise de dados, esse estágio tem como objetivo a aplicação de técnicas para extrair informações dos dados.
- *Representação* — Refere-se às atividades iniciais da representação visual dos dados. O objetivo nesse estágio é gerar o modelo visual básico dos dados, como foco principal na análise exploratória dos dados.
- *Refinamento* — Esse estágio é essencial para aperfeiçoar

a visualização dos dados analisados. É nesse momento que as técnicas gráficas são utilizadas para tornar a visualização mais eficiente.

- *Interação* — Além do refinamento, a interação também melhora a visualização dos dados, permitindo inserir funcionalidades que ofereçam uma melhor experiência ao leitor.

Podemos perceber com esses estágios que o processo de conversão de dados para informações visuais exige uma série de etapas. Um dos desafios existentes nesse processo é a preparação dos dados.

Muitas vezes, para que consigamos representar os dados, bem como suas relações, da forma visual que desejamos, é preciso um grande esforço para preparar os dados da forma adequada para serem visualizados. Por exemplo, imagine uma visualização de dados que deseja apresentar relações entre dados referentes ao estoque, às preferências dos usuários pela marca de um produto a partir de redes sociais e aos pedidos de vendas desse produto.

Normalmente, esses dados estão em bases de dados diferentes, com formatos diferentes e possivelmente com algum dado faltando. Além de adequar esses dados para serem analisados pelo computador, teremos de estruturar os resultados para serem usados na visualização. Isso pode levar grande parte do tempo do processo de visualização.

Além disso, com a alta demanda de dados que precisam ser analisados e visualizados atualmente, é importante que existam ferramentas e técnicas que nos permitam automatizar todo esse processo de visualização de dados, uma vez que realizá-los manualmente pode se tornar uma tarefa inviável. A automação completa do processo é praticamente inviável, pois cada ponto de

dado utilizado pode necessitar de uma configuração manual para ser utilizado adequadamente na visualização.

Além de facilitar o processo, a automação também é importante para agilizar a criação das visualizações, atendendo às necessidades de muitas soluções que precisam visualizar as informações em tempo real ou próximo ao tempo real.

As ferramentas atuais para visualização de dados oferecem recursos para atuar principalmente nas duas últimas etapas do processo de visualização de dados: refinamento e interação. Além da possibilidade de gerar diferentes representações visuais com o mesmo conjunto de dados, com essas ferramentas o leitor passa a ser capaz de interagir com os dados, obtendo informações mais detalhadas sobre um ponto específico identificado, permitindo fazer comparações de acordo com a necessidade.

A junção dessas habilidades torna a visualização dos dados muito mais poderosa, permitindo apresentar fatos em formato de uma história. A seguir, é apresentada uma lista de ferramentas que oferecem essas habilidades.

Ferramentas para visualização de dados

Para a criação das visualizações de dados, têm surgido nos últimos anos ferramentas que fornecem o suporte na construção gráfica. Seguindo a tendência das tecnologias de Big Data, atualmente grande parte dessas ferramentas é open source.

Por exemplo, imagine que você precise fazer uma apresentação sobre os dados analisados que fique disponível no site e aplicativo da sua empresa para que outras pessoas possam acessar a qualquer momento essa informação, seja pelo computador, televisão, tablet ou smartphone. Para essa situação, você precisará criar um gráfico responsivo, que se adapte a diferentes tamanhos de tela.

Além disso, como você não estará presente no momento em que o leitor visualizar os dados, é importante que você forneça a ele diferentes formas de interações com os gráficos, como por exemplo, a opção de selecionar algum dado específico para analisar, a opção de dar zoom a um ponto específico ou de mudar o formato do gráfico. Isso aumentará a capacidade do leitor de compreender as informações apresentadas nos dados. Mas como projetar esse tipo de visualização?

Ferramentas para visualização de dados na Web

Se você deseja criar uma visualização de dados e disponibilizá-la na Web, um recurso para facilitar essa criação é a utilização do D3.js (http://d3js.org/). Essa ferramenta consiste em uma biblioteca JavaScript que utiliza recursos de HTML, CSS e SVG para a visualização explanatória de dados.

O nome da ferramenta faz referência ao objetivo que ela se propõe: documentos dirigidos por dados (*Data-Driven Documents*), ou seja, documentos são criados contendo informações sobre o conteúdo e forma de cada gráfico. Com esses recursos, o D3.js permite que você crie diferentes gráficos e os apresente na Web de forma responsiva.

Além disso, por meio de funções JavaScript, o D3.js permite que você crie gráficos com funcionalidades interativas, como as citadas anteriormente. No site da ferramenta, você pode conferir inúmeros exemplos de visualizações para lhe inspirar.

No entanto, caso o seu perfil seja mais de analista e não tenha muito background em desenvolvimento para a Web, criar essas interfaces gráficas com D3.js pode ser um desafio. Uma ferramenta que pode ser mais adequada para esse perfil é o Shiny (http://shiny.rstudio.com/).

Shiny é um framework para desenvolvimento de aplicações Web, disponível como um pacote do software R. Dessa forma, ele torna possível a construção de interfaces Web dinâmicas e interativas, sem a necessidade de implementar interfaces com linguagens como HTML, CSS e JavaScript. O pacote oferece diversas funcionalidades para o desenvolvimento da parte gráfica da aplicação, além de permitir o uso dos inúmeros pacotes R disponíveis para a análise de dados.

Além do Shiny, outro pacote R que vem recebendo grande destaque é o Plotly (https://plot.ly/). Esse pacote oferece funcionalidades para criação de inúmeros gráficos interativos e dinâmicos que podem ser visualizados na Web, permitindo que o desenvolvedor utilize apenas código R para isso. Assim como Shiny, a tarefa de conversão para código HTML, JavaScript e CSS é realizada pelo próprio pacote. Falaremos mais sobre o Plotly na atividade prática deste capítulo.

Ferramentas para visualização de grafos e redes

Embora muitas ferramentas oferecem mecanismos para geração de diferentes tipos de gráficos, existem também as criadas para um propósito específico, como é o caso das ferramentas para visualização de grafos e de redes. Elas têm como objetivo auxiliar a visualização do relacionamento entre os dados, mostrando como os nós de uma base de dados estão conectados.

É o caso de aplicações como análise de redes sociais, análise de associações entre objetos e análise de redes biológicas. Uma das ferramentas open source que tem ganhado muito destaque nesse segmento é a ferramenta Gephi (https://gephi.org/).

Considerada uma ferramenta similar ao Photoshop, porém para grafos, com Gephi você consegue gerar inúmeras formas de representação de grafos, com a possibilidade de manipular tanto a

estrutura dos dados, bem como as formas, cores e outros atributos visuais.

Ferramentas de Business Intelligence

Além de todas as características citadas, uma outra muito importante, principalmente no mundo corporativo, é a integração de ferramentas de visualização de dados com as demais ferramentas de Business Intelligence (BI).

Por exemplo, para criar um *dashboard* com indicadores oriundos dos resultados das análises, a plataforma Hadoop pode fornecer suporte para muitas ferramentas de visualização de dados. Dessa forma, torna-se possível utilizar grandes volumes de dados no processo de visualização.

Uma plataforma que oferece recursos para processamento, análise e visualização de dados é o Statistica Big Data Analytics (SBDA) (http://software.dell.com/products/statistica/). Essa plataforma oferece ferramentas de analytics com uma interface fácil e intuitiva de utilizar.

Na parte de processamento dos dados, o SBDA combina Hadoop, Mahout e técnicas de processamento de linguagem natural para fornecer escalabilidade e alto desempenho para as aplicações. Após processados, a plataforma também oferece uma série de tipos de gráficos que podem ser rapidamente gerados.

Pentaho Business Analytics (http://www.pentaho.com/) também é outra plataforma para integração e análise de dados, por meio de ferramentas que dão suporte ao processamento, exploração e extração de dados com ferramentas para visualização e execução na plataforma Hadoop.

Além do SBDA e do Pentaho, o Tableau (http://www.tableau.com) e o Qlik (http://www.qlik.com/) são

ferramentas de BI com foco especial em visualização de dados, porém elas também permitem a integração com a plataforma Hadoop para a captura dos dados que serão visualizados. Empresas tradicionais de análise de dados como SAP e Micro Strategy perceberam essa necessidade e também já fornecem recursos similares, para que assim usuários possam ter um ambiente apropriado para todo o processo de descoberta de dados no contexto de Big Data.

Para se inspirar

Um exemplo brasileiro inspirador de visualização de dados para o setor público é o DataViva (http://dataviva.info/). Esse projeto foi uma iniciativa do Governo de Minas Gerais e da Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG), tendo como objetivo oferecer uma experiência dinâmica de acesso a um grande conjunto de dados públicos, utilizando recursos de visualização de dados.

Por meio dessa solução, é possível obter uma visão clara sobre aspectos como exportação, importação, empregos e demais atividades do setor formal brasileiro. Em parceria com o Massachusetts Institute of Technology (MIT), os milhares de gráficos gerados nesse projeto foram desenvolvidos utilizando a biblioteca D3.js

A figura a seguir apresenta um gráfico treemap desse portal, apresentando as atividades econômicas que empregam analistas de TI.

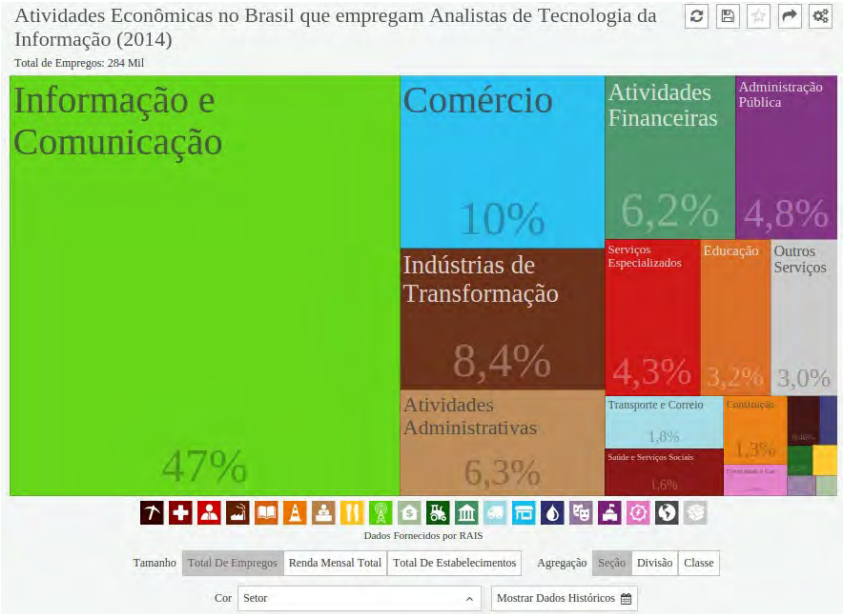

Figura 5.9: Exemplo de gráfico treemap do portal DataViva

Nessa solução, conseguimos ver um dos grandes benefícios da visualização de dados. Perceba, porém, que todos os dados utilizados no portal sobre a economia brasileira já existiam e estavam disponíveis em outras plataformas antes do projeto ser criado. Entretanto, esses dados estavam armazenados em diferentes fontes, com milhões de informações tabuladas, tornando complexa sua análise.

Com as visualizações apresentadas no DataViva, tornou-se possível extrair significados dos dados, trazendo uma nova perspectiva sobre eles. Facilmente conseguirmos fazer comparações entre cidades, estados, produtos, cursos, empregos, entre outros, bem como compreender a evolução dos resultados no decorrer do tempo. Essas visualizações foram então capazes de gerar valor sobre os dados.

Agora executivos e governantes podem utilizar essas

informações para o planejamento de decisões estratégicas. Imagine se essa mesma proposta fosse adotada em outros segmentos, se tivéssemos um portal com resumos de dados relacionados a agricultura, telecomunicação, política, transporte, entre outros. Esses portais ofereceriam informações valiosas para um grande número de profissionais que utilizam esses dados em seu negócio.

5.5 PRATICANDO: VISUALIZAÇÃO DE DADOS COM PLOTLY E R

Já fizemos até agora uma atividade para coletar dados do Twitter e armazená-los em um banco de dados NoSQL, uma para desenvolver uma aplicação em MapReduce e executá-la no Hadoop, e outra para analisar os dados utilizando o software estatístico R. Para a atividade deste capítulo, construiremos gráficos interativos utilizando o pacote `plotly` do software R (https://plot.ly/).

Conforme já mencionado, esse pacote permite criar recursos de visualização de dados do pacote `ggplot2` integrado aos recursos interativos na Web que o `plotly` oferece, tudo isso somente com código R. Para essa atividade, os seguintes passos serão executados:

- **Passo 1:** preparação de um mapa de bolhas;
- **Passo 2:** geração do mapa de bolhas.

Para essa atividade, foram utilizadas as seguintes ferramentas:

- R versão 3.3.1 — https://www.r-project.org/
- RStudio versão 0.99 — https://www.rstudio.com/

Você pode encontrar essa atividade no repositório git do livro.

Passo 1: preparação de um mapa de bolhas

Com o objetivo de compreender as regiões do Brasil onde o

aplicativo Big Compras está sendo mais acessado, vamos criar uma visualização no formato mapa de bolhas. Também conhecido como mapa de símbolos graduados, esse tipo de mapa permite apresentar diferenças quantitativas entre as entidades representadas no mapa, de acordo com a variação da dimensão dos símbolos (nesse caso, as bolhas).

Para dar início à atividade, devemos primeiramente carregar os pacotes necessários para a visualização. Podemos fazer isso com os seguintes comandos:

```
> install.packages("ggplot2")
> install.packages("plotly")

> library(ggplot2)
> library(plotly)
```

Pronto, agora já podemos utilizar em nossa visualização as funções disponibilizadas por esses pacotes. Para dar início à visualização, carregaremos nossa base de dados para o R.

Para essa atividade, usaremos o arquivo `acessos.csv`, que contém o número de acessos ao aplicativo Big Compras por cidade.

```
> df <- read.csv("acessos.csv", sep = ";")
```

Podemos perceber que a base contém 157 registros e 6 variáveis, sendo que cada registro representa informações sobre uma cidade. Para apresentarmos as informações em um mapa, essa base contém as coordenadas geográficas (`lat` e `long`) de cada cidade, conforme podemos observar na sequência.

As colunas `cidade`, `uf` e `regiao` contêm a informação da cidade, sigla e região do estado, respectivamente. Por fim, a coluna `qtd` apresenta o número de acessos realizados na cidade.

Para darmos início à construção do gráfico, criaremos uma

variável para armazenar informações sobre o layout do gráfico.

```
> grafico <- list(
  scope = 'south america',
  showland = TRUE,
  landcolor = toRGB("gray85"),
  countrycolor = ("white")
)
```

Passo 2: geração do gráfico de bolhas em um mapa

Para a geração do gráfico, utilizaremos a função `plot_geo` , que permite gerar diferentes tipos de gráficos em formato de mapas. Nessa função, foram adicionados os seguintes parâmetros:

- `data` : data frame que contém os dados a serem representados;
- `lon` : informação da longitude;
- `lat` : informação da latitude;
- `sizes` : vetor numérico usado para escalar as bolhas em pixel.

Para adicionar as bolhas em nosso mapa, usamos a função `add_markers` , indicando os seguintes parâmetros:

- `size` : vetor de valores utilizado para dimensionar as bolhas;
- `color` : expressão usada para indicar o mapeamento de cores das bolhas;
- `text` : conteúdo a ser apresentado ao passar o cursor do mouse.

Por fim, também utilizamos a função `layout` para definir questões relativas ao design do gráfico. Nesse exemplo, usamos os seguintes parâmetros:

- `title` : título do gráfico;

- geo : objeto com informações sobre o conteúdo do gráfico.

Com esses parâmetros, chegamos à seguinte função:

```
> plot_geo(df, lon = ~long, lat = ~lat, sizes = c(1, 1000)) %>%
  add_markers(
    size = ~qtd, color = ~regiao, hoverinfo = "text",
    text = ~paste(df$cidade, "<br />", df$qtd, ' acessos')
  ) %>%
  layout(title = '<b>Big Compras</b><
            /br>Número de acessos por região', geo = grafico)
```

Pronto, já temos as implementações necessárias para a nossa primeira visualização de dados. Ao executarmos a função, o R deverá gerar o mapa conforme o exemplo apresentado a seguir.

Figura 5.10: Mapa de bolhas criado com Plotly

Ao interagir com o gráfico, você perceberá que já são disponibilizados alguns recursos, conforme ilustrados na figura seguinte. Ao passar o cursor do mouse sobre uma determinada cidade, é apresentado uma box contendo informações sobre o número de acessos (a). Lembre-se de que podemos adicionar outras informações, caso necessário.

Também temos o recurso de zoom-in e zoom-out (b), para podermos visualizar melhor as cidades representadas em cada região. Por fim, temos também a opção de filtrar a visualização de acordo com a região desejada (c). Esses recursos enriquecem a experiência do usuário, permitindo que ele explore os dados apresentados em diferentes perspectivas.

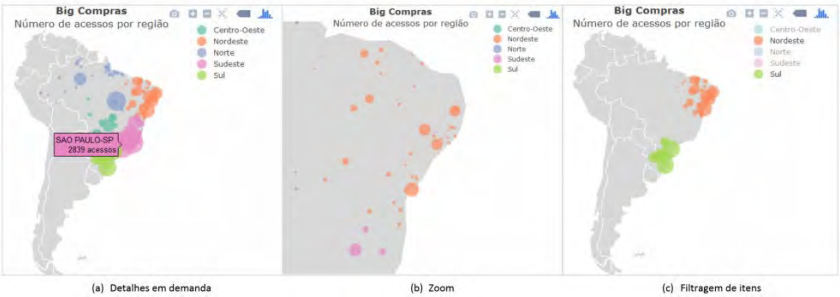

Figura 5.11: Recursos interativos do mapa de bolhas

Além de a visualização estar disponível no console do RStudio, você também pode salvá-la em formato HTML, e visualizá-la diretamente em um browser. A visualização já é apresentada de forma responsiva, sendo adaptada automaticamente de acordo com o tamanho da página. Caso queria, você também encontra no site do Plotly um recurso para hospedar sua visualização de dados na Web.

Esse exemplo é apenas uma das inúmeras possibilidades de visualização de dados que o pacote oferece. Além de estar disponível em R, Plotly também pode ser utilizado na linguagem Python, Matlab e JavaScript, com o D3.js. Aproveite a atividade para praticar

e criar diferentes formas para representar seus dados.

5.6 CONSIDERAÇÕES

Neste capítulo, podemos identificar o valor da visualização de dados. Embora essa técnica já existisse antes do conceito de Big Data, ela se tornou ainda mais importante após a existência de um grande volume de dados, de diferentes estruturas, que precisavam ser resumidos e apresentados de maneira compreensível pelos humanos.

Para tornar ainda mais efetiva a visualização dos dados, atualmente são adicionados recursos interativos aos gráficos, oferecendo uma nova experiência ao leitor, que agora pode interagir com as informações de acordo com sua necessidade. Vimos que já existem diversas ferramentas utilizadas como apoio para a geração de visualização de dados, das quais já oferecem integração com plataformas de Big Data, possibilidades de diferentes formas visuais da informação e compatibilidade com os diferentes dispositivos usados atualmente, como notebooks, tablets e smartphones.

Uma visualização de dados adequada pode gerar inúmeros benefícios, que vão desde uma comunicação mais efetiva até a obtenção de insights para a tomada de decisão. Por meio da inovação, torna-se possível extrair informações valiosas e novas percepções em conjuntos de dados que antes não eram utilizados. Ou seja, seja para aprender, para ensinar ou para compartilhar dados, a visualização pode ser o meio mais eficaz nesses processos.

Resumindo, neste capítulo foi possível responder às seguintes perguntas de um projeto de Big Data:

- Qual a importância da visualização de dados?
- Quais os passos existentes no processo de visualização

de dados?

- Como posso representar meus dados?
- Quais ferramentas utilizar na visualização de dados?

Para saber mais

1. BRATH, Richard; JONKER, David. *Graph Analysis and Visualization:* Discovering Business Opportunity in Linked Data. John Wiley & Sons, 2015.

2. CHANG, Winston. *R graphics cookbook.* O'Reilly Media, Inc., 2012.

3. CHIASSON, Trina; et. al. *Data + Design:* A Simple Introduction to Preparing and Visualizing Information., 2014. Disponível em: https://infoactive.co/data-design/titlepage01.html.

4. DEWAR, Mike. *Getting Started with D3.js.* O'Reilly Media, Inc., 2012.

5. DOUGHERTY, Jack; et. al. *Data Visualization for All.* 2015. Disponível em: https://www.datavizforall.org/.

6. FRY, Ben. *Visualizing Data:* Exploring and Explaining Data with the Processing Enviroment. O'Reilly Media, 2008.

7. GEMIGNANI, Zach; GALENTINO, Richard; GEMIGNANI, Chris; SCHUERMANN, Patrick. *Data Fluency:* Empowering Your Organization with Effective Data Communication. John Wiley & Sons, 2014.

8. HINDERMAN, Bill. *Building Responsive Data Visualization for the Web.* John Wiley & Sons, 2015.

9. ILIINSKY, Noah; STEELE, Julie. *Designing data visualizations.* O'Reilly Media, Inc., 2011.

10. MURRAY, Scott. *Interactive data visualization for the Web.* O'Reilly Media, Inc., 2013.

11. KEIM, Daniel A.; GRINSTEIN, Georges G.; WARD, Matthew O. *Interactive data visualization:* foundations, techniques, and applications. A K Peters/CRC Press, 2010.

12. CRAFT, Brock; CAIRNS, Paul. *Beyond Guidelines:* What Can We Learn from the Visual Information Seeking Mantra? Proceedings of the Ninth International Conference on Information Visualisation, p. 110-118, 2005.

Capítulo 6

O QUE MUDA COM BIG DATA

"Não podemos prever o futuro, mas podemos criá-lo." — Peter Drucker

Diante de todas as características, requisitos e possibilidades de aplicações de Big Data, podemos perceber o quanto ele pode exercer um papel transformador nas mais diversas áreas de negócio. O grande volume de informações tem possibilitado a criação de novos produtos e serviços de dados, novas formas de se otimizar uma tarefa e novas informações para a tomada de decisão estratégica.

Mas para que esses benefícios possam ser alcançados, é necessário inovar, em inúmeros aspectos. Neste capítulo final, você confere aspectos sobre como Big Data está sendo um agente de transformação nas empresas.

6.1 CULTURA ORIENTADA POR DADOS

Espero que tenha conseguido demonstrar a você como os dados podem oferecer conhecimentos valiosos se forem bem explorados. Temos agora a oportunidade de capturar uma imensidão de informação e tornar isso um produto ou serviço.

Mas será que as empresas estão preparadas para essa cultura, na qual os dados passam a ter um papel chave dentro da organização?

Trabalhar com Big Data requer uma mudança não somente técnica; é preciso uma mudança de comportamento.

Seja em uma startup ou em uma empresa de milhares de funcionários, é preciso se preocupar em criar meios para que os dados sejam utilizados eficientemente. Mas o que deve ser feito para isso?

Não existe uma receita mágica para fazer com que se crie uma cultura orientada por dados dentro da empresa. Porém, existem algumas abordagens que podem facilitar o alcance desse objetivo. Um dos entraves para se criar essa cultura é a criação de silos de dados. Podemos definir isso como conjuntos de bancos de dados da empresa que não se comunicam com outros sistemas, sendo usados de forma isolada.

Tradicionalmente utilizada pelas grandes empresas, essa maneira de separar os dados — de forma que somente um pequeno grupo tenha acesso a um conjunto específico de informação — impede que os profissionais explorem todos os dados da empresa. Além disso, pelo fato de que diferentes conjuntos de dados nunca são integrados, perde-se aí a possibilidade de identificar problemas, ou responder a questões que só surgiriam com uma visão completa deles.

Por isso é importante que a empresa crie alternativas para que todos tenham acesso a todos os dados da empresa. Conforme ilustrado na figura a seguir, em busca de gerar essa democratização dos dados, algumas empresas estão adotando um conceito chamado Data Lake. Data Lake é um ambiente no qual os dados estruturados e não estruturados, coletados de diferentes fontes, são armazenados em uma única plataforma (por exemplo, um cluster com ecossistema Hadoop), permitindo a integração dos dados e a geração de análises por meio de tecnologias de Big Data.

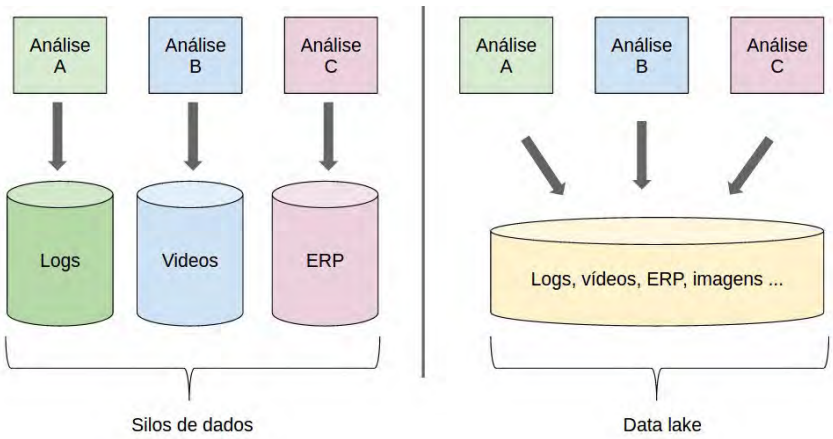

Figura 6.1: Diferença entre silos de dados e data lake

DJ Patil, atualmente *Chief Data Officer* da Casa Branca dos Estados Unidos, mencionou em seu livro *Building Data Science Teams* que uma organização preparada para ter uma cultura orientada por dados é aquela que "*adquire, processa e utiliza dados em tempo hábil para criar eficiências, iterar e desenvolver novos produtos e navegar no cenário competitivo*".

Por esse motivo, é importante também que a organização tenha a iniciativa de coletar e explorar diferentes fontes de dados. Isso possibilita a descoberta de dados relevantes para a empresa. Certamente muitos dados serão descartados, porém, com essa abordagem, a empresa poderá descobrir novas fontes de conhecimento.

Outra abordagem importante para incentivar a cultura orientada por dados é permitir que os profissionais criem e testem suas ideias. Mais importante do que uma prova de conceito, é permitir que esses profissionais consigam criar uma prova de valor sobre sua ideia, demonstrando os benefícios obtidos da solução proposta. A partir de prova de valor, a proposta pode ser aperfeiçoada dentro da empresa, para então entrar em seu portfólio

de soluções.

Sua empresa, ou a empresa na qual você trabalha, tem esse comportamento? Caso a resposta seja não, saiba que ainda são poucas as que também possuem essa maturidade em culturas orientadas por dados.

Entretanto, é importante que exista um movimento em relação a isso, para que os profissionais tenham o ambiente ideal para atuar em projetos de Big Data. Com certeza isso não é garantia de sucesso nos projetos de Big Data, porém a falta desse comportamento pode inviabilizar a execução dos projetos.

6.2 A CARREIRA DO CIENTISTA DE DADOS

Juntamente com a popularidade de Big Data, a profissão cientista de dados também se tornou notória. Entretanto, assim como Big Data, não há ainda uma definição específica do que vem a ser esse profissional e quais conhecimentos são necessários para se tornar um.

Esse termo foi cunhado por DJ Patil, como um meio de classificar uma equipe de profissionais que trabalhavam com os produtos e serviços de dados, que segundo Patil, trabalhavam com os dados e com a ciência para criar novas soluções. Por isso o nome cientista de dados.

Conforme as empresas foram se interessando por Big Data e deram início a provas de conceito sobre esse tema, elas passaram a buscar esse profissional. O problema é que elas exigiam (algumas ainda exigem) que o cientista de dados a ser contratado tivesse conhecimentos profundos em ciência da computação, programação, matemática, banco de dados, estatística, design e negócios.

O que se costuma dizer é que empresas assim não estão em

busca de um profissional, e sim de um unicórnio. Pois sabemos que não é possível (ou no mínimo muito raro) existir um profissional com tamanha habilidade. Por isso, fique calmo, você não precisará saber tudo isso para atuar em um projeto de Big Data.

As empresas atualmente estão mais conscientes de que um projeto de Big Data deve ser formado por uma equipe heterogênea, com profissionais de diferentes habilidades trabalhando em conjunto para o desenvolvimento da solução. Com isso, foram surgindo carreiras específicas para cada especialidade, como o analista de dados, o engenheiro de dados, o gerente de projetos de Big Data, o administrador de infraestrutura de Big Data, entre outros.

Por exemplo, vimos no decorrer do livro 4 estágios distintos em um projeto de Big Data: captura e armazenamento dos dados, processamento dos dados, análise dos dados e visualização dos dados. Se você tem interesse em atuar especificamente em alguma dessas áreas, veja a seguir sugestões de áreas em que você deveria se especializar:

- **Captura e armazenamento dos dados:** banco de dados relacional, banco de dados NoSQL, linguagem SQL, frameworks para transferência de dados em lote, estrutura e gerenciamento de dados, armazenamento de dados na nuvem, frameworks para transferência de dados em streaming, sistemas de arquivos distribuídos, técnicas de agregação de dados, armazenamento de dados em data warehouse, armazenamento de dados em data lakes, técnicas de tolerância a falhas.
- **Processamento de dados:** frameworks de processamento distribuído em lote e em tempo real, processamento de dados na nuvem, linguagens de programação, computação em nuvem, testes,

otimização, técnicas de escalabilidade, disponibilidade, administração de cluster, latência e desempenho de aplicações.

- **Análise de dados:** técnicas de mineração de dados, técnicas de aprendizado de máquina, métodos estatísticos, técnicas de filtragem e limpeza dos dados, expressão regular, matrizes, modelagem, frameworks e bibliotecas para manipulação de dados, frameworks de análise de dados em ambientes distribuídos.

- **Visualização de dados:** conhecimento em experiência do usuário, design, computação gráfica, *storytelling*, programação Web, frameworks para visualização de dados, habilidades em comunicação.

Além de ter profissionais qualificados para cada um desses estágios, é importante em um projeto de Big Data que esses profissionais tenham a capacidade de conversar entre eles e com os profissionais da empresa que possuem conhecimento do negócio em que o projeto será aplicado. Para isso, é necessário que esses profissionais tenham, ao mesmo tempo, conhecimento profundo sobre a área que atuam, mas também tenham um conhecimento geral sobre as outras áreas.

Esse tipo de profissional é também conhecido como o profissional "Tipo T", conforme apresentado na figura seguinte. Ou seja, mesmo que você seja um especialista no processamento de dados em Hadoop, por exemplo, é importante que você tenha conhecimentos básicos sobre outras áreas, como coleta e visualização de dados.

Essa abordagem evita situações como a de um gestor do projeto solicitar algo para os analistas que seja impossível de ser modelado, ou que os analistas preparem os dados de uma maneira que

inviabilize a construção de gráficos dinâmicos.

Figura 6.2: Profissional tipo T

Além dessas habilidades, a comunicação entre todos os profissionais durante todo o projeto é essencial. Por esse motivo, ser comunicativo é uma característica tão esperada dos cientistas de dados. Além do conhecimento técnico e habilidades em comunicação, listo a seguir outras características esperadas em um cientista de dados:

- **Curiosidade** — Em eventos de Big Data e Ciência de Dados que participei nos últimos anos, a curiosidade é citada como uma das principais características de um cientista de dados. Ele deve ter curiosidade e disposição para aprender sobre o negócio em que a empresa atua, sobre as diferentes áreas dentro da organização, e sobre diferentes tecnologias e métodos de análises de dados.

- **Colaboração** — Essa característica está aliada à comunicação. Uma equipe de cientista de dados deve ter em mente que eles devem colaborar entre si e com os demais profissionais da empresa, para alavancar a cultura orientada por dados.

- **Criatividade** — Ter um conhecimento técnico sobre a área que você atua é de suma importância para trabalhar com Big Data. Entretanto, isso não é suficiente. Trabalhar com dados exige que você exerça seu lado criativo, pensando em formas inovadoras de se utilizar os dados.

- **Pensamento analítico** — É importante que um cientista de dados saiba fazer perguntas sobre os dados, bem como tenha a capacidade de analisar os resultados e entender o que eles expressam. São esses profissionais que poderão auxiliar a empresa para obter valor sobre os dados.

- **Comprometimento** — Trabalhar com uma vasta quantia de dados de fontes internas e externas é algo complexo, e exige paciência e prática para chegar ao resultado final com sucesso. Para isso, é preciso que cada profissional tenha o comprometimento de fazer o seu melhor na área em que possui expertise.

Concluindo, você não poderá ser um cientista de dados se não estiver disposto a aprender (continuamente) novas tecnologias, novos meios de resolver problemas. Nem se não tiver interesse em trabalhar em equipe com profissionais de outras áreas, e em descobrir o que os dados podem revelar. Porém, se você acredita que se enquadra nesse perfil de cientista de dados, saiba que há muitas empresas procurando por você.

Mas onde você pode estudar para ter essas habilidades técnicas necessárias para atuar com Big Data? Isso ainda é um problema quando falamos da carreira dos cientistas de dados.

No momento ainda há poucas instituições que oferecem a capacitação para essa profissão, sendo que as que oferecem fazem isso há poucos anos, portanto, não possuem muitos alunos formados até o momento. Como resultado, temos uma oferta maior do que a demanda. Atualmente, a procura por profissionais capacitados é um dos grandes desafios das empresas que atuam ou desejam atuar com Big Data.

Além dos cursos oferecidos pelas instituições, existem outras alternativas que podem alavancar o conhecimento em Big Data. É possível, por exemplo, utilizar bancos de dados abertos disponíveis em diversos sites da Web para começar a explorá-los e assim aprender mais sobre esse processo.

Caso não saiba o que fazer com esses dados, uma alternativa é usar sites que lhe ofereçam não somente os dados, mas desafios na utilização deles. Um desses sites é o Kaggle (https://www.kaggle.com/), uma plataforma para cientistas de dados onde empresas lançam desafios, dos quais usuários e equipes de usuários podem competir em busca de resolvê-los. Além disso, muitas empresas já utilizam esse portal para encontrar profissionais que apresentem habilidades em Big Data.

Outra alternativa para aperfeiçoar o conhecimento em Big Data é a participação em hackathons. Sendo uma combinação das palavras em inglês *hack* e *marathon*, esse nome é dado para competições com foco na prototipação de uma solução tecnológica em um curto período de tempo (de 24 a 48 horas, normalmente).

Mesmo não sendo restrito a Big Data, é comum que em muitas dessas competições sejam criados desafios que envolvem a

manipulação de grande volume de dados. Além de ter a oportunidade de desenvolver um produto que desperte o interesse de investidores, nesses eventos você tem a oportunidade de conhecer diversas pessoas (inclusive profissionais renomados) que atuam na mesma área que a sua, aumentando o seu networking.

6.3 A PRIVACIDADE DOS DADOS

Embora o cientista de dados seja considerado atualmente uma das carreiras mais promissoras quando se fala em Big Data, além dele, existe um outro profissional que certamente atuará em larga escala na era do Big Data: o advogado especialista em violação de privacidade de dados.

O aumento do volume de dados gerados por pessoas e dispositivos ocorreu de forma acelerada. Juntamente com esses dados, surgiram milhares de aplicações com diferentes serviços orientados a dados. Enquadram-se aqui exemplos como os sistemas de recomendações utilizados por sites de e-commerce, streaming de músicas e vídeos, a publicidade personalizada e a análise de sentimento em redes sociais.

Esses e inúmeros outros serviços surgiram de forma tão avassaladora, que não foi possível identificar previamente os limites e o impacto do uso de tais serviços. Como resultado, muitos deles estão coletando um grande número de informações dos usuários, como localização, hábitos de compra, estado de saúde, hábitos de leitura, o que come, o que veste, o que assiste, como se exercita. Esse cenário desencadeia uma série de possíveis problemas de privacidade.

Um dos grandes problemas com esses serviços é que, na maioria dos casos, os dados são capturados sem a anuência do usuário. O usuário utiliza o serviço sem ter conhecimento de que, por exemplo,

os dados do seu carro estão sendo coletados pela seguradora, que as informações de sua localização geográfica estão sendo usadas para oferecer promoções, ou que o modo como ele navega nas páginas Web também está sendo utilizado para identificar padrões de comportamento na internet.

Já temos exemplos reais de problemas relacionados à privacidade de dados. Talvez um dos mais notáveis seja o ocorrido com a empresa Target. Essa grande empresa americana no setor varejista usou estatísticas de compras para criar um modelo capaz de prever quais mulheres estavam grávidas, para assim fazer a propaganda de produtos direcionadas a elas.

Isso foi feito avaliando comportamentos nos históricos de compras que estivessem relacionadas à gravidez, como por exemplo, a compra de loções e suplementos de cálcio e zinco. Esse modelo possibilitou a Target a enviar cupons de desconto para essas mulheres.

O problema ocorreu quando a empresa recebeu uma reclamação de um homem, solicitando uma explicação sobre o fato de sua filha adolescente ter recebido cupons para compras de roupas de bebês. Entretanto, alguns dias depois, o mesmo homem voltou a falar com a empresa dizendo que a filha assumiu que, de fato, estava grávida.

Esse problema dá origem a diversas questões: como garantir que os dados capturados e utilizados para traçar seu perfil estão de fato condizentes? Como ter a opção de escolher quais informações podem ser usadas por terceiros? Como criar uma política de privacidade que me mantenha protegido no uso dos dados de terceiros?

Outra questão deve ser avaliada quando falamos em Big Data: a possível discriminação com base na análise dos dados. Uma empresa que decide não recrutar um usuário para um emprego

devido ao seu histórico de comportamento nos sites e nas redes sociais está agindo de forma ética? E se ela recusar ocorrer ao usuário que desejar obter um cartão de crédito, um seguro de vida ou uma vaga em uma universidade?

Sabemos que discriminação é algo ilegal, mas como identificar que ela ocorre no contexto de Big Data? Como saber que dados ilícitos estão sendo utilizados? Temos aqui um grande problema a ser debatido, principalmente para evitar que Big Data tenha um impacto negativo nas classes mais vulneráveis.

Infelizmente, a pessoa ou empresa que sofrer alguma violação de privacidade ainda possui pouco respaldo da legislação. Essa situação ocorre em nível global.

Cada governo está se adaptando para regulamentar questões de violação de privacidade. No Brasil, tivemos a iniciativa do Marco Civil da Internet, contendo um conjunto de regras e guias sobre a utilização dos dados pelos serviços. Entretanto, essa tarefa é muito complexa e difícil de ser controlada.

Big Data é um conceito poderoso e pode oferecer inúmeros benefícios. Para evitar que sua utilização gere problemas de violação de privacidade, muitas questões ainda precisam ser debatidas e novos procedimentos devem ser criados, tais como o maior controle individual sobre os dados coletados pelas empresas, maior transparência no uso desses dados e maior segurança em relação ao armazenamento, utilização e divulgação dos dados.

6.4 NOVOS MODELOS DE NEGÓCIOS

Estamos vivendo em uma era de transição, em que soluções disruptivas estão sendo criadas, transformando negócios e mudando o status quo de como as coisas funcionam. Já temos diversos

exemplos de soluções que estão tendo sucesso nesse sentido.

Um desses exemplos é o Uber, uma aplicação com foco no transporte de passageiros. Essa empresa trouxe um novo modelo de negócios para esse segmento, que até então era realizado somente por taxistas.

Com sua proposta colaborativa, outras pessoas que possuem um carro passam a oferecer o mesmo serviço, podendo assim cobrar um valor do serviço mais barato, já que eles não pagam as mesmas taxas que os taxistas. Por sua característica disruptiva, a aceitação desse novo serviço não é bem vista por todos, tanto que estamos acompanhando como está sendo polêmica a entrada dessa solução no Brasil.

Outras duas empresas que estão revolucionando o segmento em que atuam é o AirBnB e o Netflix. O AirBnB (*AirBed & Breakfast* — cama de ar e café da manhã) trouxe uma nova alternativa às pessoas que precisam de uma hospedagem temporária, além dos tradicionais serviços oferecidos pelos hotéis.

Com essa solução, tornou-se possível alugar um quarto, casa ou apartamento de milhares de pessoas do mundo todo, que oferecem seu espaço para alugar dentro do aplicativo AirBnB. O resultado foi uma nova forma das pessoas terem lucro com seu imóvel, além de uma infinidade de alternativas de hospedagem. Com certeza essa solução impactou os negócios do ramo hoteleiro.

O mesmo impacto está causando a Netflix, na indústria da televisão. Esse serviço trouxe uma nova forma de assistir filmes e seriados, na qual o usuário paga uma assinatura mensal e tem a sua disponibilidade filmes e seriados para que ele possa assistir no dia e horário que quiser.

Além disso, nos últimos anos, a empresa passou a criar

conteúdo exclusivo de filmes e séries, atraindo ainda mais os usuários para adquirirem o serviço. Imaginem quantas pessoas estão deixando de assistir programas da TV aberta, ou estão cancelando sua TV por assinatura, para utilizar somente o serviço da Netflix. Com certeza ela está revolucionando a forma com que vamos assistir televisão daqui para a frente.

Além dessas 3 soluções, podemos identificar outros serviços que mudaram a experiência dos usuários. Por exemplo, qual é o método que você atualmente mais ouve músicas? A maioria das pessoas aderiram aos serviços online de streaming, como o Spotify, Pandora e Deezer.

Se pensarmos que a 10 anos atrás não utilizávamos o smartphone para ouvir música e muito menos serviços como esses, podemos perceber como em pouco tempo houve uma transformação na indústria musical. Nesse estilo inovador, temos também serviços como o Dropbox para compartilhamento de arquivos, e o Coursera para aulas online.

Mas imagino que você esteja pensando: qual a relação dessas soluções com Big Data? Pois bem, sem tecnologias de Big Data para oferecer uma infraestrutura escalável e de alta disponibilidade para essas soluções, possivelmente não seria possível criar uma solução colaborativa que fosse utilizada por milhões de usuários 24 horas por dia.

Além disso, essas soluções usam os dados de forma criativa para conseguir analisar e extrair informações que tornem seus serviços ainda mais eficientes. Por exemplo, a própria Netflix captura dados do comportamento do usuário, tais como: quais filmes foram assistidos, quais os usuários desistiram no meio, qual cena o usuário voltou a assistir, em qual momento o usuário adiantou um filme e quais foram as categorias de filme mais assistidas.

Todas essas métricas são usadas para que o serviço seja oferecido de forma cada vez mais personalizada ao usuário, e também serve de insights para a escolha do conteúdo na produção das séries e filmes. Ou seja, Big Data tem um papel crucial nessas soluções, pois somado às capacidades oferecidas pelos avanços da Internet, da computação móvel e de computação em nuvem, elas estão abrindo portas para a criação de soluções que rompem as barreiras existentes nas soluções tradicionais.

O que é importante notar é que essa mudança que estamos vivenciando está apenas em seu início. Ainda há muito que pode ser criado utilizando as tecnologias que temos disponíveis atualmente. Por isso a inovação é tão importante quando falamos em Big Data.

Além de todas as questões técnicas envolvidas em um projeto, é preciso criatividade para extrair o melhor que Big Data pode oferecer. Você trabalha na área de agricultura, publicidade, telecomunicação, saúde, esporte, biologia, manufatura, direito, ou qualquer outra área? Você pode pensar em como Big Data pode auxiliar o negócio em que você atua.

Por ser uma abordagem ainda recente, você tem grandes possibilidades de se destacar com sua solução proposta. Eu sei, não é uma tarefa fácil, mas sua oportunidade é agora.

6.5 MENSAGEM FINAL

Chegamos ao fim do último capítulo do livro. Espero que você tenha se interessado por essa jornada ao mundo de Big Data e que tenha se motivado a atuar nessa área tão promissora.

O que mais me motiva em Big Data é saber que temos a possibilidade de utilizar algo que temos em abundância (o grande volume de dados) e que, com isso, podemos criar soluções

inovadoras. Saber que você pode gerar valor por organizar os dados que antes não eram compreendidos, por contar uma história que esclareça o porquê dos fatos ocorridos, por disponibilizar informações de forma tão rápida que permita acelerar a resolução dos problemas, por criar novos produtos e serviços que mudem a vida das pessoas para melhor. Isso não é sensacional?

Se este livro foi um dos primeiros contatos que você teve com Big Data, você deve estar ciente de que a jornada para se aprofundar no tema é longa. Entretanto, ela pode ser muito divertida e cheia de descobertas. O que pode acontecer durante o caminho é você se apaixonar por dados e criar uma relação eterna com eles.

Bem-vindo ao mundo de Big Data!

Para saber mais

1. BARLOW, Mike. *Learning to Love Data Science.* O'Reilly Media, Inc., 2015.
2. BARLOW, Mike. *The culture of big data.* O'Reilly Media, Inc., 2013.
3. HARRIS, Harlan; VAISMAN, Marck; MURPHY, Sean Gordon. *Analyzing the Analyzers:* An Introspective Survey of Data Scientists and Their Work. O'Reilly Media, Inc., 2013.
4. KING, John; MAGOULAS, Roger. *Data science salary survey:* tools, trends, what pays (and what doesn't) for data professionals. O'Reilly Media, Inc., 2016. Disponível em: http://www.oreilly.com/data/free/2016-data-science-salary-survey.csp.
5. PATIL, DJ; MANSON, Hilary. *Data Driven:* Creating a Data Culture. O'Reilly Media, Inc., 2015. Disponível em: http://www.oreilly.com/data/free/data-driven.csp.