XPe

Aprenda com quem faz

Coleta e Armazenamento

Taynan M. Ferreira

2022

SUMÁRIO

Capítulo 1. Computação em Nuvem

O que é Computação em Nuvem?

A AWS define a computação em nuvem como “a disponibilização sob demanda de poder computacional, banco de dados, armazenamento, aplicações e outros recursos de TI através de uma plataforma de serviços via internet com cobrança conforme o uso”.

Podemos listar seis grandes vantagens da adoção de computação em nuvem para as organizações:

- Substituição de CapEx por OpEx: a grande variedade de serviços disponíveis e cobrança, conforme o uso, permite que se tenha acesso e que se pague apenas os serviços consumidos e sem a necessidade de mobilizar grande capital em servidores e data centers.

- Economias de escala: o fato de os provedores de serviços em nuvem atenderem a uma grande quantidade de clientes faz com que sejam capazes de gerar importantes economias de escala e, por consequência, menores preços do que se cada cliente se responsabilizasse pelos investimentos em data centers.

- Planejamento de capacidade: os serviços em nuvem eliminam (ou reduzem em muito) a necessidade de assumir com antecedência a capacidade que será necessária provisionar para um determinado produto ou aplicação. Sem o uso de serviços em nuvem, as empresas podem ficar com recursos ociosos ou com demanda maior que a capacidade provisionada, ao terem que planejar a disponibilização de

infraestrutura computacional com antecedência. Ao utilizar a nuvem, é possível adequar os recursos provisionados à demanda, aumentando/diminuindo com facilidade frente a um crescimento/redução da demanda.

- Maior velocidade e agilidade: o ambiente em cloud permite que recursos computacionais sejam disponibilizados aos desenvolvedores em minutos, aumentando significativamente a agilidade da organização no desenvolvimento de novos produtos e aplicações.

- Eliminação de custos com Data centers: o uso de computação em nuvem permite que as organizações possam focar em projetos e inovações relacionadas à sua área de atuação, ao invés de despender tempo e recursos com infraestrutura.

- Alcance global em minutos: o uso de cloud computing permite a implantação de aplicações em diversas regiões do mundo em poucos minutos, viabilizando baixa latência e melhor experiência aos consumidores a custos mínimos.

Aspectos financeiras

Para entender as implicações e benefícios financeiros de se utilizar a computação em nuvem, é importante primeiramente definirmos dois termos: CapEx e OpEx.

Capital expenditures (CapEx) consistem em investimentos feitos por uma companhia para adquirir ou aprimorar ativos com o intuito de serem usados a longo prazo.

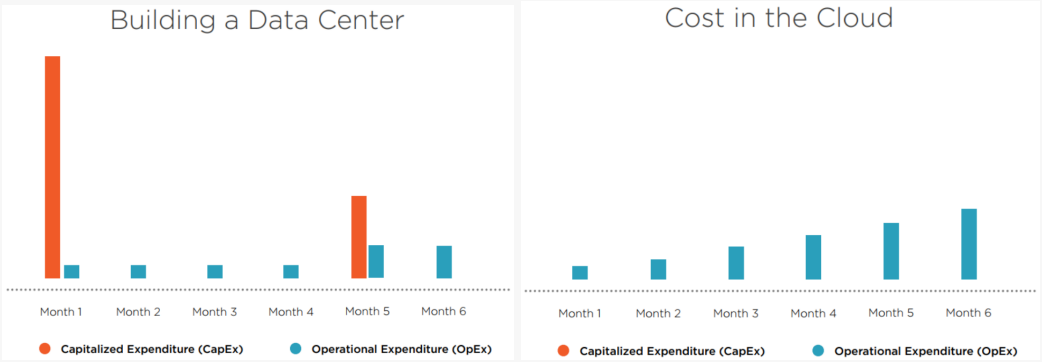
Investimentos em CaPex têm o intuito de expandir o escopo ou agregar algum benefício financeiro futuro à empresa. Exemplos de CaPex incluem propriedades, plantas, edifícios, tecnologia e equipamentos.

Trazendo para a realidade dos investimentos que uma empresa pode fazer na sua infraestrutura computacional, como exemplos de CaPex podemos citar os investimentos relacionados a adquirir ou manter um data center (terreno, edifício, servidores, etc.)

Operating expenses (OpEx) consistem nos custos que uma empresa incorre para sustentar o dia a dia de suas operações. Entre os exemplos de despesas que podem comumente ser consideradas como OpEx, podemos citar os custos com aluguéis, salários, pagamentos de juros, entre outros.

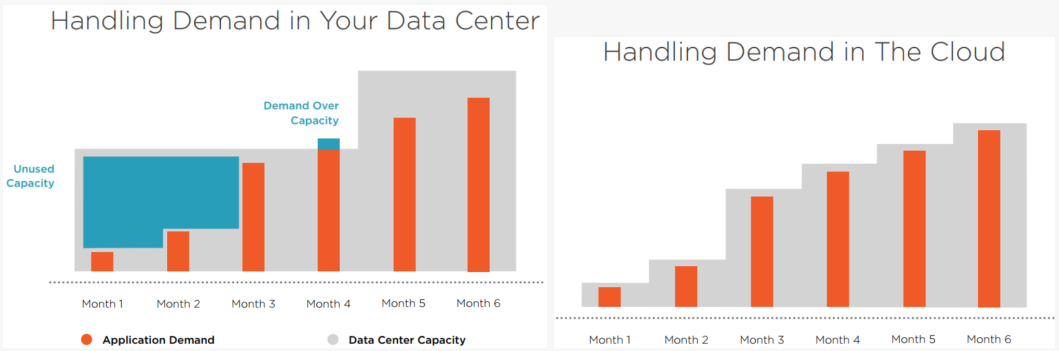
Voltando aos exemplos relacionados à infraestrutura computacional de uma companhia, os custos necessários no dia a dia à operação de um data center (conectividade, energia elétrica, água, salário de funcionários, etc.) são considerados OpEx.

Figura 1 – Comparação de custos em data center on-premise e na cloud.



Fonte: Pluralsight: Fundamental Cloud Concepts for AWS (David Tucker).

Figura 2 – Comparação de relação entre oferta e demanda de recursos em cenário on-premise e cloud.



Fonte: Pluralsight: Fundamental Cloud Concepts for AWS (David Tucker).

Terminologia Básica

- Elasticidade: elasticidade é a capacidade de aquisição de recursos computacionais quando necessário e a liberação desses mesmos recursos quando não mais necessário.

- Confiabilidade: confiabilidade é a capacidade de uma solução de fornecer a seus usuários, sempre que solicitada, a funcionalidade para a qual foi projetada.

Modelos de Deploy

- Público: modelo no qual a aplicação é executada totalmente na nuvem pública, tendo sido migradas de ambientes on-premise ou construídas originalmente na cloud pública.
- Privado: nesse modelo, infraestrutura on-premise é disponibilizada via virtualização e ferramentas de gerenciamento de recursos. Nesse modelo, nem todos os benefícios da computação em nuvem são usufruídos.
- Híbrido: na implantação híbrida, serviços e infraestrutura on-premise e em nuvem são conectados entre si. Na maior parte das vezes, essa estratégia é utilizada para estender e crescer a infraestrutura já existente de uma organização, conectando seus sistemas internos à cloud pública.

Modelos de Serviço

- Infraestrutura como Serviço (Infrastructure as a Service - IaaS): nesse modelo, há a provisão da infraestrutura mais fundamental, ou seja, de armazenamento, rede, sistema operacional, servidores e outros recursos básicos de computação. É o modelo com maior nível de flexibilidade e customização, permitindo o gerenciamento dos mais diversos aspectos da infraestrutura conforme as necessidades da corporação.

- Plataforma como Serviço (Platform as a Service - PaaS): no modelo PaaS o gerenciamento da infraestrutura - planejamento de capacidade, manutenção de software, patching, etc. - fica a cargo do provedor de nuvem, de modo

que as empresas podem se focar unicamente no desenvolvimento, implantação e manutenção de suas aplicações.

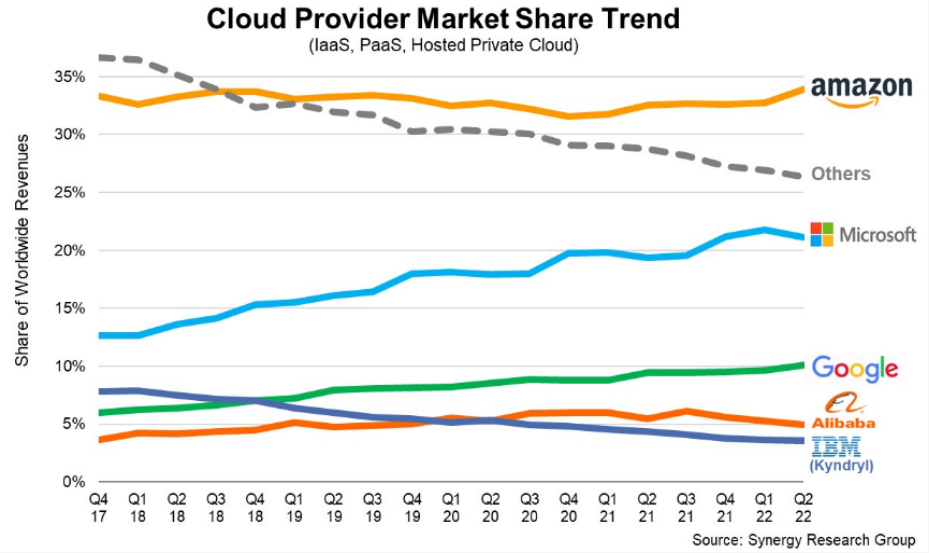
- Software como Serviço (Software as a Service - SaaS): solução na qual uma aplicação é totalmente gerenciada pelo provedor, liberando do usuário a necessidade de se preocupar com qualquer tipo de manutenção, planejamento de capacidade. Nesse modelo, o usuário se preocupa unicamente com o uso do software em si.

Overview do Mercado

Dados do Synergy Research Group referentes ao segundo trimestre de 2022 mostram que os gastos de companhias ao redor do mundo com infraestrutura em nuvem atingiram a marca de USD 55 bilhões. Esse número representa um crescimento de 29% frente ao ano anterior e de 35% a dólar constante.

O gráfico abaixo mostra a evolução do market share dos principais provedores de cloud desde o fim de 2017. Nota-se a dominância constante da Amazon, com forte crescimento do Google e da Microsoft nos últimos anos e queda vertiginosa de outros players fora do top 5.

Figura 3 – Evolução do market share dos principais provedores de serviços de computação em nuvem ao longo dos últimos anos.



Fonte: Synergy Research Group.

A Gartner, tradicional empresa de consultoria, avalia anualmente diversos setores e tecnologias (bancos de dados, plataformas de CRM, ERPs, computação em nuvem, entre muitos outros), avaliando o mercado com relação a sua direção, maturidade e participantes. A metodologia da Gartner - sintetizada no chamado “quadrante mágico” de Gartner - avalia cada um dos players do mercado com relação à sua abrangência de visão e capacidade de execução, classificando-os em desafiadores, líderes, visionários ou operadores de nicho.

A figura abaixo mostra o resultado do quadrante mágico de Gartner para infraestrutura em nuvem e serviços de plataforma.

Figura 4 – Posicionamento competitivo dos principais provedores de serviços em nuvem segundo a Gartner.

Fonte: AWS.

Vemos que os provedores com maior market share são também os classificados como líderes nesse mercado pela Gartner: Amazon (com Amazon Web Services - AWS), Microsoft (Microsoft Azure) e Google (Google Cloud Platform - GCP).

> Capítulo 2

Capítulo 2. Conhecendo a AWS

Infraestrutura AWS

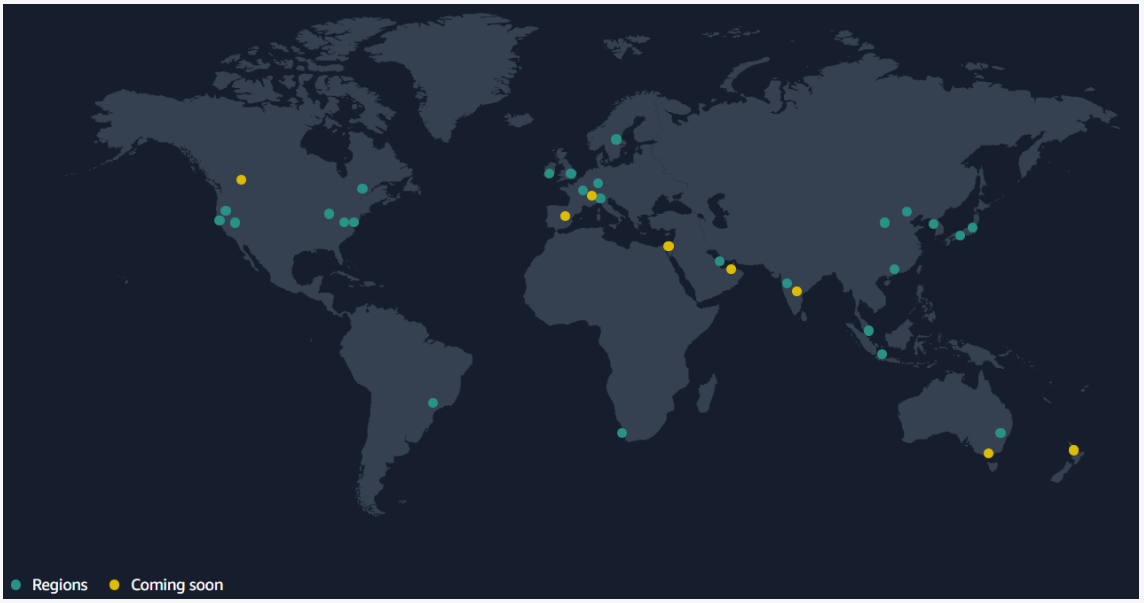
A AWS dispõe de uma infraestrutura global baseada nos conceitos de Regiões (AWS Regions) e Zonas de Disponibilidade (Availability Zones). Uma Região consiste em uma localização geográfica com múltiplas Zonas de Disponibilidade. Cada Zona de Disponibilidade, por sua vez, é composta por um ou mais data centers, cada qual com sistema redundante de energia elétrica e conectividade e dispostos em instalações separadas.

A existência de múltiplas Zonas de Disponibilidade permite a criação de infraestrutura e soluções de alta disponibilidade, tolerantes a falhas e escaláveis, características essas dificilmente replicáveis em um único data center.

Atualmente (Setembro/2022) a infraestrutura global da AWS conta com 26 regiões e 84 zonas de disponibilidade, atendendo a 245 países e territórios. Pelo menos 8 regiões adicionais estão previstas para serem lançadas em breve.

A figura abaixo mostra a disposição das Regiões ao redor do mundo, tanto as já existentes quanto as planejadas para serem inauguradas em breve.

Figura 5 – Evolução do market share dos principais provedores de serviços de computação em nuvem ao longo dos últimos anos.



Fonte: AWS.

Acessando a AWS

São três as principais formas de interação com os serviços da AWS, cada qual com suas vantagens e principais casos de uso:

- AWS Management Console
- AWS Command Line Interface (CLI)
- AWS Software Development Kits (SDK)

Neste curso, estaremos interagindo com a AWS através do Console e SDK para o Python (Boto3).

Custos na AWS

Antes de começar com a "mão na massa" é importante entender em alto nível como funciona o sistema de custos e cobrança da AWS, bem como maneiras efetivas de economizar ou até mesmo explorar a plataforma sem nenhum tipo de custo.

É importante conhecer os drivers fundamentais de custos na AWS: computação, armazenamento e tráfego de saída. Além disso, na AWS há três diferentes tipos de ofertas gratuitas (*free tier*):

- 12 meses gratuitos: ofertas disponíveis por 12 meses a partir do cadastro na AWS
- Testes gratuitos: ofertas de gratuidade de curto período, cujo prazo se inicia após o início da utilização do serviço
- Para sempre gratuitos: ofertas que não expiram e estão disponíveis para todos os clientes da AWS

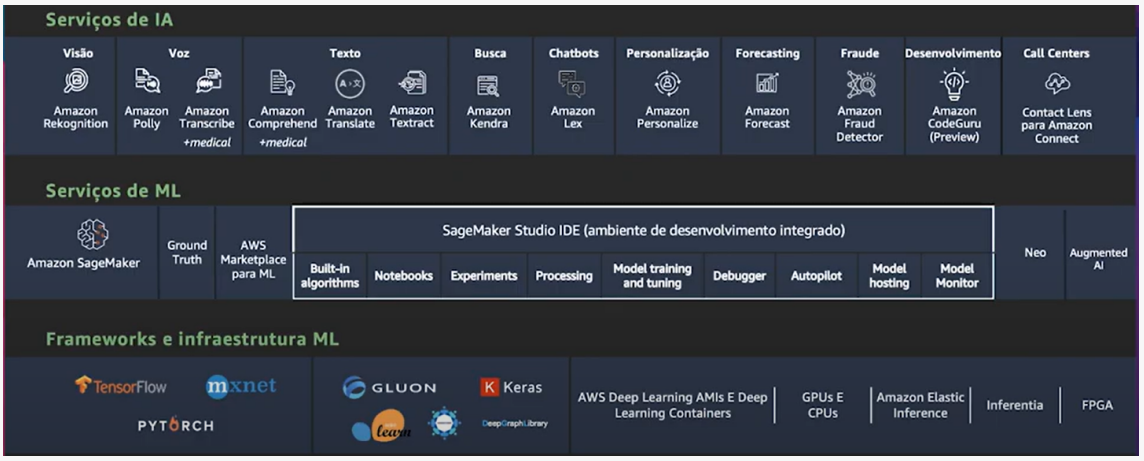
> Capítulo 3

Capítulo 3. Overview soluções AWS para Big Data & Analytics

Amazon SageMaker

AWS SageMaker é um serviço totalmente gerenciado, que permite que cientistas de dados treinem e implantem modelos de machine learning com facilidade e em escala. O AWS SageMaker consiste em uma suíte de soluções que buscam abarcar todo o ciclo de vida do desenvolvimento de um modelo: desde a coleta e preparação dos dados de treino até a disponibilização de infraestrutura para servir o modelo.

Figura 6 – Serviços de Inteligência Artificial e de Machine Learning na AWS.



Fonte: Crie, treine e implemente modelos de aprendizado de máquina com o Amazon SageMaker (YouTube).

Abaixo listamos alguns dos recursos disponibilizados pela suite SageMaker para cada uma das etapas do desenvolvimento de um modelo.

- Preparação:
  - Feature Store
  - Clarify
  - Data Wrangler

- Construção:
    - SageMaker Studio
    - Autopilot
    - Built-in Algorithms
- Implantação e gerenciamento:
    - Real-time endpoints
    - SageMaker Model Monitor
    - SageMaker Pipelines

Cada um dos diversos módulos que compõem o Amazon SageMaker pode ser utilizado de maneira conjunta ou independente no desenvolvimento de um modelo de Machine Learning.

Amazon S3

Amazon Simple Storage Service (Amazon S3) é um serviço de armazenamento de objetos que oferece escalabilidade, disponibilidade, segurança e desempenho. O S3 permite aos usuários desde um armazenamento simples de arquivos, até sua utilização como ferramenta de backup, hospedagem de websites, aplicações mobile até a o uso em Big Data e Analytics, servindo como base para a construção de Data Lakes. O Amazon S3 provê funcionalidades de gerenciamento dos dados de maneira a permitir configurações de acesso de controle que atendam às distintas demandas de negócio e de compliance.

AWS Glue

AWS Glue é um serviço totalmente gerenciado de extração, transformação e carga de dados (conhecido por ETL: Extract, Transform and Load) que tem como objetivo viabilizar a preparação e carga de dados para Analytics. Dado se tratar de um serviço serverless (ou totalmente gerenciado), tarefas relacionadas à integração e gestão de dados são simplificadas, reduzindo-se substancialmente a sobrecarga associada ao gerenciamento da infraestrutura. Com poucos cliques é possível a criação e execução de tarefas de ETL a partir de dados armazenados na AWS. Os dados gerados por essas tarefas de ETL também podem ser catalogados e posteriormente acessados com facilidade por outros serviços da AWS.

O AWS Glue possui diversos componentes, os quais podem ser utilizados em conjunto ou de maneira independente:

- AWS Glue Data Brew
- AWS Glue Data Catalog: catálogo centralizado de metadados, que permite a consulta dos dados catalogados através de outros serviços da AWS, tais como Amazon Athena, Amazon Redshift e Amazon EMR.
- AWS Glue Crawlers: serviço utilizado para varrer dados armazenados na AWS e gerar automaticamente metadados que serão centralizados no Glue Data Catalog.
- AWS Glue ETL Jobs: permite a extração de dados de diversas origens, o processamento desse dado e posterior carga

AWS Athena

Amazon Athena é um serviço interativo de consulta (*query service*) que permite a consulta a dados armazenados em diversas fontes de dados e formatos utilizando SQL padrão. Por ser um serviço totalmente gerenciado, para o uso do AWS Athena não é necessário o provisionamento

ou configuração de servidores e a cobrança é feita unicamente pelas *queries* executadas.

Construído com o paradigma de separação entre armazenamento e processamento, o Amazon Athena não armazena o dado em si, sendo responsável pelos recursos computacionais (CPU, memória, etc.) para o processamento de dados armazenados em outros serviços. Athena foi construído baseado no Presto, um motor distribuído open-source originalmente desenvolvido pelo Facebook, sendo expandido pelos engenheiros da Amazon com uma série de recursos adicionais. Esses recursos adicionais, inclusive, facilitam e alavancam sua utilização em conjunto com os demais serviços da AWS, como o Amazon S3, AWS Glue Data Catalog e AWS DynamoDB. Sua integração, no entanto, não se limita aos serviços e tecnologias disponíveis na AWS: também podemos integrar ferramentas como Looker ou Tableau ao Amazon Athena.

O Amazon Athena possui suporte tanto para dados estruturados quanto dados não estruturados e semi-estruturados. A gama de formatos de arquivo suportados também é ampla: CSV, TSV e AVRO são alguns dos exemplos. Entre os formatos de arquivo colunares, há suporte para Apache ORC e Apache Parquet. Para o trabalho com dados não estruturados e semi-estruturados, destaca-se o suporte a Textfile e JSON.

AWS Aurora

Amazon Aurora é um motor de dados relacional (conhecido em inglês como RDBMS: Relational Database Management System) totalmente gerenciado, o que implica que, ao utilizar esse serviço, muitas das funções normalmente realizadas por um DBA (Database Administrator) são realizados automaticamente pela AWS. Com isso, tarefas custosas, relacionadas a provisionamento de hardware, configuração do banco de dados, *patching* e *backups,* são gerenciadas automaticamente.

O Amazon Aurora possui diversas características que o tornam altamente confiável, escalável e durável. Possui a capacidade de escalar quase instantaneamente, armazena os dados por padrão em 6 diferentes regiões e 3 Zonas de Disponibilidade por padrão, além de ter a capacidade de se recuperar muito facilmente de falhas.

O Aurora é compatível com o MySQL e o PostgreSQL. Apesar de sua compatibilidade com esses bancos de dados *open source*, o uso do Aurora possui diversas vantagens frente a eles. Para começar, seu desempenho é muito superior: ele é até 3 vezes mais rápido do que o PostgreSQL e até 5 vezes mais rápido do que o MySQL. O fato de suportar apenas bancos de dados open source tem ainda como vantagem a ausência de custos de licenciamento. Temos, assim, um banco de dados com elevado desempenho e baixo custo.

> Capítulo 4

Capítulo 4. Fundamentos de Engenharia de Dados

Tipos de Dados

Podemos dividir os tipos de dados com relação à sua estrutura:

- Dados estruturados: são dados cuja estrutura é bem definida. Esse tipo de dado tem como característica ser repetitivo, com a única diferença entre um e outro registro sendo o conteúdo desse registro, e normalmente são facilmente organizados em formato tabular (linhas e colunas). Exemplos de dados estruturados incluem as vendas de uma corporação, a folha de pagamentos, as transações financeiras de um banco ou as taxas de juros utilizadas no apreçamento de instrumentos de renda fixa.
- Dados não estruturados: são dados que não possuem estrutura fixa e pré-definida. Incluem-se nessa categoria texto, imagem, áudio, vídeo, entre outros.

Apesar de historicamente o gerenciamento e exploração de dados estruturados ter se desenvolvido mais cedo e mais rapidamente, estima-se que os dados estruturados representem apenas uma pequena fração dos dados gerados nas empresas, a grande maioria sendo de dados não estruturados.

Essa subdivisão entre tipos de dados é de extrema importância: dados estruturados e não estruturados possuem cada um suas particularidades, ferramentas e técnicas específicas para realização de pré-processamento, representação de atributos e modelagem.

Tipos de processamento de dados

Com relação aos tipos de processamento e padrões de consulta realizados por bancos de dados, esses são tradicionalmente classificados em dois tipos:

- Online Transaction Processing (OLTP): processamento de dados relacionado a transações (ou eventos) típicos do dia a dia de uma empresa: a realização de uma venda, um pedido feito a um fornecedor, a realização de uma transação financeira, etc. Nos casos de uso relacionados a OLTP, dados são inseridos ou atualizados baseados em alguma interação com um usuário. Por serem transações imediatas e por historicamente terem nascido num ambiente de transações financeiras, esse tipo de padrão de acesso e processamento ganhou o nome de Online Transaction Processing, mesmo quando suas aplicações superaram o das transações de negócio.
- Online Analytical Processing (OLAP): padrões de acesso OLAP surgiram com o nascimento de Data Analytics, ou seja, da necessidade das empresas de extrair valor e analisar a enorme quantidade de dados que se começou a gerar e armazenar nos sistemas de bancos de dados. Quando se busca responder a perguntas como:
    - Qual o mês do ano com mais vendas pra cada categoria de produtos?
    - Qual bebida é mais vendida em conjunto com a venda de carne?
    - Qual a receita e o lucro de cada uma das lojas?

  o padrão de acesso ao dado muda radicalmente. Ao invés de inserir ou atualizar registros individuais, consultamos (normalmente apenas poucos campos) de uma única vez milhares ou milhões de registros gerados ao longo do tempo e

calculamos estatísticas em cima desses campos (somas, médias, máximo/mínimo, etc.). O resultado dessas consultas já não é fruto direto da interação entre empresas ou entre empresa e cliente: os valores calculados retornam normalmente a uma equipe interna da empresa, que utiliza esses dados para a tomada de decisões estratégicas.

Apesar de muitas vezes a distinção entre OLTP e OLAP não ser tão bem definida, a tabela abaixo lista algumas das principais características de cada uma delas.

Figura 7 – Comparação entre OLTP e OLAP.

Table 3-1. Comparing characteristics of transaction processing versus analytic systems

Property	Transaction processing systems (OLTP)	Analytic systems (OLAP)
Main read pattern	Small number of records per query, fetched by key	Aggregate over large number of records
Main write pattern	Random-access, low-latency writes from user input	Bulk import (ETL) or event stream
Primarily used by	End user/customer, via web application	Internal analyst, for decision support
What data represents	Latest state of data (current point in time)	History of events that happened over time
Dataset size	Gigabytes to terabytes	Terabytes to petabytes

Fonte: Designing Data-intensive Applications (Martin Kleppmann).

Modelos de Dados

Um modelo de dados define uma forma particular de representar o dado. A existência de diversos modelos de dados já nos dá um indício de algo importante: não existe um modelo que seja, por si mesmo, superior ao outro. Cada modelo de dados possui vantagens e desvantagens a depender do caso de uso. Assim, o importante é conhecermos os principais modelos de dados, seus prós e contras e, como consequência, o melhor caso de uso para cada um deles.

Relacional: o modelo de dados mais amplamente conhecido é o relacional, proposto inicialmente em 1970 por Edgar Codd. Modelo para o qual foi desenvolvida a linguagem de consulta SQL (Structured Query Language), já era um modelo relevante na década seguinte e dominou a modelagem de dados por 25 a 30 anos. Neste modelo, dados são organizados em relações (chamadas tabelas em SQL) e cada relação é uma coleção não ordenada de tuplas (linhas no SQL).

Orientado a Documento: surgindo da insatisfação com algumas das características do modelo relacional, cresceu a partir de 2010 o interesse por modelos e tecnologias que trouxessem características como maior escalabilidade e permitisse uma modelagem mais dinâmica do dado. Essa representação permite um modelo mais flexível e autocontido do dado em um único documento (como um JSON ou XML). Como vantagens frente ao modelo relacional, podemos citar sua maior flexibilidade (*schema on read,* em oposição ao *schema on write* do modelo relacional), melhor desempenho (devido à localidade do dado) e que, para algumas situações, reflete melhor os modelos de dados utilizados na programação das aplicações (programação orientada a objetos).

Baseado em Grafos: nos modelos de dados baseados em grafo, todos os registros são organizados em vértices (também conhecidos como nós ou entidades) e arestas (também chamados relações ou arcos). Esse

modelo de dados é ideal para quando existem muitas relações n-n (*many-to-many*) para serem representadas, tais como redes sociais, relações entre páginas da internet ou mesmo representação de uma malha viária.

Referências

Para seguir aprendendo

Designing Data-Intensive Applications. Martin Kleppmann. O'REILLY Media.

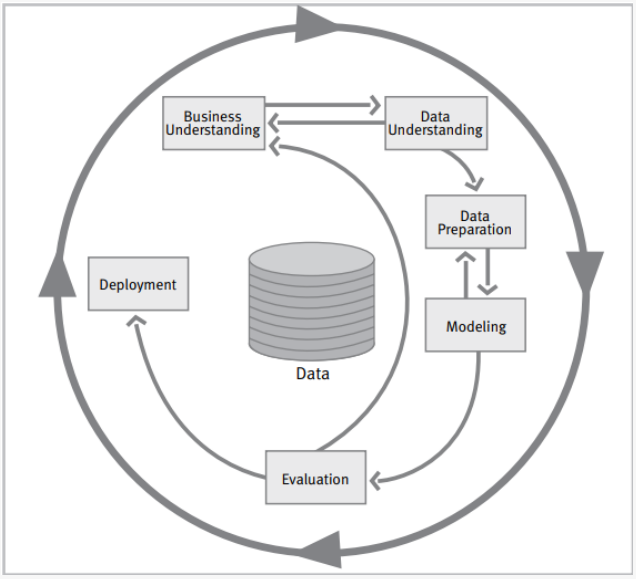
> Capítulo 5

Capítulo 5. Pipeline de Ciência de Dados

CRISP-DM

Um framework bastante utilizado na indústria para ilustrar as principais etapas de um projeto de ciência de dados - e como cada uma dessas etapas se relacionam entre si - é o Cross Industry Standard Process for Data Mining (CRISP-DM). Desenvolvido originalmente na década de 1990 por um consórcio de empresas, essa metodologia segue até hoje como uma das mais conhecidas e utilizadas.

Figura 8 – Framework CRISP-DM.



Fonte: CRISP-DM Twenty Years Later: From Data Mining Processes to Data Science Trajectories.

I. Entendimento do Negócio

Segundo preconizado pelo CRISP-DM, a primeira etapa de todo projeto de ciência de dados deve iniciar pelo entendimento do negócio. Essa etapa inclui o entendimento dos objetivos de negócio, a avaliação da

situação atual, os recursos disponíveis para o empreendimento do projeto, bem como potenciais riscos.

II. Entendimento dos Dados

A etapa seguinte consiste no entendimento do dado. Nessa etapa, tarefas como a coleta, descrição e exploração de dados, bem como a avaliação da qualidade dos dados são realizadas.

As etapas de entendimento de negócio e entendimento de dados não são sequenciais e independentes, mas se retroalimentam.

III. Preparação dos Dados

Nessa etapa é realizada uma série de tarefas de preparação para a modelagem e costuma ser uma das mais extensas fases de um projeto de ciência de dados. Ela inclui a seleção e limpeza de dados e a engenharia de atributos (*feature engineering*).

IV. Modelagem

Nessa fase, diversos modelos são construídos e testados, explorando-se variadas técnicas estatísticas e algoritmos. Cada um dos modelos não será somente construído, como também avaliado nessa etapa. O resultado de cada avaliação serve como insumo para nova iteração no processo de construção de modelos.

V. Avaliação

Enquanto a avaliação de modelo ocorrida na etapa anterior foca em métricas técnicas de modelagem, a fase de avaliação propriamente dita avalia o modelo de maneira mais ampla, considerando as particularidades e restrições de cada negócio e projeto.

VI. Implantação

Na fase de implantação, o produto resultado das etapas anteriores deve ser disponibilizado de maneira a poder ser consumido de acordo com as necessidades do negócio. A complexidade dessa etapa pode variar de maneira expressiva dependendo dos requisitos de negócio, da maturidade tecnológica da organização, dentre outros. Nessa etapa também se deve garantir que o modelo tem condições de receber manutenções e de ser monitorado.

Artigos interessantes

CRISP-DM Twenty Years Later: artigo que traz uma discussão sobre os principais frameworks de projetos de ciência de dados, bem como discussão sobre a evolução da disciplina ao longo das últimas duas décadas.

> Capítulo 6

Capítulo 6. Introdução ao SQL

Modelagem Relacional

O modelo de dados mais amplamente conhecido é o relacional, proposto inicialmente em 1970 por Edgar Codd. Modelo para o qual foi desenvolvida a linguagem de consulta SQL (Structured Query Language), já era um modelo relevante na década seguinte e dominou a modelagem de dados por 25 a 30 anos. Neste modelo, dados são organizados em relações (chamadas tabelas em SQL) e cada relação é uma coleção não ordenada de tuplas (linhas no SQL).

SQL

Juntamente com a definição do modelo relacional, Codd também propôs uma linguagem para manipulação de dados relacionais chamada DSL/Alpha. A partir da publicação de Codd, um grupo de engenheiros da IBM propôs uma linguagem chamada SQUARE, que consistia de uma versão simplificada do DSL/Alpha. Refinamentos posteriores de SQUARE levaram a outra linguagem, chamada SEQUEL, que finalmente deu origem ao que hoje chamamos simplesmente de SQL (Structured Query Language). Atualmente com mais de 40 anos, o SQL passou por diversas padronizações e evoluções pelo American National Standards Institute (ANSI) e é uma das linguagens mais importantes para a consulta e manipulação de dados.

Linguagem de Definição de Dados (DDL)

Linguagem de Definição de Dados (Data Definition Language - DDL) consiste no conjunto de instruções SQL utilizado para definir, alterar ou excluir objetos (e.g. tabelas). Abaixo seguem três das principais cláusulas:

- CREATE: utilizado na criação de novos objetos (tais como tabelas) no banco de dados;
- DROP: utilizado na exclusão de objetos do banco de dados;
- ALTER: utilizado para alterar objetos do banco de dados;

Linguagem de Manipulação de Dados (DML)

A Linguagem de Manipulação de Dados (Data Manipulation Language - DML) é o conjunto de instruções de SQL utilizado para adicionar, atualizar ou remover dados de uma tabela. Abaixo as principais cláusulas de DML.

- INSERT: utilizado para inserir novos registros em uma tabela;
- DELETE: utilizado para remover registros de uma tabela (eventualmente filtrando por determinada condição);
- UPDATE: utilizado para atualizar registros;

Linguagem de Consulta de Dados (DQL)

Compõe as instruções de consulta aos dados armazenados numa tabela. Uma consulta (ou *query*) SQL é composta por diversos componentes, conhecidos como cláusulas. Abaixo mostramos as cláusulas de uma consulta SQL, na ordem em que devem estar dispostas, juntamente com uma breve explicação de cada uma.

- SELECT: determina quais colunas devem ser retornadas pela consulta;
- FROM: determina quais tabelas devem ser consultadas e como devem ser unidas;
- WHERE: utilizada para filtrar dados de interesse;
- GROUP BY: agrupa linhas por colunas que tenham valores em comum;
- HAVING: filtra grupos de linhas;
- ORDER BY: utilizado para ordenar o resultado final por uma ou mais colunas;

> Capítulo 7

Capítulo 7. Processamento de Linguagem Natural

Introdução ao Processamento de Linguagem Natural

O conjunto de técnicas utilizado para tornar a linguagem humana acessível aos computadores é conhecido como Processamento de Linguagem Natural (*Natural Language Processing* - NLP).

Aplicações de Processamento de Linguagem Natural incluem:

- Classificação de textos: uma das tarefas com o maior número de aplicações, a classificação de textos busca categorizar documentos em determinadas classes de interesse. Exemplos de aplicação incluem a detecção de spam em e-mails e a análise de sentimentos;
- Recuperação de texto (*Information Retrieval*): tarefa que tem como intuito encontrar documentos relevantes (de acordo com alguma consulta) dentro de uma coleção de documentos. O exemplo mais comum é um buscador;
- Sumarização de textos: tem como objetivo gerar um novo documento, mais curto, a partir de um texto original mais longo, mantendo o sentido e conteúdo principal do documento original;
- Tradução (*Machine Translation*): tarefa de tradução de texto entre diferentes idiomas;

Pré Processamento

O tratamento computacional de textos exige uma série de pré-processamentos de modo a facilitar a sua análise, interpretação e modelagem. Abaixo citamos alguns dos pré-processamentos mais comuns.

Tokenização (*tokenization*): processo no qual se segmenta o texto original em pedaços menores (conhecidos como tokens), eventualmente já eliminando certos caracteres indesejados (como pontuação). Um exemplo de tokenização é a transformação de uma frase em uma lista de palavras, onde cada palavra é um *token*.

Remoção de *Stop Words*: determinadas palavras, extremamente comuns e frequentes em um idioma, podem ser de pouco valor no processamento de linguagem natural, dado que não acrescentam valor à análise. Essas palavras são conhecidas como *Stop Words*. Como exemplos de *stop words* em português podemos citar, entre muitas outras, as seguintes palavras: a, o, as, os, ao, do, da, em, um.

Remoção de pontuação: semelhante ao que ocorre com Stop Words, para diversas aplicações de NLP (mas não todas), as pontuações podem ser irrelevantes para a solução do problema. Nesses casos, parte do pré-processamento envolve a remoção de toda e qualquer pontuação.

Levar para minúsculo (*lowercasing*): outra técnica comumente utilizada no pré-processamento de textos é levar todo o texto para o minúsculo.

Técnicas mais avançadas de pré-processamento, que não trataremos nesse curso, incluem *stemming*, *lemmatization*, POS (*Part-of-Speech*) *tagging*, entre outras.

Aplicações interessantes

Algoritmo para leitura de atas de BCs (Itaú BBA): parceria entre as áreas de Macroeconomia, Tesouraria e Dados do Itaú, esse relatório descreve uma ferramenta sistemática para análise da comunicação do Banco Central do Brasil. Percorrendo atas do Copom entre 2016 e 2022, o objetivo do estudo foi a construção de modelos de NLP para uso como previsor dos próximos passos da política monetária.

> Capítulo 8

> Capítulo 9

> Capítulo 10