## Engenharia de dados

### Tópicos

* Criação de repositórios de dados para ML
  + Determinar os meios de armazenamento (Dentre serviços como S3)
* Identificar e implementar soluções de ingestão de dados
* Identificar e implementar soluções de transformação de dados

Engenharia de dados

Consiste em analisar os requisitos do usuário e desenhar programas para armazenar, transferir, transformar e estruturar dados com propósito de análise e relatório

Um exemplo de transformação de dados seria a substituição de valores em branco por rótulos textuais de melhor compreensão como não aplicável

**Ferramentas de ingestão de dados**

Auxiliam a extrair tipos de dados crus como logs

* **Kinesis Data Firehose**
  + Entrega streams de dados em tempo real para um bucket do S3 permitindo capacidades como transformação por meio de funções lambda
  + Coleta dados em tempo real para dentro de data lakes e serviços de análise
* **AWS Snowball**
  + Dispositivo para migração de dados on premise para dentro da cloud
* **Storage Gateway**
  + Permite um backup no S3 ou EBS a partir de dispositivos locais por meio de sistema NFS

As ferramentas focadas em dados na AWS podem ser categorizadas em

* **Armazenamento**
  + Onde os dados repousam após coleta ou extração por meio de serviços como S3
  + Deve ser decidido com base não somente nas características dos dados e sua natureza como também requisitos como periodicidade de recuperação
  + Também inclui EFS e FSx for Lustre
* **Integração**
  + A ideia geral é reunir de maneira centralizada dados provenientes de diferentes fontes
  + Geralmente contempla o processo de normalizar dados de fontes variadas e schema (Definição dos nomes de campos, seus tipos, convenções e ligações) diferente
  + Aqui se utiliza o AWS Glue
* **Visualização**
  + O último passo da engenharia de dados consiste em prover a visualização da maneira mais intuitiva e significativa quanto possível
  + Aqui temos o QuickSight
* **Warehouse**
  + Armazenamento de dados só que ao invés de brutos e crus, já filtrados e normalizados após coleta e transformação de diferentes fontes através do processo de integração
  + Aqui temos o Amazon Redshift

Enquanto um data lake armazena dados crus e não processados, sem propósito específico um data Warehouse já contém dados em formato normalizado e padronizado com propósitos específicos

### S3

O S3 é um serviço de armazenamento de dados generalista e de alta disponibilidade que trabalha com os conceitos de objeto (Contempla além do arquivo em si, seus metadados) e bucket (Localização lógica de um conjunto de objetos relacionados), além de chaves (Identificador único de um objeto dentro do bucket) e regiões (Localização física escolhida para o seu bucket por razões regulatórias ou de performance). No contexto de ML se integra a vários serviços relacionados e é normalmente usado por exemplo para armazenar arquivos de treinamento/conjuntos de dados (Formato CSV é o mais comum) em diversos formatos, além dos próprios modelos gerados, além de imagens, vídeos e outros tipos de objeto

Permite objetos de até 5 TB e adição de tags em cada objeto que servem como metadados customizados adicionais

Permite versionamento de arquivos, restringir deleção com MFA, replicação entre diferentes regiões

É possível realizar consultas interativas diretamente em objetos do S3 por meio do Athena

Acesso pode ser gerenciado por uma combinação de políticas do IAM e ACLs que são regras de controle aplicadas diretamente no bucket ou objeto

Suporta tanto encriptação no lado do servidor como no lado do cliente

Classes de armazenamento

* + São diferentes modelos de armazenamento com base em redundância, rapidez de recuperação, disponibilidade e custos
    - **Standard**
      * Para dados acessados com frequência, provê baixa latência na recuperação. Classe mais custosa
    - **Intelligent-Tiering**
      * Move automaticamente os dados de classe de armazenamento com base na frequência real de acesso dos dados
    - **Standard-IA**
      * Para dados acessados com menos frequência, mas que exigem acesso rápido quando solicitados
    - **One Zone-IA**
      * Enquanto outras classes armazenam os dados com redundância em três AZs, essa utiliza uma única AZ. Sendo assim, é recomendado para dados não críticos facilmente recriáveis
    - **Glacier**
      * Classe focada em arquivamento e backup de longo prazo
        + Instant Retrieval

Dados raramente acessados, porém exigem recuperação em milissegundos

* + - * + Flexible Retrieval

Dados cujo acesso é feito anualmente e podem ser recuperados de maneira assíncrona. Para recuperação de backup

* + - * + Deep Archive

Para cenários de armazenamento de longuíssimo prazo por razões regulatórias como é o caso de serviços financeiros, jurídicos ou médicos

* + - **Outposts**
      * Para armazenar no on-premises com interface do S3

Cada bucket tem um identificador único globalmente, o versionamento permite manter um histórico de diferentes versões de um mesmo arquivo objeto

É possível interagir com os serviços em sua conta por meio da linha de comando através do CLI. Após fazer o download é preciso gerar as chaves de acesso do usuário através do qual será feita a autenticação

Para o caso de múltiplas contas ou usuários é possível usar a opção profile para criar e acessar especificamente

Regras de ciclo de vida

Realiza a transição automática entre classes de armazenamento com base no tempo de retenção do objeto

A regra é declarada usando um json e então pode ser anexada por meio do respetivo comando via CLI

### AWS Glue

Serviço de integração que permite mais de setenta diferentes fontes de dados, é serverless

Permite incorporar dados de fontes diversas para o data lake ou automaticamente mapear a estrutura de dados de arquivos não estruturados a partir de um data lake existente

Resumidamente serve aos propósitos de extração, transformação e integração (Ou ETL)

Sua distinção no que tange a ETL com relação a EMR é que é serverless e possui uma série de transformações predefinidas, mas também permite scripts customizados escritos com Spark, Python e Scala

Terminologia comum

* **Data Catalog**
  + Contempla o repositório de metadados, bem como a definição dos Jobs e outras informações de controle necessárias ao ambiente do Glue
* **Classifiers**
  + São os algoritmos próprios para cada tipo de formato de arquivo (Como por exemplo JSON ou XML) que realizam a leitura dos dados crus no arquivo de acordo com sua extensão e geram o schema padronizado no Glue
* **Crawler**
  + Mecanismo de extração de schema de arquivos não estruturados de forma automatizada
* **Jobs**
  + Funções de processamento agendáveis ou acionadas via evento que aplicam transformações ou agregações nos dados de origem e salvam no repositório de destino por meio de scripts
* **Table**
  + Contém os metadados mapeados de um dado conjunto de dados como a estrutura dos dados em termos de identificadores das colunas bem como seus respectivos tipos
* **Script**
  + Código escrito com PySpark ou Scala utilizado para os propósitos de ETL

Precificação

Taxas mensais pelo armazenamento dos metadados além de cobrar pela execução dos crawlers e Jobs de ETL

DataBrew

Permite inspeção e exploração visual dos dados por parte dos cientistas de dados. Útil na etapa de engenharia de Feature

Descoberta e organização

Permite unificar dados de fontes diversas por meio de um catálogo centralizado

Criamos uma base de dados no Glue por meio do console, em seguida criamos um crawler configurando sua fonte de dados com base em um bucket do S3 (E indicando se será escaneada uma pasta em específico ou todas as subpastas), precisaremos de uma role do IAM e em seguida definiremos a saída como uma tabela na database do Glue com a qual estamos trabalhando (Ou então criar uma nova). Após isso executaremos o crawler e ele criará automaticamente o schema através do mapeamento do CSV dentro do bucket (Também podemos definir um agendamento por meio de expressão Cron além da opção manual sob demanda)

O guia acima demonstra um cenário clássico de criação de um banco de dados estruturado automaticamente a partir de arquivo CSV

Os crawlers escaneiam os objetos e criam o catálogo de metadados de maneira automatizada, facilitando o processo de descoberta de dados

Transformar, preparar e limpar dados para análise

Permite visualizar nossos dados e preparar nossa rotina de ETL de maneira visual por meio de um editor drag and drop

Abrimos o console do DataBrew e criamos um novo projeto e com ele será gerada uma nova role do IAM, será aberto um editor visual com nosso dataset carregado e teremos um menu de opções de operações que poderão ser aplicadas nos dados como conversão para uppercase, etc. Ele cria uma receita com uma série de Steps que serão aplicados aos dados e dessa forma dispensamos uma abordagem programática ou usando scripts. É uma ferramenta do tipo low code

Após criar a receita de transformação configuramos a saída e poderemos executar um job com base nela a qualquer momento

Construir pipelines de dados

Podemos reagir a eventos ou agendar de forma periódica a execução de Jobs de ETL

No Glue Studio podemos criar Jobs a partir de scripts do Apache Spark

Precisamos prover acesso ao bucket onde estão localizados nossos dados e também naquele em que está armazenado o script por meio de uma policy na role

Esses Jobs servem para realizar processamentos como agregações e outros tipos de transformações analíticas (De acordo com a lógica do script selecionado) cujo resultado será salvo em um bucket de saída

### Athena

Serviço interativo de query no qual podemos aplicar consultas SQL diretamente no S3. Possibilita também comandos de transformação e junção. Utiliza o Presto como mecanismo de consulta internamente

Precificado pelas consultas realizadas (Com base na complexidade da query e quantidade de dados retornados por ela)

Serviço serverless que nos dispensa da necessidade de manter a infraestrutura de uma base de dados permanente com o propósito único de realizar consultas em nossos Data lakes

No console da Athena devemos configurar um prefixo em um bucket para que sejam armazenados os resultados das consultas. Lá também criaremos de maneira volátil uma base de dados e a tabela (Que servirão apenas como uma espécie de ponteiro ou referência para os dados nos arquivos ao invés de conter os dados em si assim como no Hive). No comando DDL de criação da tabela haverão cláusulas especiais para definir a localização do bucket e prefixos cujos arquivos serão consultados, bem como os delimitadores utilizados para mapear o schema

Permite consultas ad hoc em dados do S3 por meio do Glue e HiveQL para definição das tabelas voláteis por meio de um catálogo

### Batch

Permite execução de Jobs de processamento em lote na AWS (De forma escalável e com base em container). Faz a provisão dinâmica de recursos necessários de forma otimizada para atender os requisitos das funções submetidas para execução

Pode ser usado em conjunto com o Fargate ou os outros tipos de oferta de computação como EKS ou instâncias

Trabalha com o conceitos de filas de processamento que podem ser priorizadas com base em critérios

Amplamente utilizado para treinamento de modelos de ML

Possui monitoramento e logs integrados ao cloudwatch

### Step Functions

Serviço dedicado a orquestração de processamento em aplicações distribuídas ou microsserviços por meio do conceito de workflow

Sistema de orquestração de workflow de funções do tipo serverless com mecanismos de controle operacional embutidos

O workflow é definido por meio de um JSON (Também pode ser feito visualmente pelo console) contendo os ARNs das funções (Lambda) envolvidas no fluxo de processamento (Ou seja, a pipeline de funções/passos). Com isso criamos uma máquina de estados (State Machine) que contém o fluxograma de nosso processamento, passos, ramificações lógicas, definições de tratamento de erro, paralelizações, etc. O workflow também pode ser montado integrando diretamente componentes de arquitetura com outros serviços da AWS (Como ler de tabelas do DynamoDB e publicar mensagens no SNS sem necessidade de um Lambda como intermediário)

O console nos permite disparar execuções teste do fluxo e monitorar todo o passo a passo executado de acordo com nossa máquina de estados

* **Standard**
  + Ideal para fluxos de longa duração (Até um ano)
* **Express**
  + Recomendado para fluxos de alto volume, baseados em eventos como ingestão de sensores IoT ou streaming com curta duração (Até cinco minutos)

O serviço estrutura e lida com aspectos secundários e transversais desse tipo de processamento sequencial e dividido como

* **Orquestração**
  + Controla a interação entre as diversas funções lambda envolvidas no workflow, executando-as na ordem especificada e repassando o resultado da anterior para a próxima até conclusão do fluxo
* **Ramificação**
  + Diz respeito a segmentação lógica que um fluxo pode ter com base na lógica de negócio. Com base em condições a função adequada aquele ponto será invocada
* **Tratamento de erros**
  + Permite definir quais funções serão chamadas em cenários de exceção
* **Cenários de loop**

Com isso permite ao desenvolvedor maior foco na lógica de cada passo de seu fluxo dispensando seu foco em escrever código boilerplate dedicado aos cenários acima citados

É possível por meio da integração com SageMaker executar pipelines de machine learning

Integra também diretamente ao Glue, Athena e EMR sem necessidade de uma função Lambda intermediária

Por meio da integração com tantos outros serviços permite ser o orquestrador para automatização de processos completos de desenvolvimento e implantação de casos de uso envolvendo ML

Tipos de passos

* **Tarefa**
  + Unidade de trabalho em um estado que pode envolver a invocação de uma função Lambda ou serviço AWS diretamente. Também pode ser uma aprovação humana manual por meio de e-mail por exemplo
* **Escolha**
  + Condicional lógica para ramificar o processo por meio de operadores que formam condições para dois ou mais caminhos possíveis a seguir no fluxo de trabalho
* **Espera**
  + Tempo arbitrário assinalado em que o processamento fica em estado de espera até o próximo passo
* **Paralelização**
  + Permite paralelizar passos que podem ser realizados simultaneamente e sua saída será uma combinação dos resultados de todas as ramificações
* **Mapeamento**

Precificação

Com base no número de transições entre estados para cada função

## Análise exploratória de dados/EDA

Investigação inicial em um conjunto de dados visando descobrir padrões, detectar valores discrepantes, testar hipóteses e checar qual o modelo mais adequado a ser aplicado nestes dados

Auxilia a compreender a estrutura subjacente dos dados, bem como detectar erros, valores discrepantes, relações entre variáveis, etc.

Normalmente consiste em carregar um dataframe com base no arquivo fonte (Por exemplo um CSV), a biblioteca pandas do Python é amplamente utilizada para tal propósito. Ao carregar e exibir alguns registros de amostra (DataFrame.head) daquele conjunto já podemos observar alguns padrões como a convenção de identificadores no cabeçalho para o nome das colunas (Se camel case, snake case, etc.), tipos de enumeração e valores aceitos em determinada coluna, delimitadores utilizados em campos do tipo array, valores faltantes ou nulos, se os valores textuais estão em maiúsculo ou minúsculo, dentre muitos outros pontos

O primeiro passo então é estabelecer a normalização dos dados, equalizando uma única convenção para atributos que estejam eventualmente discrepantes, por exemplo optar pelo pascal case para os nomes das colunas no cabeçalho, para isso será preciso alterar as que já estejam separadas por sublinhado fazendo com que haja uma padronização e assim facilitar a futura manipulação dos dados

Comandos úteis do Python

* DataFrame.columns
  + Retorna o vetor com as colunas daquele conjunto de dados
* str.lower
  + Converte o texto para minúsculo
* str.replace
  + Substitui um dado caractere em todo o texto por outro (Passado no segundo argumento)
* DataFrame.dtypes
  + Trás os tipos de dados de cada coluna do conjunto de dados
* DataFrame.info
  + Mostra alguns metadados daquele dataset como índice das colunas, identificador, contagem de valores não nulos, tipo de dados, dentre outras coisas
* df.isna().sum()/df.isnull().sum()
  + Exibe a contagem de valores ausentes/nulos para cada coluna em nossa base
* DataFrame.fillna
  + Preenche os valores ausentes ou nulos. Produz uma nova série de dados (Ou array) de acordo com o que é passado como argumento e cujo retorno deve atribuído a série original
* pandas.unique
  + Como base no índice de uma coluna (Sua respectiva série/array) retorna as possibilidades de valor único dentro do conjunto de dados
* pandas.concat
  + Permite mesclar DataFrames de diferentes fontes em um único
* DataFrame.apply
  + Análogo ao map de stream do Java. Recebe uma Function de transformação do valor original em um novo com base em uma lógica de mapeamento (DE-PARA)
* DataFrame.duplicated
  + Retorna os índices dos registros duplicados
* Series.drop\_duplicates
  + Elimina os registros duplicados no conjunto
* seaborn.boxplot
  + Desenha uma caixa contendo a distribuição de valores dentro do conjunto para uma ou mais colunas quantitativas. Exibe a variação
* seaborn.histplot
  + Renderiza um histograma
* seaborn.countplot
  + Renderiza um gráfico de contagem que consiste em uma linha horizontal com a distribuição das diferentes possibilidades de ocorrência de uma coluna e uma linha vertical com a contagem de ocorrências de cada um dos valores distintos. A intersecção entre o valor e sua respectiva contagem gera uma visualização em barras
* seaborn.barplot
  + Exibe um gráfico de barras para quando fazemos uma análise bivariada onde um atributo é categórico e o outro numérico, exemplo seria permanência no emprego e salário
* seaborn.scatterplot
  + Gráfico de dispersão onde podemos realizar uma análise multivariada considerando a correlação entre dois atributos e distinguir os registros individuais com base em um terceiro atributo categórico cujas categorias serão diferenciadas pela cor
* seaborn.swarmplot
  + Gráfico de dispersão onde o eixo principal analisado são as categorias de um campo de enumeração textual (Categórico)
* seaborn.jointplot
  + Exibe mais de um gráfico para duas variáveis em uma mesma saída
* msno.matrix
  + Exibe uma matriz em um formato gráfico com a distribuição de valores nulos e faltantes por atributo em nosso conjunto de dados
* DataFrame.value\_counts
  + Exibe a porcentagem de ocorrências de cada valor distinto para uma determinada coluna
* Series.value\_counts
  + Exibe a contagem de ocorrência de cada valor distinto de uma determinada coluna selecionada
* DataFrame.shape
  + Retorna o número de linhas e colunas respectivamente em dado conjunto de dados
* DataFrame.groupby
  + Dado uma coluna de escolha gera para cada valor distinto da enumeração os valores (Por exemplo médios) para cada uma das demais colunas. Exemplo para cada estado empregatício visualizar a média de idade, etc.
* pandas.crosstab
  + Usado para criar um gráfico em barras com a correlação entre duas variáveis do conjunto de dados, útil para encontrar um previsor forte entre os atributos para determinada classe

Iremos eleger a coluna de saída ou variável alvo de nossa análise preditiva (Por exemplo o preço de um carro)

Podemos exibir um histograma da variável alvo com base no conjunto de dados (seaborn.histplot) para observar a frequência e distribuição de valores e com isso obter uma melhor compreensão do padrão esperado de valores em nossa base (Por exemplo um histograma com base na coluna de preço do carro nos faz saber se a maioria dos veículos na base tem alto ou baixo valor)

O histograma da variável alvo nos permitirá saber se a distribuição de valores para a variável alvo tem uma grande discrepância entre os registros (O que poderá ser inferido visualmente com base na diferença entre o tamanho das barras em cada faixa de valor). Ao encontrar uma discrepância desse tipo podemos ser obrigados a realizar uma transformação para colocar os dados em uma mesma grandeza ou mesmo extrair um subconjunto com valores mais harmônicos para ser utilizado como base de aprendizado para nosso modelo

O preenchimento de valores nulos ou pendentes é geralmente um próximo passo, a prática mais comum é preencher com valores médios os campos numéricos e nas colunas com rótulos textuais inserir o valor mais frequente no conjunto como um todo

Tipos de dados

No que tange aos tipos de dados temos

* **Qualitativo**
  + Dados textuais/String de natureza descritiva ou de categorização que não podem ser alvos de operações matemáticas
    - **Nominal**
      * Dados não mensuráveis e de mesma importância/significado como cor favorita
    - **Ordinal**
      * Dados nos quais é possível inferir uma priorização, ordenação ou importância como por exemplo avaliação de um restaurante. Atributos que embutem algum tipo de julgamento de valor
* **Quantitativo**
  + Dados numéricos
    - **Discreto**
      * Campos numéricos que apesar de não serem categorias, tem geralmente uma faixa de valores finita e previsível como é o caso de número de filhos em uma família e tamanho de sapato
    - **Contínuo**
      * Campos numéricos de maior grandeza, faixa de valores mais ampla e menos previsíveis como é o caso de salário ou lucro de uma empresa

Importante ressaltar um dado que tem a característica de categorização ou enumeração é qualitativo, mesmo que representado por números

Tipos de variáveis

* **Dependente**
  + Uma coluna de saída ou cujo valor estamos tentando prever, exemplos são se uma pessoa é afetada por COVID ou não, o preço esperado de um veículo
* **Independente**
  + Colunas individuais de entrada, atributos nos quais nos baseados para realizar a previsão da variável alvo
  + Também conhecidas como features
* **Categórica**
  + Colunas qualitativas
* **Contínua**
  + Colunas quantitativas

É importante conhecer os tipos de variáveis e as classificações dos tipos de dados, pois isso permitirá aplicar a metodologia adequada para cada atributo durante o pré processamento e análise exploratória

Análise estatística

Processo no qual coletamos os dados e analisamos visando encontrar padrões de maneira não enviesada. O objetivo é converter os dados brutos em insights/percepções úteis

* **Análise descritiva**
  + Envolve a sumarização por meio de gráficos, tabelas ou outros meios visuais. Diluímos a complexidade do conjunto em alguns métricas chave como média, contagem, etc.
  + Diz respeito a métricas mais simples e básicas como média, soma, distribuição de frequência, dentre outras que não geram por si só previsões futuras ou inferências mais complexas
  + Outras métricas são medida de variabilidade (Diferença entre maior e menor valor), tendência centrais (Média e mediana por exemplo), desvio padrão
* **Análise inferencial**
  + Visa generalizar um padrão encontrado para toda a população
  + Exemplo: Pega uma cidade específica de um país, descobre a média de salário da população e generaliza para o restante do país. Como é possível denotar no exemplo nesse tipo de análise usamos as conclusões obtidas a partir de uma amostra de indivíduos e generalizamos para o restante da população, visto que a análise de toda a população é inviável ou muito custosa
* **Análise preditiva**
  + Análise as tendências passadas para prever eventos futuros
* **Análise prescritiva**
  + Prescreve uma decisão ou toma ação pontual com base no resultado da análise
* **Análise exploratória**
* **Análise causal**
  + Determina as relações de causa efeito

Normalmente a análise estatística envolve os seguintes passos: Coleta dos dados, compreensão das relações entre os dados, criação e validação do modelo, geração de previsões

Outliers

Valores atípicos ou que fogem do padrão do que é observado no conjunto de dados como um todo. Por exemplo se a quantidade de gols da maioria dos jogadores na lista de artilheiros é entre 11 e 20, então Kane e Haaland são outliers por terem feito 30 e 36 gols cada respectivamente

A identificação dos outliers é importante pois podem representar erros no conjunto de dados ou causar anomalias e discrepâncias a depender do tipo de análise e modelo aplicado ao dataset

1. **Aquisição** dos dados. O DataFrame no pandas geralmente é criado a partir de arquivos com extensão CSV, DB ou XLSX (A depender do formato precisamos de bibliotecas adicionais como SQLAlchemy ou openpyxl para os dois últimos respectivamente) que podem ser provenientes de um Bucket (De um Data Lake no S3)
2. Podemos mesclar dados de diferentes fontes com pandas.concat. Devemos eliminar os registros duplicados do DataFrame, isso porque quando considerados na análise irão corromper os resultados, como por exemplo ao considerar duas vezes o preço de um veículo em uma soma
3. Uma maneira de identificar outliers é por meio de um box plot (Gráfico contendo a distribuição/dispersão de valores para uma coluna do dataset)
   1. O gráfico de caixa possui parâmetros (Q1, Q2 e Q3 que são obtidos a partir da função quantile/média e que servem como faixa de valor referência aceitável para registros “normais“) e infere limites a partir dos quais os indivíduos/registros serão considerados outliers
   2. Uma vez identificados usando a abordagem acima devemos excluir os outliers do conjunto
4. A biblioteca missingno nos permite exibir uma matriz com a distribuição de valores nulos ou ausentes no conjunto de dados de maneira visual. O método de imputação consiste em preencher as lacunas ausentes (Nas colunas nulas ou sem valor) com valores obtidos estatisticamente. A estratégia de imputação depende do tipo de coluna, para campos categóricos geralmente é utilizado o valor mais frequente (Ou mode) ou se há um valor na enumeração estabelecido para representar default ou não especificado (Como NA ou unknown) também pode ser aplicado, já para atributos numéricos se utiliza a mediana (Valor do meio ao ordenar os valores do menor ao maior). Sendo assim é preciso analisar o tipo de cada coluna que possui ocorrência de nulos individualmente para traçar a melhor estratégia
5. O próximo passo é **correção de erros** no conjunto de dados. Isso consiste em normalizar o preenchimento de valores, pois um mesmo valor pode estar digitado de maneiras diferentes e até mesmo encontraremos variações da mesma coisa devido a erros de digitação/gramaticais. Para averiguar isso basta visualizarmos os valores únicos encontrados para cada coluna e criar uma função para mapear e corrigir as discrepâncias (Ela pode consistir em condicionais fazendo o match do valor antigo e retornando o novo padronizado ou então um mapa/dicionário com a correspondência dos valores errôneos e o valor novo correto a ser aplicado), em seguida aplicamos a função transformando a série original para aquela coluna com base na função de mapeamento
   1. A correção de erros no conjunto de dados consiste em não somente encontrar eventuais erros de digitação ou gramaticais e variações linguísticas de um mesmo valor, como também mapear eventuais valores que podem ser combinados em uma mesma categoria
6. A fase de **transformação** dos dados, dentre outras coisas, consiste em mapear os dados de tipos mais genéricos e abstratos para mais específicos, de maneira a facilitar nossa posterior análise, um exemplo seria converter uma coluna que contém datas, porém está no formato String para datetime de fato para que seja possível aplicar operações específicas de data e processamentos mais assertivos. Um outro exemplo clássico é uma coluna de preços, em que encontraremos valores em escalas, representações ou moedas diferentes e precisaremos padronizar para um único formato numérico, para esse caso pode ser necessário uma fase intermediária como criar uma nova coluna auxiliar com o tipo de moeda com base em uma função aplicada na coluna original de preço, em seguida eliminamos o símbolo monetário da coluna original, preenchemos os preços ausentes com 0 e por último aplicamos uma função de conversão cambial na coluna de preço usando como base a nova coluna de moeda (Podemos também criar novas colunas com o resultado do processamento de maneira a manter os valores originais)
7. **Codificação**
   1. Envolve transformar um dado categórico textual em um formato numérico, a estratégia de encoding aplicável depende se o atributo em questão é nominal ou ordinal
      1. Para campos do tipo ordinal (No qual os valores distintos possuem uma hierarquia ou julgamento de valor embutido) a abordagem aplicada é o uso do LabelEncoder (Utilitário da biblioteca scikit-learn que converte cada valor distinto da coluna ordinal em índices inteiros de 0 até o número de classes -1)
      2. Já para os campos nominais usaremos o OneHotEncoder, sua abordagem consiste em converter as colunas de ocorrência de cada valor distinto em uma tabela com campos booleanos para cada possível valor onde teremos sempre 1 para o valor encontrado em si e 0 para os demais. Isso também é possível com a função get\_dummies
8. **Dimensionamento/Scaling**
   1. Consiste em padronizar a escala numérica usada nos diferentes campos (Ou features) de maneira que estejam em uma mesma dimensão. Essa otimização é necessária para aplicação de modelos de ML posteriormente, visto que não funcionarão com a melhor assertividade caso haja atributos contendo números em faixas muito discrepantes
   2. Os métodos disponíveis são
      1. StandardScaler/Z-Score
      2. MinMaxScaler

A análise univariada consiste em se debruçar em um único atributo por vez, por exemplo dada uma base de funcionários e ex-funcionários podemos observar a distribuição em porcentagem com relação a quem ainda permanece ou não na empresa com base na respectiva coluna, podemos também renderizar um gráfico de contagem para observar a distribuição por idade

Já a análise bivariada é seleção e uso de duas variáveis em diferentes técnicas e abordagens com intuito de inferir a correlação entre elas, um exemplo no mesmo domínio do anterior seria averiguar quais departamentos tem maior saída de funcionários

Conjunto de dados desbalanceados caem no cenário de sobreamostragem ou subamostragem, o desbalanceamento consiste em ter um número muito discrepante de registros com ocorrência de uma classe em detrimento de outra dentro do dataset. A biblioteca imblearn facilita o balanceamento de conjunto de dados desbalanceados. O processo de balanceamento consiste em equalizar o número de ocorrências para cada classe através de diferentes abordagens e técnicas

Dimensionalidade

Diz respeito ao número de atributos presentes no conjunto de dados, não só originalmente, como também as colunas adicionais incluídas ao longo do processamento prévio e análise exploratória. Um excesso de dimensões poderá causar penalidades no desempenho e acurácia do modelo de ML e é aí que entra o conceito de redução de dimensionalidade que pode ser feito das seguintes maneiras

* Remoção de colunas duplicadas e/ou desnecessárias ao processamento
* Combinação de múltiplas colunas em uma única
  + Exemplo, os atributos altura, peso e idade podem ser calculados em um único denominado BMI (Ou IMC)
  + Essa abordagem requer conhecimento de negócio/domínio para decidir quais campos podem ser combinados/reduzidos e seguindo qual lógica. Na ausência desse conhecimento específico é possível alternativamente o uso de uma abordagem matemática (Conhecida como PCA) para compressão do conjunto original em um Dataset bidimensional
  + A técnica de PCA (Principal component analysis) requer a prévia padronização de escala dos dados (Scaling)
  + Outra técnica possível é a LDA que é proveniente da aprendizagem supervisionada

### Amazon QuickSight

Ferramenta de BI completamente gerenciada em cloud e serverless. Permite criar dashboards a partir de várias fontes de dados em nuvem ou locais

### Amazon EMR

Suporta cargas de trabalho com base no framework de processamento de big data Spark e demais componentes Hadoop como o Hive, Pig e Jobs do tipo MapReduce. Permite a implantação de clusters virtuais com base nos serviços da Amazon como o EC2 e S3

É um cluster gerenciado baseado no ecossistema Hadoop que funciona com base no EC2. Permite o uso de ferramentas como Spark, Presto, Flink, Hive, Pig e Zeppelin. Sua variação do sistema de arquivos distribuído HDFS permite sincronização nativa com o S3

No contexto de ML será útil para executar processamento e ETL de dados massivos utilizando Spark. Recomendado para a etapa de pré-processamento quando os dados em questão tem escala de petabyte e o time tem algum conhecimento e experiência com big data, quando esse não é o caso utilizar o SageMaker Processing

Permite uma flexibilidade maior quando as parametrizações e possibilidades da infraestrutura de computação EC2 subjacente e isso pode tornar uma opção mais efetiva em termos de custo para cenários de big data e/ou processamento mais denso

## Modelagem

Engloba as questões conceituais sobre Machine Learning

### Introdução a ML

Nada mais é do que a aplicação de métodos matemáticos e estatísticos de maneira a derivar padrões e previsões a partir de conjuntos de dados existentes com intuito de automatizar/otimizar a tomada de decisão em sistemas de informação digitais. Um algoritmo de ML aprende com base em exemplos (Um Dataset existente) gerando um modelo treinado capaz de receber um input e gerar uma saída automatizada e assertiva. Sem ML na abordagem tradicional de engenharia de software o desenvolvedor provê os dados, bem como o código escrito utilizando regras derivadas manualmente com o intuito de tomar uma decisão ou gerar uma saída para um dado input, já com ML selecionamos um algoritmo mais adequado ao problema/domínio em questão e com base em dados históricos de treinamento/exemplos será gerado um modelo a partir do qual novos inputs serão submetidos (Para realizar uma classificação, previsão, etc.)

Na detecção de Spam por exemplo, podemos elaborar manualmente regras com base no e-mail do remetente, assunto ou conteúdo de maneira a classificar se um e-mail é ou não spam. Com base nessas regras podemos então escrever código para classificar novos e-mails recebidos e descarta-los caso sejam identificados como spam, essa seria uma abordagem sem ML. No caso de ML, nós carregaríamos um base exemplo histórica de e-mails recebidos e já classificados como spam ou não, em seguida selecionaríamos um algoritmo (Que consiste em determinado método/abordagem lógica, matemática ou estatística já preconcebida) e construiríamos um modelo capaz de realizar por nós a classificação. Repare que ambas as abordagens são intrinsecamente humanas, visto que mesmo os algoritmos de ML foram concebidos por seres pensantes, a diferença ao aderir ML é que não reinventamos a roda para problemas comuns e usamos o poder computacional atual a nosso favor de maneira a automatizar uma tarefa exaustiva/humanamente não escalável

Um bom modelo de ML é capaz de inferir deduções assertivas em dados não vistos previamente

Tipos de ML

**Supervisionada**

Na qual provemos exemplos já rotulados com os atributos previsores (Ou features /independentes) e o atributo alvo já rotulado/preenchido (Dependente ou target)

O conceito de supervisão provém do fato de que nós (Humanos fazendo o cruzamento dos algoritmos disponíveis e os dados do domínio/problema que temos em mão) controlamos os conjuntos de dados providos como base de aprendizado da máquina. Ainda é nossa responsabilidade obter uma amostragem significativa de registros validados, extrair as características relevantes ao contexto, fazer o processamento prévio e sanitização dos dados, bem como outras atividades sem as quais não é possível extrair resultados significativos

A premissa para aprendizagem supervisionada é possuir dados rotulados de maneira fundamentada ou assertiva (Ground truth) com a qual o modelo possa aprender e obter um sistema de inferência generalizado para não somente decorar os dados com os quais aprendeu, mas também e principalmente novas ocorrências

Dentro da aprendizagem supervisionada temos subcategorias

* Classificação binária
  + Por exemplo detectar se uma transação é ou não fraudulenta
* Classificação com múltiplas classes
  + Por exemplo para identificar a posição de um jogador de futebol
* Regressão
  + Para previsão de valores numéricos como preços de imóveis

**Não supervisionada**

Na qual temos somente as features. Nesse tipo buscamos obter resultados significativos sem previamente prover exemplos corretos e validados

Nesse tipo de aprendizagem não há dados previamente rotulados nos quais se basear e precisamos inferir padrões e obter insights a partir dos dados

**Aprendizado por reforço**

Sistema otimiza a si mesmo com base em tentativa e erro, bem como experiência

Essa abordagem funciona com base em punições e recompensas enquanto o agente interage iterativamente com o ambiente e extrai aprendizado e excelência em suas ações. Um exemplo seriam os aspiradores de pó robóticos

Cabe ressaltar que não são os tipos ou natureza dos dados que indicam se o tipo de aprendizado será ou não supervisionado, mas sim a presença ou não de rótulos

Fluxo de ML

1. Obtenção de uma base/fonte de dados
2. Amostragem
3. Divisão entre base de treinamento/aprendizado e teste
4. Aplicação do algoritmo de ML para criação do modelo
5. Predição e avaliação
   1. Consiste em submeter os registros de teste ao modelo construído com base nos registros de treinamento e avaliar a assertividade dos resultados comparado aos demais algoritmos
   2. Aqui ocorrem otimizações cíclicas no dataset e na parametrização do algoritmo de maneira a incrementar o percentual de acerto

### Regressão linear

Conceitual

A regressão linear visa estabelecer uma reta preditora para valores quantitativos (Assume relação linear entre os atributos independentes e o rótulo), a partir de registros de exemplo uma vez traçada esta reta preditora comparamos os valores reais da base de treino com os valores previstos na reta pelo modelo, a diferença (Distância) entre cada valor real e previsto irá compor o residual ou perda (Nosso objetivo é reduzir a perda ao mínimo possível)

O coeficiente de correlação estabelece (r) indica o grau de relação entre duas variáveis X (Independente) e Y (Dependente) quantitativas, por exemplo altura do pai com relação à altura do filho. Uma correlação positiva indica que quanto maior X maior Y (Uma reta crescente no gráfico) e uma correlação negativa o contrário (Conforme aumente X o valor de Y diminui). Quando não há correlação entre as variáveis (Ou ao menos essa correlação não é linear) o valor de r tende a estar próximo de 0, quanto mais próximo de 1 ou -1 estiver maior a correlação perfeita positiva ou negativa respectivamente entre ambas

Algoritmo de ML supervisionada. Visa a previsão de um atributo alvo numérico

Ao aplicar qualquer algoritmo geralmente temos um DataFrame com o conjunto de dados completo (Atributos previsores e alvo e ainda não dividido em aprendizado e teste), sendo assim precisamos primeiramente dividir em X (Todo o conjunto exceto a variável alvo) e Y (Apenas os registros com a variável alvo como única coluna), em seguida dividimos o conjunto entre aprendizado e teste (Usando a função train\_test\_split da biblioteca sklearn)

No caso da regressão linear importamos o LinearRegression e invocamos o fit

A avaliação de desempenho do modelo geralmente consiste conceitualmente em uma tabela contendo os valores reais para o atributo alvo com base nos registros de teste e uma segunda coluna com os valores previstos pelo modelo (Como estamos falando de aprendizagem supervisionada temos como comparar ambos). No caso da regressão linear como lida com atributo dependente numérico observamos a distância entre o valor real e o predito para observar a margem de erro e aferir a assertividade. Dito isso derivamos algumas métricas a partir das quais tiraremos um veredito da efetividade do algoritmo ao nosso cenário como por exemplo o MAE (Erro absoluto médio que nada mais é do que a soma da diferença entre real e predito para todos os registros dividido pelo número de registros testados) e MSE (Anterior ao quadrado), há funções respectivas no pacote metrics do sklearn para essas métricas de erro

Regularização

* **Lasso**
  + Tende a tornar o coeficiente em zero absoluto
  + Elimina os atributos previsores irrelevantes
* **Ridge**
  + Nunca torna o coeficiente zero absoluto
* **Elastic net**
  + Também não gera coeficiente zero absoluto

Premissas para utilização da regressão linear

* Deve haver uma relação linear entre variáveis independentes e a dependente (Valor das features influenciam diretamente a variável alvo)
  + Isso pode ser visualizado por meio do gráfico pairplot do seaborn ou função corr do pandas
* Residuais médios devem ser iguais a 0
* Distribuição normal de termos errôneos
  + Pode ser visualizado pelo displot
* Não deve haver multicolinearidade
  + Ou seja, apesar das variáveis independentes afetarem a variável dependente, não deve haver uma correlação direta exata entre as variáveis independentes em si. Se temos por exemplo como features altura e peso para prever o IMC, existe multicolinearidade pois a tendência é que a altura impacte diretamente o peso

### Regressão logística

Visa prever uma variável dependente do tipo binária, ou seja, usado basicamente para responder questões de sim ou não ou então classificação com múltiplas classes. O algoritmo de regressão linear não é adequado para esses casos, pois, traça uma linha preditora que por natureza não contemplará adequadamente os registros de aprendizado visto que há apenas dois possíveis valores (O objetivo será determinar a qual das duas classes o registro pertence)

Os parâmetros para esse algoritmo são encontrados por meio da técnica de descida do gradiente. Se trata de uma técnica de otimização aplicável a vários algoritmos, visa encontrar a taxa de perda para cada parâmetro e atualizar cada um para tornar essa taxa tão mínima quanto possível

Eliminação recursiva de feature

Métricas de avaliação de modelo de classificação

Os modelos de classificação visam estabelecer o padrão entre os parâmetros de entrada e a classe de saída, seja em possibilidade binária ou múltiplas possíveis classes, a avaliação desse tipo de modelo consiste em observar as métricas que visam aferir a adequação do padrão estabelecido com a base de dados de teste e principalmente com os dados reais, deve haver uma boa capacidade de generalização, ou seja, o modelo não deve ser enviesado para funcionar somente com os registros com os quais aprendeu, mas sim, também se adaptar a novos registros reais, dentre as métricas disponíveis estão

* **Acurácia**
  + Número de previsões corretas dividido pelo número total de previsões resultará na porcentagem de acerto na classificação
* **Matriz de confusão**
  + Exibe a distribuição de classificações corretas e incorretas feitas pelo modelo em uma execução. Trabalha com os conceitos de falso/verdadeiro (Diz respeito a se o resultado da previsão foi correto ou incorreto) negativo/positivo (Diz respeito a se o valor previsto foi sim ou não)
* **Recall/TPR**
  + Define a porcentagem/quantidade de classificações corretas positivas dentre todas as previsões
* **Precisão**
* **F1-score**

### Árvores de decisão

Algoritmo estabelece uma estrutura de nós condicionais onde cada nó abre uma bifurcação ou classifica o registro com base nos valores dos atributos. É um algoritmo de aprendizagem supervisionada

Popular para sistemas baseados em regras

* **Nó**
  + Condição com base no valor do atributo
* **Ramificação**
  + Caminhos possíveis originários verdadeiro/falso de uma condição e que levam a uma nova condição ou nó folha
* **Nó folha**
  + Pontos mais profundos da árvore de onde são emitidas as classificações finais

Podem ser usadas tanto para classificação (Sim ou não) e regressão (Valor real contínuo)

* **Ganho de informação**
* **Entropia**
  + Medida de aleatoriedade no conjunto de dados
* **Coeficiente de Gini**
  + Medida de probabilidade de classificar incorretamente um registro com determinada classe
  + Encontrar o erro durante a classificação

Um problema comum com esse algoritmo é a probabilidade de overfit ou sobre ajuste aos dados de origem, uma vez que um excesso de nós de decisão ou bifurcações podem ser criados e nesse caso o modelo decora os dados de treinamento e não generaliza bem para novas ocorrências. Para esse cenário existem duas saídas

* Estabelecer um número mínimo de registros por folha
  + Ou seja, garantir que uma determinada regra gerada atende um número mínimo de dados, de forma que não seja tão específica a uma ou poucas ocorrências
* Poda ou pruning
  + Consiste em remover algumas das folhas ou regras geradas que não agreguem ao modelo

O sagemaker oferece uma implementação do algoritmo relacionado XGBoost

A árvore gerada pode não levar em consideração atributos irrelevantes para seu processo de decisão. O treinamento do modelo baseado em árvore consiste em encontrar a ordem, combinações e condições de atributos no conjunto de dados que formarão o método pelo qual novos registros serão classificados

Os melhores/mais relevantes atributos são detectados por meio das fórmulas de entropia e ganho de informação e são colocados nos ramos iniciais/acima da árvore. A entropia é um cálculo geral, que leva em consideração a distribuição de cada valor distinto da classe com relação ao total de registros avaliados (Exemplo: Entre 10 jogadores há 3 zagueiros, 3 atacantes, 1 goleiro e 3 meiocampistas), em seguida para cada atributo e para cada valor distinto destrincha a distribuição de ocorrências com relação a cada valor possível da classe (Exemplo: Para o atributo número da camisa e valor 1 há 1 ocorrência e esse único é goleiro, então, 1/1 para goleiros, 0/1 para atacantes e assim por diante), por fim com a entropia da classe e entropia de cada valor distinto do atributo avaliado se aplica a fórmula de ganho de informação cujo valor sinaliza a relevância do atributo para a classificação. Observasse que quanto menor a entropia específica de cada valor distinto de um atributo com relação a classe mais fácil é a classificação com base nele visto que não há grande distribuição (No exemplo do número de camisa igual a 1 todos são goleiros, então a correlação é maior)

O processo acima é executado para extrair o nó raiz e detectar atributos com pouco ganho de informação que podem ser excluídos (Conceito de poda), o algoritmo continua de maneira recursiva isolando cada valor do atributo nó raiz como bifurcação em uma nova base de dados e repetindo os cálculos até que todas as saídas sejam nós folha de decisão/classificação

A entropia conforme explicado acima é um dos critérios possíveis para uso de árvore de decisão, o outro é a impureza de Gini

Árvores muito grandes geralmente representam um sobre ajuste aos dados de treinamento (Pois gerou o número de bifurcações necessárias para contemplar todos os registros de treinamento e não irá generalizar bem para novos)

O algoritmo de árvore de decisão tem como vantagens sua fácil interpretação, visto que o modelo gerado visualmente é uma estrutura de condicionais aninhada e o fato de que dispensa normalização/padronização (Colocar por exemplo, os atributos de salário e média de pontuação de ataque do jogador em uma mesma escala numérica)

O termo CART se refere ao uso de árvores de decisão tanto para problemas de classificação quanto para regressão

### Overfitting

Modelo viciado nos dados de treinamento (Os decorou ao invés de aprender) e com pouca capacidade de generalização para novos registros. Pode ocorrer por várias razões como dados de treinamento com ruído e sub processados, modelo muito complexo

Geralmente endereçado por meio de validação cruzada KFold, consiste em dividir o conjunto de dados um número de partes conhecido como K e repetir o processo de treinamento variando as porções utilizadas para aprendizagem e validação

### Underfitting

Modelo não consegue abstrair os padrões a partir dos dados de treinamento. Pode ser solucionado por aumento no número de dimensões

### HPO

Método manual

* + Ajuste manual dos parâmetros do modelo com base em conhecimento intuitivo e abordagem de tentativa e erro observando os resultados iterativamente

GridSearchCV

* + Ajuste automatizado com base em força bruta e repetição exaustiva aplicando todas as possíveis combinações para os parâmetros do modelo com o intuito de encontrar o melhor resultado e sua configuração
  + Analogia: Tem três camisetas, duas calças e três pares de tênis para um encontro. Testará todas as combinações de roupa e se avaliará de zero até dez selecionando a melhor para o encontro ao final

Busca randômica

* + Escolhe aleatoriamente os valores a serem testados nos parâmetros e combinações ao invés de especificar manualmente

### k-NN

Algoritmo baseado no paradigma de aprendizagem baseada em instâncias, dado um novo registro calcula a distância lógica entre ele e os demais pontos de dados existentes (Cujo número de vizinhos avaliados é determinado pelo parâmetro k) e o classifica de acordo com o rótulo do vizinho mais próximo. Ao contrário de outros algoritmos o KNN não gera um artefato de modelo descartando os dados de treinamento, ou seja, para uma classificação ele sempre carregará os dados em memória para realizar seu processo de decisão (Nesse caso o cálculo de distância), por esse motivo ele é categorizado como algoritmo de aprendizagem tardia ou lazy. Uma vez calculada a distância de uma nova instância com todos os registros o rótulo dela será inferido com base na maioria relativa a quantidade de vizinhos selecionados por meio do parâmetro k

Pode ser usado em sistemas de recomendação (Exemplo: Dado um novo usuário de streaming que assistiu poucos filmes qual outro usuário com maior semelhança cujos conteúdos podemos sugerir ao primeiro)

Exemplo de caso de uso de classificação: Dado uma base de dados de características de filme (Grau de violência, romance, ação e comédia) e gênero (Terror e comédia) classificar um novo filme cujo gênero é desconhecido. Uma vez feito o cálculo de distância entre o novo filme e todos os demais caso o k seja 3 e esses 3 vizinhos mais próximos sejam: 2 filmes de terror e 1 de comédia, então o novo filme será classificado como terror

Utiliza internamente a fórmula de Distância Euclidiana (Permite utilizar outras), quanto menor esse valor menor a distância entre dois pontos de dados

Requer padronização/normalização (Fórmulas diferentes) de escala dos atributos numéricos (Pois aqui essa diferença poderá distorcer a noção de distância obtida pela fórmula, por exemplo com idade e renda anual, a tendência é que a importância da idade seja muito menor ou subestimada pelo algoritmo, mesmo que ela tenha uma forte correlação com a classe). Sem a equivalência de escala dos atributos numéricos os atributos de maior escala são arbitrariamente considerados mais importantes, o que tende a causar erros

**Encoding**: Transformação/Mapeamento de dados categóricos em numéricos para fins de otimização de performance de algoritmo

Com valor de k pequeno pode ocorrer influência de valores discrepantes, enquanto que com valor muito grandes pode ocorrer Overfitting por força da maioria. Mais lento devido a não manter estado (Modelo treinado gerado), o processamento é sempre repetido para cada requisição (Cálculo de distância feito para cada nova instância a partir de toda a base)

O parâmetro random\_state ao dividir a base entre treinamento e teste permite usar sempre as mesmas porções ao invés de realizar uma separação distinta de registros a cada execução, sua definição permite melhor comparabilidade de otimização de parâmetros e diferentes experimentos

### SVM

Visa assim como a regressão logística traçar uma reta de separação dos dados entre classes, aqui temos a especificidade de também buscar uma margem máxima entre os vetores de suporte (Pontos de dados mais próximos da linha divisória/hiperplano) e a divisória em si. Há uma envoltória convexa em cada lado da linha divisória entre as classes no gráfico contendo os registros pertencentes a ela

Deriva novos atributos e conceitos automaticamente a partir das características existentes

Útil para classificação e regressão

Busca o melhor hiperplano e a maior margem possível de separação entre duas classes (Distância entre o hiperplano e os pontos de dados mais próximos dele)

Pode ser do tipo Hard (Não permite pontos de dados mal classificados na região da margem) ou Soft (Permite)

Utiliza cálculos matemáticos muito mais complexos do que os algoritmos anteriores e por isso é de difícil interpretação para não especialistas, sendo um modelo de caixa-preta

O valor de C especifica a flexibilidade possível para a linha separatória entre as classes. Em tese quanto maior mais customizada e assertiva ela será, porém há um limite

**Problema linear**: Uma classificação onde é possível estabelecer uma reta de separação entre duas classes distintas

**Problema não linear**: Cenários onde não é possível traçar uma reta que isole uma classe de outra. O método ou truque de Kernel visa tornar problemas não lineares em lineares através de diferentes técnicas (Linear, Polynomial, Gaussian, etc.)

### K-means

Intuição teórica

Algoritmo para agrupamento ou Clustering, típicos casos de usos são perfis de clientes, grupos de produtos similares, agrupamento de documentos e notícias, etc. Pertencente a aprendizagem não supervisionada, detecta as relações entre os dados e gera grupos de similaridade

Cada grupo possui um ponto de dados central arbitrariamente inicializado, para cada registro calcula as distâncias para os centroides e o atribui para o grupo mais próximo. O valor de K indica o número de grupos ou centroides iniciais, o algoritmo é iterativo e a cada rodada ele reposiciona o centroide de acordo com a média realocando os elementos até encontrar a distribuição final. Utiliza Distância Euclidiana (Assim como KNN) para esses cálculos de distâncias que precedem as movimentações dos indivíduos ao grupo, essas distâncias se tornam humanamente visualizáveis com os grupos já criados, o algoritmo, entretanto, realiza os cálculos iterativamente com base em fórmula matemática nos bastidores

A inicialização e posicionamento dos centroides é relevante, visto que conforme onde eles são colocados inicialmente grupos diferentes serão gerados. Ao definir o número de clusters será preciso conhecer previamente o número esperado de grupos no problema (Exemplo: Se os grupos esperados são criança, jovens, adultos e idosos o número de clusters é 4). O valor do parâmetro WCSS/Inertia (Obtido pelo gráfico/método de cotovelo) determina a distribuição dos clusters e deve ser manipulado a fim de encontrar o melhor resultado (Caso o usuário desconheça o número necessário previamente)

O parâmetro Inertia também deve ser observado (Mede a coesão interna)

O melhor número de clusters pode ser visualmente avaliado por meio do gráfico de cotovelo (Mostra o SSE que é uma métrica de erro para cada quantidade de clusters), deve buscar uma faixa onde não haja uma quebra abrupta entre um número de clusters e outro. Caso não seja possível aferir claramente de maneira visual um método alternativo consiste em utilizar o Silhouette score

Requer padronização de escala dos atributos numéricos, pois, por ser um algoritmo de agrupamento utiliza distâncias e com escalas divergentes irá interpretar erroneamente a importância dos atributos

As premissas buscadas na formação dos clusters são coesão interna entre os pontos de dados em um mesmo grupo e separação externa entre os diferentes clusters. Os tipos de Clustering possíveis são

Uma vez executado o agrupamento geramos um gráfico com os clusters e extrairemos insights de negócio para embasamento de decisões. Para gerar um gráfico 2D precisamos de dois eixos, ou seja, um problema de agrupamento levando em conta apenas dois atributos, para visualização gráfica de agrupamento envolvendo múltiplos atributos podemos utilizar a técnica PCA (Reduz o número de dimensões combinando atributos)

* Exclusivo
  + Implementado pelo algoritmo kmeans, nesse tipo cada ponto de dados pertencerá a um único cluster
* Com sobreposição
* Hierárquico

### Agrupamento hierárquico

Tipo de agrupamento no qual o produto é um gráfico onde o eixo x é composto pelos pontos de dados e o eixo y a distância euclidiana, o algoritmo consiste em desenhar uma linha de intersecção entre os primeiros registros arbitrariamente escolhidos cuja medida é a distância entre eles, cada uma dessas linhas compõe um cluster e o processo segue recursivamente calculando a distância entre os pontos de dados e clusters formando a figura de um Dendrograma

Para construir o Dendrograma cada um dos pontos de dados é disposto em um eixo como seu próprio cluster, registros iniciais formam as primeiras combinações de cluster e isso é repetido recursivamente até que todos os clusters individuais e combinados formem um único cluster fechando a estrutura

Pode ser Aglomerativo (A partir dos pontos de dados individuais, ou seja, de baixo para cima faz a combinação e montagem dos clusters até todos sejam contemplados por um comum) ou Divisivo (De cima para baixo, a partir de um cluster raiz faz a derivação dos menores até os pontos de dados individuais), ao utilizar o algoritmo selecionamos a métrica de distância a ser considerada para montagem dos clusters. Observar o Dendrograma é o primeiro passo para uso do agrupamento hierárquico visto que fornece uma figura visual a partir da qual é possível aferir o número ideal de clusters

### DBSCAN

Baseado em densidade, agrupando os pontos similares no mesmo espaço. Não precisa especificar o número de clusters. Especifica um raio (Controlado pelo parâmetro eps) ao redor de um ponto de dados arbitrário a partir do qual um cluster será criado (Os demais pontos presentes nesse raio serão considerados vizinhos de um mesmo cluster). Outro parâmetro relevante é o mínimo de amostras por cluster que deve ser ajustado de acordo com o tamanho da base de dados

Tem como vantagens se encaixar a qualquer cenário de agrupamento, enquanto os demais tem limitações quanto a natureza dos agrupamentos e o fato de não ser sensível a valores discrepantes

O termo ruído se refere a aspectos nos dados que prejudicam a coesão em seu agrupamento (Sendo comum encontrar indivíduos de mesmo aspecto sobrepostos em grupos distintos). O DBSCAN funciona melhor para cenários com muita sobreposição entre grupos

### Análise de dados temporais

Ocorre quando pontos de dados estão ligados a períodos de tempo sequenciais, nesse caso o tempo atua como uma variável independente. Útil para compreender tendências (Aumento, redução ou estabilidade/horizontalidade), sazonalidade (Padrões como picos ou quedas bruscas em determinados períodos com caráter fixo), ciclos (Padrões de longa duração não atrelados e períodos específicos) e irregularidades (Comportamentos erráticos ou variações abruptas não esperadas)

Quando aos aspectos e limitações quanto a esse tipo de conjunto de dados temos

* Requer linearidade em termos de relacionamento entre os pontos de dados
* Modelos geralmente limitados a uma única variável (A segunda obrigatoriamente se trata dos períodos temporais)

Dados podem ser estacionários (Constante valor mediano, não apresenta tendências, sazonalidade, ciclos ou irregularidades) ou não. Existem métodos estatísticos para avaliar se as séries temporais são ou não estacionárias (Como por exemplo Dickey-Fuller) e diferentes métodos para transformar dados não estacionários em estacionários (Necessário como passo prévio para alguns modelos)

Um passo comum de pré-processamento para dados dessa natureza é a transformação da coluna contendo a data de textual ou numérico para o formato de data em si, caso seja necessário

O modelo ARIMA provê os melhores parâmetros para realizar previsões das próximas ocorrências em períodos futuros a partir das séries temporais originais (Que devem ser convertidas para formato estacionário)

## Implementação e operações de ML

A implantação de um modelo de classificação de imagem criado com TensorFlow pode ser feito por meio de uma função Lambda

### Docker

Permite encapsular aplicações junto a seu ambiente de execução em contêineres que são pacotes análogos a máquinas virtuais, porém não persistentes e portáveis entre diferentes sistemas operacionais. Um container é criado com base em uma imagem, que é sua especificação de criação e parametrização geralmente contendo o ambiente de execução base, por exemplo Python, a cópia do código-fonte da aplicação e seu Build, dentre outros passos de configuração necessários (Em um arquivo denominado Dockerfile)

A implantação habitual com Lambda geralmente consiste no upload do código fonte avulso da função após seleção do ambiente de execução (Na lista opta por Runtime predefinida com base nas linguagens suportadas) no formato esperado de requisição e resposta. É possível também uma maior customização, ao escrever um Dockerfile com uma imagem de execução personalizada e armazenar ela no ECR (Repositório privado de imagens Docker na AWS), pode parecer contraintuitivo no início visto que a proposta do Serverless é abstrair ao máximo a interação do desenvolvedor com aspectos de infraestrutura, porém essa opção de deploy com imagem de container é apenas uma maneira de termos maior controle e personalização do ambiente de execução onde a função será hospedada, não precisaremos gerenciar a operação como no ECS (Para essa opção a imagem deve estar no ECR)

Podemos em seguida utilizar o API Gateway para expor o endpoint de previsões baseado no modelo em uma URL pública. Método útil para quando o modelo precisa ser exposto como um Web service

Nessa abordagem, ao contrário de uma implantação local, temos o CloudWatch como centralizador de logs e painel para uma série de automações e ações relacionadas a monitoramento e alarmística

### Implantação com SageMaker

O serviço oferece as instâncias de notebooks que são um tipo de VM do EC2 previamente configurada com o Jupyter, ao provisionar criamos uma função no IAM com acesso aos serviços auxiliares que serão necessários como S3 e optamos pelo tipo de instância de acordo com o tipo de modelo que iremos treinar, bem como o volume de dados. Podemos também integrar de imediato nosso repositório do github, ao fim do provisionamento é exposta uma interface do Jupyter na qual podemos codificar nossos scripts de ML

O SageMaker oferece um SDK que permite acesso aos principais algoritmos e modelos por meio de contêineres, ao utilizar um KNN por exemplo, especificamos a localização da imagem no ECR, tipo de instância desejada e número de instâncias, etc. Ao disparar um job de treinamento usando o SDK do SageMaker e a partir de uma instância de notebooks poderemos monitorar através de uma aba específica no console o status e métricas, o que não é possível em um treinamento local. O artefato do modelo e outras saídas provenientes do treinamento são armazenados no S3. O SDK do SageMaker permite também já implantar o modelo diretamente por meio de um Endpoint visualizável através de uma aba no console que nos permitirá o escalonar e monitorar diretamente dentro do serviço

Nessa abordagem podemos ter uma função Lambda com acesso direto ao SageMaker que ao invés de expor ela própria o modelo, irá apenas invocar o Endpoint disponibilizado por meio do Boto3. Essa mesma função poderá também ser exposta com o API Gateway

### Jumpstart

Serviço que provê modelos previamente treinados para problemas comuns como classificação de imagens de animais, permite o uso da abordagem de aprendizagem por transferência

### Rastreabilidade ou Lineage

Por razões regulatórias, éticas e técnicas (Principalmente relacionadas a performance e acurácia) é preciso que aplicações e processos relacionados a ML sejam interpretáveis, rastreáveis e reproduzíveis, com relação ao desenvolvimento tradicional de software esses requisitos são mais criteriosos e as ferramentas e processos envolvidos mais complexos, pois além de versionamento do código, há necessidade também de controlar versões dos conjuntos de dados e experimentos (Diferentes pré-processamentos, parâmetros de modelos, diferentes algoritmos, etc.)

O SageMaker e a AWS em geral oferecem ferramentas para facilitar esse requisito de rastreabilidade, que para ML é intrinsecamente complexo tendo em vista a própria complexidade das etapas do ciclo de vida e o fato de que muitos algoritmos são de caixa preta

Feature Store

* + Visa armazenar diferentes versões de conjunto de dados de treinamento, de maneira a poder reaproveitar diferentes massas com pré-processamento e engenharia de Features distintas. Isso traz economia em termos de processamento e esforço de desenvolvimento

ECR

* + Armazena diferentes versões de imagem de contêiner nas quais os modelos são parametrizados para execução

CodeCommit

* + Repositório privado Git. No contexto de ML teremos o histórico de nosso código Python para pré-processamento, treinamento de modelos, HPO, exposição de endpoints, etc.

Experiments

* + Permite comparar e rastrear de ponta a ponta diferentes experimentos de ML para solução de um mesmo problema

Model Registry

* + Catálogo de modelos e versões

### Fraud Detector

Serviço dedicado a solução de casos de uso relacionados a detecção de fraudes utilizando ML

### Deep Learning

Subárea de ML (Tradicionalmente utiliza métodos estatísticos) concentrada em algoritmos inspirados no funcionamento do cérebro humano, utilizando o conceito de redes neurais artificiais

**AI**: Fazem com que computadores e aplicações possam emular capacidades humanas

**ML**: Modelos estatísticos que analisam e processam dados com base em técnicas matemáticas para gerar previsões ou extrair intuições/padrões

**DL**: Subconjunto de ML que utiliza técnicas e algoritmos que visam emular o funcionamento do cérebro humano no quesito de aprendizagem e tomada de decisão. Métodos simulam características de um neurônio biológico

Possui uma camada de entrada (Pode receber texto, dados tabulares, imagens, áudio, vídeo, etc.), neurônios (Camadas de processamento) e saída

As funções de ativação simulam as sinapses (Que no caso do cérebro humana decide se um neurônio é ou não ativado)

TensorFlow é uma biblioteca de código aberto para ML especializada em DL e redes neurais

Ao criar uma rede neural especificamos o número de camadas, função de ativação, função de perda, otimizador de modelo e métrica de avaliação. O objetivo permanece sendo encontrar a correlação entre eixo x e y

Os algoritmos tradicionais ou os de ML que vimos até aqui são direcionados ou tem uso mais comum com problemas específicos pré-determinados (Por exemplo sistemas de recomendação, ordenação, previsão de séries temporais, etc.). As redes neurais são aplicáveis para problemas mais complexos em que um único algoritmo não é o suficiente para chegar em uma resolução como é o caso de reconhecimento facial, processamento de linguagem natural, geração de arte e conteúdo, dentre outros

Uma analogia possível é a seguinte: No cotidiano a mente humana se depara com desafios como ir de casa até o trabalho, para isso, fazemos uso de uma série de conhecimentos e tomadas de decisão como colocar o alarme para despertar (Com base no início do expediente calcula o horário que precisa acordar levando em consideração tudo o que precisa fazer como preparação), escolhe uma roupa (Além de seguir o código de vestimenta do local outros fatores sociais são avaliados para causar boa impressão ou não), toma ou não café da manhã (Seus hábitos alimentares e fatores como metabolismo são entradas aqui) e dirige até o local (Precisa saber dirigir, bem como conhecer a rota ou utilizar um GPS). Ao abstrair uma tarefa aparentemente simples e algumas de suas etapas, enxergamos que grande parte de nossas ações rotineiras não podem ser destrinchadas em uma única métrica ou passos simples, pelo contrário, são uma série de algoritmos extremamente complexos uma enormidade de parâmetros e variações envolvendo ainda aspectos subjetivos e de difícil planificação, sendo assim, não há um único algoritmo ou fórmula que resolve um problema de maior panorama. Dessa mesma maneira redes neurais e DL são aplicáveis em cenário com um volume massivo de dados e problemas complexos não solucionáveis pelos algoritmos mais simples vistos até aqui

Redes neurais artificiais são uma abstração simplificada que visam emular o funcionamento do cérebro humano

O neurônio processa as informações de uma determinada entrada, os axônios e sinapses transmitem os sinais ou estímulos e a ativação de um neurônio depende do resultado de uma função de ativação, ao final é gerada uma saída (Resposta final ou input para a próxima camada de neurônios)

As entradas são os atributos ou variáveis independentes, cada entrada possui um determinado peso (Análogos as sinapses), o conjunto de entradas e respectivos pesos são submetidos a função de ativação

# Livro

## Capítulo 1

### Rekognition

Serviço gerenciado que utiliza IA para análise de vídeo e imagens. Útil para cenários/aplicações como detecção de objetos, classificação de imagens, detecção de texto em imagem, reconhecimento facial, de sentimentos e segurança pública. Seu uso não requer conhecimento sobre ML e dispensa a necessidade de gestão de infraestrutura para treinamento e implantação dos modelos, isso pois, oferece modelos previamente treinados para os cenários mais comuns anteriormente mencionados

Internamente utiliza os conceitos de CNN/ConvNet ou Rede neural convolucional e Transfer learning que é a readequação de um modelo treinado com um dado conjunto de dados para outro

Casos de uso

* **Classificação ou rotulação de imagens**
  + Consiste em identificar de quais objetos a imagem é formada ou encontrar objetos ou conceitos de interesse de maneira a derivar do que se trata a imagem (Por exemplo uma festa, uma paisagem, uma partida de algum esporte, etc.)
* **Classificação ou rotulação customizada de imagens**
  + O mesmo conceito do anterior se aplica, com a diferença de que os objetos e situações detectadas aqui são específicas e contextuais de uma determinada área de negócio ou nicho e não uma classificação geral de objetos comuns universais como árvores ou carros. Um exemplo seria uma fábrica que aplicou um processo para detectar peças defeituosas no final da linha de produção
* **Detecção e busca facial**
  + Consiste em não somente encontrar rostos desconhecidos em imagens como também procurar uma face específica proveniente de seu banco de dados em uma nova imagem
* **Movimentação**
  + Consiste em rastrear a movimentação ou caminho percorrido por uma pessoa ou objeto em um vídeo. Um exemplo seria gerar estatísticas sobre a movimentação de um jogador no campo de futebol
* **Detecção de texto**
  + Consiste em extrair e converter texto em imagens e obter o mesmo em formato cru e passível de leitura pelo computador que em seguida pode ser utilizado em operações ou processamentos posteriores

Trabalha tanto com imagens estáticas como também vídeos, porém alguns detalhes de funcionamento e operações disponíveis variam conforme o formato, para imagens o processamento é síncrono enquanto que para vídeos é assíncrono (Publica os resultados em tópico SNS após concluir e para obter é preciso posteriormente realizar uma chamada GET), a detecção de EPIs em trabalhadores por exemplo só funciona para imagens enquanto rastreamento de movimentação necessariamente se baseia em vídeos. A resposta consiste em um json com a coleção dos objetos encontrados, o bounding box que delimita a posição em que cada objeto se encontra na imagem e o grau de confiança da análise que podemos usar em ações posteriores como critério para o uso do resultado da análise em nossas aplicações

Enquanto que para imagens é possível o envio de um arquivo avulso, para lidar com vídeos eles necessariamente devem ser colocados no S3 e devemos prover um tópico SNS para receber o resultado de maneira assíncrona. O grau de confiança da análise depende dentre outros fatores da qualidade do vídeo ou imagem enviados para a API

Também é possível trabalhar com streaming de vídeos por meio da integração com o Kinesis

Principais capacidades: Detecção de rótulos (Objetos e respectivas posições em imagem), leitura facial (Se está sorrindo, gênero, faixa etárias, dentre outros), reconhecimento de celebridades, identificação de EPIS e reconhecimento de atividades em imagem e vídeo

### Amazon Textract

Utiliza OCR (Reconhecimento óptico de caracteres) para extrair texto processável e elementos de layout de documentos digitalizados (Como em PDFs, imagens, tabelas e formulários). Permite criar workflows automatizados de análise de documentos para substituir processos manuais. Não faz classificação de documentos, análise de sentimento ou reconhecimento de entidades no documento em si (Isso é escopo de outra ferramenta denominada Amazon Comprehend)

Casos de uso

* Extração de texto puro cru e processável de documentos digitalizados para processamento posterior como NLP
* Captura automática de dados de formulários como registros médicos, relatórios financeiros, documentos de impostos, dentre outros. Essa capacidade é amplamente útil para digitalização de valiosos dados disponíveis originalmente somente em formato físico ou escaneado
* Digitalização de diários, cadernos de estudo e outros tipos de documento escritos a mão
* Leitura de documentos como RG e CNH para extração de dados processáveis relevantes

O serviço é pago por documento analisado

Possui a API que detecta somente o texto e também a de análise que extrai os componentes estruturais de layout como linhas, formulários e tabelas. Sempre que possível apresenta uma resposta estruturada que permite maior facilidade e dinamismo na consulta e uso dos dados textuais recuperados como formato chave e valor para campos de formulário, células e linhas para tabelas, etc.

Também possui variação de funcionamento assíncrono e síncrono igual ao Rekognition, nesse caso PDFs de múltiplas linhas ou com formulários e documentos maiores utilizarão o SNS como sistema de mensageria para paralelizar o processamento e recuperação dos resultados. Permite armazenar os resultados no DynamoDB ou S3 dentre algumas das opções possíveis

Pode ser usado por exemplo para arquivar os dados de notas fiscais e recibos físicos cuja base em seguida pode alimentar um aplicativo de finanças pessoais

### Amazon Transcribe

Serviço de transcrição de voz/áudio para texto. Suporta streaming, áudio de entrada em diversos idiomas e inclusive arquivos com múltiplos idiomas em uma mesma gravação (Parametrizável via opção na chamada da API), enfileiramento nativo de jobs, filtro de palavras indesejadas e inclusão de vocabulário específico, remoção de informações sensíveis da transcrição (Detecção de PII), identificação de idioma e os diferentes locutores em uma conversa por exemplo

Para que a transcrição seja mais assertiva e contemple o reconhecimento de termos específicos de um nicho ou domínio é possível criar um modelo customizado incluindo os termos adicionais usando o Transcribe

O serviço possui uma versão específica para transcrição de áudios relacionados a medicina como gravações de atendimento por telemedicina, conversa entre médico e paciente, dentre outros cenários semelhantes, isso devido à alta especificidade do vocabulário utilizado em um hospital por exemplo. Essa versão contempla a redação de PHI (Dados pessoais de saúde de um indivíduo)

### Amazon Translate

Serviço de tradução de texto entre diferentes idiomas baseado em deep learning. Pago conforme o uso e assim com os serviços mencionados anteriormente possui modos de processamento síncrono e assíncrono em sua API a depender do volume dos documentos a serem traduzidos, integra ao S3 para busca dos documentos de input e também possui parametrização para terminologia customizada de termos que não devem ser traduzidos ao pé da letra, mas que são contextuais do vocabulário de nicho específico onde a tradução será aplicada

Não permite modelos customizados e utiliza por padrão os dados de cliente a fim de corrigir e melhorar os próprios algoritmos da AWS

Não traduz áudios como calls diretamente, para isso é possível montar um workflow integrando Transcribe e Translate

### Amazon Polly

Serviço de TTS, ou seja, conversão de texto para discurso em áudio, o oposto do que faz o Transcribe. Possui uma série de vozes predefinidas em diferentes linguagens e ao receber como input texto cru diretamente ou SSML (Uma linguagem de marcação específica para TTS que permite personalizar o discurso de saída em aspectos como entonação de voz, trechos de maior ênfase, velocidade de fala, etc. por meio de tags estilo XML)

O retorno pode conter não somente o áudio com o discurso de saída, mas também metadados sobre a leitura gerada por meio de Speech Marks que será um JSON com dados do discurso gerado como início e fim de um trecho, etc.

A saída pode ser armazenada em arquivos MP3, OGG ou PCM

A proposta é gerar discurso natural e fluído com base em texto

### Amazon Lex

Serviço que permite a criação de chatbots ou interfaces conversacionais sem necessidade de conhecimento em deep learning. Sua funcionalidade se baseia em ASR (Reconhecimento automático de discurso) e NLU (Compreensão de linguagem natural)

Trabalha com os conceitos

* **Bot**
  + É a interface conversacional com a qual o usuário lida para atingir um objetivo, no caso de um e-commerce por exemplo seria o fechamento de um pedido
* **Intent**
  + É uma ação possível previamente mapeada para o seu Bot que pode ser manifestada a partir da intenção do usuário. Em uma pizzaria por exemplo poderíamos ter a intenção OrderPizza. Uma intenção é manifestada ou acionada por meio de um enunciado e suas variações
  + São os propósitos possíveis dentro de sua aplicação, cada uma atendendo a uma intenção de usuário
* **Utterance**
  + É o enunciado ou Regex aceito para que o usuário acione determinada intenção. No exemplo da pizzaria um enunciado aceitável em inglês seria: "I want to order a pizza"
  + Com o Lex você provê alguns exemplos de enunciados para cada intenção e com isso já será possível generalizar para inúmeras variações dos mesmos enunciados sem que seja preciso mapear todos eles manualmente
  + No caso de não ser possível mapear um enunciado desconhecido com uma intenção possível existe a intenção de fallback que é uma ação default configurável no caso de múltiplas tentativas com input desconhecido, nela podemos passar um contato de suporte ou mensagem padrão ou até acionar atendimento humano
* **Slot**
  + São as variáveis ou parâmetros de uma intenção. No exemplo da OrderPizza os slots possíveis poderiam ser tamanho, sabor e quantidade de pizzas
* **Slot type**
  + Análogo aos tipos de variáveis do java, podemos usar os predefinidos pela Amazon ou criar tipos customizados para o nosso domínio como enumerações

As ações de backend podem ser acionadas por meio de funções Lambda, por exemplo se é pedido ao chatbot para que realize um agendamento de uma consulta ele irá disparar a função Lambda equivalente para realizar a ação e gravar no RDS por exemplo. Essa abordagem permite separar a estrutura ou fluxo de conversação (Respostas e intenções) da lógica de negócio e processamento associados (Por exemplo aplicação de descontos ou chamadas de APIs internas), evitando que o chatbot seja um monolito em que tudo isso está mesclado

O Lex se integra com plataformas como Facebook, Slack e Twilio

### Amazon Kendra

Permite estabelecer um mecanismo de busca customizado com base em fontes dispersas e específicas de seu domínio como documentos corporativos

Conceitos

* **Index**
  + São os metadados gerenciados sobre determinado documento que será fonte de informação e resposta para as buscas
* **Documentos**
  + São as variadas fontes e formatos de informação específica para o seu mecanismo de busca corporativo como arquivos pdf, páginas HTML de base de conhecimento, apresentações PowerPoint, documentos Word, etc.
* **Fontes de dados**
  + Com o Kendra não é preciso indexar manualmente os documentos que servirão com base para nosso mecanismo de busca. Basta plugar o serviço de origem de onde eles estão armazenados originalmente como um bucket do S3, banco de dados, Dropbox, Google Drive, Confluence e muitos outros

É possível otimizar os resultados de busca do Kendra provendo sinônimos corporativos ou termos do vocabulário interno como frases específicas

O Kendra não é uma NLP de propósito geral, mas sim focada em buscas em documentos e FAQ/Q&A

Pode ser integrada com o Lex para que o chatbot tenha como insumo fontes de informação interna da corporação para responder as intenções dos usuários

### Amazon Personalize

Serviço de ML para criação de sistemas de recomendação personalizados. Levando em conta dados de usuário (Demográficos), de item (Dados do produto a ser recomendado) e de interação (Eventos cruzando usuário e item como compras passadas e avaliações) possibilita a criação de soluções (Modelo baseado em uma de várias receitas para algoritmos e características de dados diferentes) e campanhas (Aplicação das recomendações)

Trabalha sempre com dados de usuário, de catálogo ou produtos e de interação entre ambos

### Amazon Forecast

Serviço que utiliza tanto algoritmos estatísticos clássicos quanto algoritmos tradicionais de deep learning combinados com métodos pesquisados e desenvolvidos internamente por cientistas da Amazon para desafios de previsão de séries temporais

Dedicado a casos de uso relacionados a previsão de séries temporais e dados históricos

Casos de uso

* Empresa de entrega que precisa prever demanda por navios em vários portos para planejar rotas de envio
* Empresa de cuidado médico que precisa mensurar demanda de suprimentos hospitalares em suas diversas unidades separadas geograficamente
* Varejista que precisa calcular demanda por produto para organizar seus estoques
* Provedor de cloud que precisa prever o uso de infraestrutura por seus clientes para planejar a capacidade de seus data centers de acordo

Podemos prover os dados e confiar no algoritmo escolhido pelo Amazon Forecast ou escolher dentre

* **ARIMA**
* **DeepAR+**
* **ETS**
* **Prophet**
* **CNN-QR**
* **NPTS**

A depender do cenário e especificidade do negócio e dados as métricas de avaliação a serem utilizadas para determinar a acurácia e porcentagem de erros do modelo obtido também variam. Algumas delas são

* **RMSE**
* **WAPE**
* **wQL**

### Amazon Comprehend

Serviço de análise de documentos para extração de insights do texto, os seguintes aspectos podem ser analisados

Classifica documento ou extrai entidades gerais ou específicas presentes no texto

* **Entidades**
  + Abstração para todo tipo de objeto ou conceito que pode estar incluso em um texto como datas, localização, organizações, pessoas, quantidades, títulos, eventos, produtos e outros
* **Frases ou termos chave**
* **PII**
  + Informações pessoas como endereço, nome, conta bancária, etc.
* **Idioma dominante no texto**
* **Sentimento ou viés emocional do texto**

Além dos modelos previamente treinados padrão é possível criar modelos customizados capazes de identificar entidades específicas. Existem três tipos de modelos customizados

Classificação customizada de documentos

* + Você provê um conjunto de documentos específicos associados a um rótulo e uma vez treinado o serviço será capaz de identificar o rótulo em novos documentos

Detecção de entidade customizada

* + Para que o serviço seja capaz de detectar entidades específicas de seu domínio como por exemplo posições de jogadores em um PDF sobre o jogo FIFA você provê a lista de entidades e um conjunto de documentos de aprendizado que as contem

Modelagem de tópicos em documentos

* + Consiste em treinar um modelo capaz de identificar assuntos ou tópicos específicos em seus documentos com base na associação de palavras comuns

A codificação de caracteres reconhecida por padrão é UTF-8 e há um limite de 25 documentos por requisição e até 5 KB por documento na requisição síncrona, acima desse contingente é preciso usar operação batch

Serviço possui variação nativa dedicada para documentos do domínio médico (Ou Healthcare)

### Amazon CodeGuru

Serviço de sugestões de melhoria de performance e qualidade de código com base em ML e que usa como base a base de código da Amazon. Consiste em dois serviços

Reviewer

* + Detecta código defeituoso e sugere melhorias para código Java ou Python. Não identifica erros de sintaxe. Recomenda melhorias relacionadas as melhores práticas da AWS no que tange ao uso de seu SDK, prevenção de vazamento de recursos, concorrência, vazamento de informação sensível, refatoração, validação de input, etc.
  + Focado em qualidade do código-fonte

Profiler

* + Avalia a performance de aplicações em tempo de execução e provê recomendações de ajuste fino na performance em questões como uso de CPU e memória. Provê um dashboard para visualização da performance de código executando em serviços de computação da AWS como o Lambda
  + Focado em performance das aplicações

### Amazon Augmented AI (A2I)

Serviço para criação de um workflow de revisão humana de previsões de baixa confiança geradas por modelo de ML ou auditoria aleatória de previsões por pessoas reais. Para isso selecionamos uma força de trabalho que pode ser proveniente do MTurk (Programa da Amazon que permite que trabalhadores humanos remotos façam renda realizando micro tarefas de validação) ou privada (Próprios funcionários do negócio ou terceirizados)

Por meio de parametrização JSON e das APIs do A2I acionamos um review humano

Integrado nativamente para revisão de saídas do Rekognition e Textract, mas podemos parametrizar para gerar feedback de previsões de nossos próprios modelos de ML (Criados usando o SageMaker)

### Amazon SageMaker

Serviço para criação, treinamento, refinamento e deploy de modelos de ML. Provê funcionalidades relevantes para cada passo em um típico ciclo de vida de projeto machine learning

Pode conectar também ao EFS e FSx como fontes de dados

Suporta regressão logística e linear por meio do algoritmo Linear Learner que suporta os formatos recordio e csv. Uma variação recomendada para dados com muitas dimensões e caráter disperso como fluxo de clicks é o algoritmo denominado Factorization Machines

Análise e pré-processamento

* + Consiste em preparar os dados para treinamento, o que geralmente significa tratar o conjunto de dados cru do modo em que foi obtida de maneira que se adeque aos modelos aos quais os dados serão submetidos. Isso tradicionalmente envolve o uso do Jupyter Notebook para criar as células de código em Python que farão e sanitização e transformações necessárias como remoção de colunas irrelevantes, remoção de valores nulos, etc.
  + Para essa etapa o SageMaker oferece
    - **Notebook Instances**
      * Instâncias computacionais com base no ec2 executando o Jupyter e contendo instalação padrão de frameworks de ML. Arquivos gravados em uma pasta específica predefinida do SageMaker são persistidos entre sessões. A instância pode ser associada diretamente a um projeto do github que é clonado na inicialização
      * Permite o uso de Docker no modo local para treinamento dos modelos, não funciona para os algoritmos prontos embutidos
    - **Studio**
      * É uma IDE baseada em web derivada do JupyterLab. Executa nos bastidores imagens de container para execução dos scripts. Provê interface visual para versionamento no git e muitas outras tarefas comuns ao desenvolver modelos de ML
    - **Data Wrangler**
      * Criação de pipelines de dados com passos de importação e transformação, bem como possibilita a análise dos conjuntos de dados
      * Integra ao Amazon Redshift
    - **Processing**
      * Para execução de cargas de trabalho relacionados a processamento de dados
      * Totalmente gerenciado e permite o uso tanto do sklearn quanto do PySpark para criação dos scripts e isso torna uma melhor opção com relação ao EMR para cientistas de dados sem especialidade em big data e em cenários com escala menor de dados
    - **Ground Truth**
      * Provê uma terceirização da etapa de rotulagem ou classificação de dados brutos que é uma etapa típica do pré-processamento por meio do Mechanical Turk (Mão de obra descentralizada especializada em tarefas de ML)
      * Provê uma UI com workflows nativos para tarefas de rotulação comuns como classificação de imagens, detecção de objetos por caixas delimitadoras, classificação de texto, reconhecimento de entidades, dentre outros
      * Permite criar workflows customizados caso os embutidos não atendam ao caso de uso específico
      * Resultados da revisão são publicados no S3
      * Exemplo de caso de uso seria empresa de carros autônomos submetendo vídeos das câmeras veiculares para reconhecimento manual de placas, pessoas, construções, animais, outros veículos e tudo o mais que for relevante para esse contexto
      * Possui capacidade de aprendizado ativo que permite rotulação automática e envio somente dos casos com baixa confiança para os revisores humanos, isso permite um escalonamento que resulta em uma massa de dados de entrada para nossos modelos muito maior

Permite também disparar o treinamento remoto e/ou distribuído, útil pois, nesse momento o conjunto de dados é carregado em memória, o que é inviável ou excessivamente custoso para se realizar local, ao fazer uso do EC2 podemos treinar modelos complexos em dados massivos enquanto continuamos a fazer exploração localmente

Treinamento

* + O SageMaker provê algoritmos nativos para casos de uso típicos como classificação e previsão de séries temporais e execução em ambiente computacional gerenciado de scripts de frameworks populares de ML
  + Outras funcionalidades de treinamento providas são
    - Treinamento distribuído
      * Paraleliza em nodos o processamento do treinamento por meio da divisão do dataset
    - Treinamento por instâncias pontuais
      * Utiliza o sistema de instâncias pontuais (EC2 Spot instances) ou de leilão do EC2 para reduzir o custo de processamento do treinamento
    - Refinamento automático de modelo
    - Monitoramento de Jobs de treinamento
    - Depurador

Inferência do modelo

* + Os modelos treinados no SageMaker geram artefatos comprimidos no S3
  + Pode gerar um endpoint customizado do tipo REST para gerar previsões com base no seu modelo, o número de instâncias que suporta essa API pode ser dinamicamente escalonado com base na carga de trabalho
    - Nos bastidores utiliza containers e o artefato no S3 com toda a parte de configuração como healthcheck abstraída para o desenvolvedor e usuário final
  + Pode também submeter diretamente um batch de dados fixos para processamento pelo modelo
    - Resultados armazenados de volta no S3

Um endpoint de modelo também possui capacidades de escalonamento automático assim como instâncias EC2 com opções de menor ou maior controle sobre os critérios e detalhes de escalonamento, para atender tanto usuários iniciantes ou casos de usos mais simples, quanto usuários mais avançados ou casos de uso mais refinados respectivamente

Estratégias de implantação

Dizem respeito a diferentes abordagens para atualização de um modelo operante através de um endpoint testável/invocável

* Recriação
  + Implica em tempo de inatividade visto que primeiro interrompe o endpoint antigo e em seguida recria o novo
* Blue/Green

Estratégias de teste

Dizem respeito ao controle de audiência/público e comparação de modelos durante o teste e validação

* Canário
  + É endereçada apenas a uma pequena porcentagem de clientes a nova versão de forma arbitrária, enquanto a maioria da audiência segue operando com a anterior. A migração ocorre gradualmente
* Sombra
  + Replica artificialmente requisições reais na versão antiga para nova de maneira que haja amostra suficiente para validação dela
* Teste A/B
  + Determina manualmente a bifurcação entre tráfego de produção e nova versão de acordo com critérios como tipo de dispositivo, localização, tipo de navegador, dentre outros

As estratégias de teste se diferem e precedem as estratégias de implantação, as complementando, visto que as primeiras são formas de validar o quão boa é uma nova versão, antes de as promover utilizando as últimas

### Dispositivos de ML da AWS

#### DeepLens

* + Câmera de vídeo programável com modelos previamente treinados e exemplos para aplicações de sistema de visão utilizando ML
  + Dispositivo com câmera integrado a serviços de ML da AWS para aplicações didáticas de casos de uso de visão computacional

#### DeepRacer

* + Carrinho de controle remoto para executar modelos treinados associados a movimentação (Aprendizagem por reforço)

#### DeepComposer

* + Geração de música com IA usando um teclado musical
  + Composição de música própria utilizando o teclado virtual e o console com apoio de modelos de ML

#### Panorama

* + Sistema para uso de câmeras IP associado a técnicas de ML
  + Visa adicionar capacidades de visão computacional a sistemas de câmeras/monitoramento existentes, reduzindo a necessidade de inspeção humana ou a otimizando

Nesse capítulo foram vistos os serviços que cobrem cada um dos subdomínios de aplicação de ML como visão, discurso, sistemas de recomendação e previsão

## Capítulo 2

### EFS

Serviço de sistemas de arquivo virtuais compartilhados do tipo NFS

No contexto de ML pode ser usado por exemplo por um grupo de cientistas de dados em um projeto para que compartilhem arquivos e dados sem que haja a necessidade de realizar o download e upload deles de volta ao S3 a todo momento

### FSx para Lustre

Sistema de arquivos altamente performático baseado em Lustre que é um popular FS para Linux

Permite espelhar objetos presentes no S3, facilitando o acesso e atualização ao eliminar a necessidade de sincronização manual

### Versionamento para ML

O Git e ferramentas análogas são para versionamento de código e portanto não recomendadas para versionamento de volumosos conjuntos de dados (Ou datasets) de aprendizado, treinamento ou teste. Para isso existe o DVC

### Amazon VPC

Permite criar redes lógicas virtuais dentro de sua conta AWS para melhor separação de recursos e segurança de acordo com o contexto organizacional de cada projeto

Os endpoints de VPC permitem conexão privada de recursos em uma VPC a serviços da AWS suportados sem precisar passar por um gateway de internet, o que proporciona maior segurança na comunicação. Este mecanismo é suportado e recomendado também em uma arquitetura de ML visto que muitos dos serviços já mencionados tem suporte

### Lambda

Serviço de computação serverless (Usuário não precisa explicitamente provisionar qualquer infraestrutura uma vez que isso fica encapsulado) e orientado a eventos. Permite escrever funções (Código geralmente sucinto e de responsabilidade única inserido em uma arquitetura maior) nas linguagens mais conhecidas do mercado. É amplamente usado dentro da AWS pois sua integração nativa a possibilidade de disparo com a grande maioria dos demais serviços permite acoplar nosso processamento backend específico dentro da arquitetura da solução de maneira bastante dinâmica e flexível

Possui limitações quanto a alocação de memória, tempo máximo de execução (O timeout), tamanho do payload de execução e pacote de código que devem ser levados em consideração ao considerar o seu uso (Idealmente seu uso é para processamentos pontuais avulsos de curta duração que podem ser implantados e geridos de maneira isolada, ou seja, não é indicado para construir hospedar aplicações complexas ou lidar com cargas de trabalho demasiadamente custosas em termos de processamento)

O acesso do código dentro da função via SDK para outros serviços AWS se dá por meio de funções de execução (Ou Execution Role), enquanto o acesso dos serviços a sua função Lambda se dá por meio de políticas de recurso (Ou Resource-based policies). Ambos são baseados em JSON diferindo apenas em que lado o controle de acesso é feito, sendo o primeiro da função para os serviços que a utilizarão e o segundo os serviços passíveis de disparar uma dada função por meio do código ARN (Identificador de recursos criados na Cloud)

### RoboMaker

Serviço voltado aos desenvolvedores focados na área de robótica

## Capítulo 3

O entendimento e alinhamento com negócio ou contexto da solução se faz necessário para que seja possível estabelecer objetivos claros para nossos modelos de ML, bem como mensurar sucesso ou fracasso, viabilidade de uso de algoritmos para o problema em questão, refinamento mais assertivo de todo o processo por meio de conhecimento especializado dos dados, dentre outras razões

O ciclo de um workflow para ML normalmente segue etapas e uma ordem lógica sequencial comum independente do problema ou contexto em questão (Obviamente a especificidade do domínio afeta detalhes ou remove/adiciona etapas no processo)

O CRISP-DM estabelece uma norma ou descrição conceitual de um workflow típico de ML em seus passos comuns, sendo agnóstico a tipos de negócio ou aplicações específicas

1. **Identificação do problema** de negócio com os stakeholders e discussão de **viabilidade do uso de ML**
2. Uma vez que a resposta é sim para a viabilidade do uso de ML para o contexto o próximo passo é o **enquadramento do problema específico** a ser sanado por **ML**
3. A **obtenção e avaliação da qualidade dos dados** a serem utilizados deve ser feita
4. O pré-processamento e preparação dos dados para exploração e visualização ocorre (**Análise exploratória**)
5. Isolamento, estudo, inserção e transformação de atributos relevantes para a previsão ou problema em questão (**Feature engineering**)
6. **Treinar um modelo de ML** adequado para o problema em questão
7. **Avaliação do desempenho do modelo** para aferir se atende aos objetivos e **refinamento de parâmetros** para se extrair a maior assertividade possível naquele contexto
8. **Implantar o modelo em produção** e obter previsões ou extrair valor/inteligência da automação proporcionada por ele
9. **Reiterar o processo** com as lições aprendidas caso os objetivos não sejam alcançados

Nem todo desafio de negócio é adequado ou passível do uso de ML e por isso discussões com as partes interessadas e especialistas operacionais e de regra de negócio se faz necessário para aferir a viabilidade do uso dessa abordagem e em caso positivo enquadrar o problema de maneira assertiva e direcionada a fim de obter o melhor resultado e evitar confusões, desperdícios e retrabalhos. Como projetos de ML podem ser extensos, complexos e custosos cabe sempre iniciar por estudos e POCs conceituais e em seguida fazer o trabalho de convencimento para obter o patrocínio necessário para tal empreitada. Cabe sempre se perguntar se soluções mais simples e tradicionais de software ou processo já não seriam o bastante para aquele contexto antes de embarcar nas etapas posteriores

Questões úteis para avaliar a viabilidade de uso de ML ou aplicar qualquer solução de software em geral

* Como o problema é resolvido hoje
* Como o sucesso é medido e quais são os objetivos, em natureza e número. De maneira clássica serão reduzir trabalho manual, reduzir custos, aumentar receita e alcançar novos clientes
* Qual o formato e características dos dados com os quais irá se trabalhar
* Temos os recursos técnicos para endereçar uma solução técnica de determinado tipo
* Se há noção e entendimento sobre os possíveis custos e investimentos necessários a serem feitos ao optar por cada solução

É útil em uma organização sem familiaridade com o uso de ML conquistar pequenas vitórias como apresentar uma PoC funcional ou MVP para determinado problema menor antes de estabelecer o comprometimento com um projeto maior e mais complexo

## Capítulo 4

Para entender se um problema de negócio é passível de uso de ML é útil relembrar e se ater aos conceitos básicos de ML

O objetivo de ML é prever ou classificar valores textuais ou quantitativos de um atributo (Dependente) em registros em que tal atributo é desconhecido utilizando abordagens matemáticas e/ou lógicas, para isso é preciso que tenhamos atributos conhecidos ou previsores (Independentes) a partir dos quais possamos estabelecer uma lógica de correlação e derivação do atributo alvo (Objetivo do modelo). Precisamos também de um conjunto histórico de dados ou registros para os quais sabemos o valor do atributo alvo (Base de aprendizado)

Um modelo assertivo consegue extrair os padrões e correlações da base de aprendizado e atende não somente a uma base de treinamento, mas também obtém bons resultados com registros desconhecidos e diferentes por sua capacidade de generalização. Um modelo pobre pode apenas decorar sua base de aprendizado e funcionar para registros idênticos, porém não responderá bem com novas ocorrências

É possível também usar ML quando não temos uma variável alvo em específico e queremos apenas extrair padrões e insights sobre os dados, esse é o caso justamente do aprendizado não supervisionado (Tem esse nome pois não temos conhecimento prévio ou exemplos nos quais se basear referente ao conhecimento que estamos tentando extrair do modelo) no qual agrupamentos podem ser uma das possíveis aplicações e saídas

O enquadramento de um problema como passível de ML do tipo aprendizado supervisionado passa por observar se em nosso contexto podemos observar claramente as características e entidades mencionadas acima

* Base histórica no qual é possível derivar conhecimento
* Registros minimamente homogêneos com variáveis independente e uma variável dependente

Etapas de enquadramento

1. Como o negócio ou interessado afere o sucesso do projeto?
   1. Será preciso um esforço de comunicação e alinhamento para mapear objetivos de alto nível como maximizar receita ou reduzir perdas em métricas matemáticas contra as quais a eficácia de um modelo de ML é aferida como porcentagem de acurácia ou precisão
2. Quais são os inputs e saídas esperadas?
   1. É preciso levantar se há conjuntos de dados específicos e mensuráveis com os quais possamos entrar como parâmetro em nosso processo de ML e caso não haja de imediato verificar se há a viabilidade de rotular/categorizar de imediato uma fonte de dados que possamos utilizar como ponto de entrada
   2. Também é preciso definir o escopo esperado de saída do projeto, pois ele estará diretamente correlacionado com o dos modelos gerados
3. Quais são as fontes de dados e se há suficientes registros categorizados?
   1. Aqui levantamos todas as fontes de dados internas, bem como a facilidade ou desafios de as acessar, consideramos também eventuais fontes externas como datasets públicos, notícias, APIs relevantes que possam ser chamadas e páginas nas quais podemos fazer Web Scraping, bem como as limitações burocráticas e éticas de as obter
   2. Aqui avaliamos também se há quantidade suficiente de registros categorizados para se gerar um modelo útil, por exemplo se há pouquíssimas aplicações de empréstimo fraudulentas será difícil criar um sistema de detecção de fraude útil usando ML supervisionada. No caso de ausência devemos avaliar o custo e viabilidade de se obter uma força de trabalho humana para rotular dados que possamos ingerir no processo de ML (O serviço SageMaker Ground Truth é dedicado a esse propósito)
   3. Na ausência e inviabilidade de se obter dados rotulados em número suficiente podemos nos perguntar se há alguma outra métrica ou conhecimento de negócio especializado que possa nos auxiliar a encontrar uma label alternativa
4. Comece simples e construa complexidade com o tempo e conforme necessidade
   1. Modelos simples como árvores de decisão são populares, pois, oferecem capacidade de interpretação (É possível com um teste de mesa e sessão de depuração humana rastrear a origem de uma previsão). Devemos tentar alcançar valor com métodos e soluções mais simples e conforme necessidade ou evolução adicionar robustez por meio de algoritmos mais complexos, iniciar com eles pode acabar sendo um desperdício e gerar uma experiência traumática com uso de ML tanto para o time de desenvolvimento quanto para a liderança
5. Considerar os riscos de cada modelo de acordo com as regras específicas de contexto como regulamentações industriais

## Capítulo 5

### Tipos de dados

* Estruturados
  + Conjuntos de dados nos quais se observa um padrão claro e homogêneo de formato com relação a posição e tipos de registros e atributos, um exemplo são os dados tabulares
* Não estruturados
  + Dados sem um padrão estrutural legível ou bem definido como logs, imagens, vídeos e arquivos de texto puro
* Semi estruturados
  + Conjuntos de dados heterogêneos que podem se apresentar nos dois formatos acima ou mesmo em vários formatos diferentes de dados estruturados como uma mescla de JSON e XML

### Conceitos de dados específicos para ML

* Dados categorizados ou rotulados
  + Consistem em registros que possuem um ou mais atributos dependentes ou alvos
* Dados não categorizados
  + Não possui atributo alvo ou rótulo
* Característica ou Feature
  + São as colunas ou atributos independentes ou previsores, ou seja aqueles que serão a base da previsão, ou todos os atributos com exceção do rótulo alvo
* Registro ou Data Point
  + Um registro consistindo de atributos previsores ou características e no caso de dados categorizados também de uma coluna com valor para o atributo alvo
  + Uma linha
* Dataset
  + Conjunto de dados ou linhas utilizadas no treinamento e validação do modelo
* Atributos numéricos
* Atributos categóricos ou textuais
  + Não há ordenação e são valores finitos de uma numeração, mesmo que compostos somente de números como é o caso de códigos Zip
* Imagens
  + Aparecem comumente por meio de arquivos com extensão JPEG ou PNG, dentre outros
* Áudios
  + Arquivos de conversas, música, gravações, transcrições e outras entidades de natureza auditiva. Aparece comumente por meio dos formatos MP3 e WAV
* Texto
  + Também conhecido como corpus são os mais variados tipos de arquivo de texto como PDF, TXT, JSON, CSV, dentre tantos outros
* Séries temporais
  + Consiste em conjuntos de dados que representam ou mostram a variação do valor de algo ao longo do tempo ou pontos de medição
* Dados de treino
* Dados de validação
* Dados de teste
  + Devem ser o mais próximo possível do que é esperado para o modelo receber em produção

### Repositórios de dados

Dados tabulares são comumente usados em aplicações OLTP (Como por exemplo transações bancárias online, compras, entrada de pedidos ou envio de mensagens de texto). São contempladas principalmente pelo serviço RDS

Para dados semiestruturados DynamoDB com seu formato chave e valor se apresenta como uma boa opção, já para armazenamento de arquivos não estruturados como imagens e documentos em texto puro temos o S3

A maioria das organizações ou contextos irão apresentar dados de todos os tipos e a Amazon provê e recomenda opções de serviços adequados a cada formato e tipo de aplicação

O Lake Formation permite criar um catálogo centralizado de dados de diversos tipos e origens armazenados em diferentes tipos de serviço da AWS e aplicar uma governança de controle de acesso unificada para todos os dados da organização independente de tipo ou o serviço em que estão localizados

### Migração de dados para AWS

Para uso dos serviços de ML é preciso que os dados estejam no S3, para migração há dois tipos de abordagem

* Batch
  + Serviços que fazem a transferência de dados em repouso como bancos de dados e arquivos armazenados
* Stream
  + Para dados em tempo real como sensores IoT, vídeo ao vivo, logs, feed de notícias e outras fontes contínuas de informação

### Data Pipeline

Serviço que permite migrar dados entre serviços da AWS facilitando nativamente a conversão de formato, acesso e outros detalhes que tornariam esse processo manualmente exaustivo

No contexto de ML é comumente utilizado para trazer conjuntos de dados de serviços como o DynamoDB e RDS para dentro do S3

### AWS DMS

Para migração e replicação entre diferentes bases de dados relacionais homogêneas ou heterogêneas, tanto on-premise quanto já dentro da AWS

Também pode entregar de base relacional para o S3

Possui o SCT para conversão de Schema entre bases de provedores diferentes como em uma migração entre Oracle e MySQL

### Kinesis

Família de serviços dedicados a etapas de manipulação de dados em streaming

Data Streams

Usado para coletar e processar grandes volumes de dados em fluxo contínuo ou tempo real (Streams)

Um stream consiste em um conjunto de registros gerados por eventos em tempo real como cliques em uma página web por exemplo. Os registros são recebidos e organizados em shards que podem ser pensadas como segmentações ou funis pelas quais as Streams passam após serem produzidas

O período de retenção corresponde ao tempo ocioso em que um fluxo de dados é recebido e não consumido permanecendo em repouso no funil em que foi recebido. Existe um valor padrão e é parametrizável até certo limite

Um produtor cria registros e Streams como é o caso de um servidor de logs. Podemos usar o KPL que é a biblioteca para produzir fluxos de dados contínuos em tempo real em nossas aplicações para consumo no Kinesis

Consumidores recebem os fluxos de dados provenientes das Streams e os processam. Podemos fazer uso do KCL para preparar nossas aplicações consumidoras de maneira mais abstraída e simplificada

Em resumo é uma mensageria focada para dados gerados em tempo real

#### Data Firehose

Permite fazer o repasse de Streams do Kinesis para armazenamento em outros serviços nativos ou de terceiros como o S3 e Redshift dispensando a necessidade de uma lógica de produção e consumo customizada

Provê algumas conversões utilitárias nativas como JSON para Parquet dos dados recebidos antes de repassar para o armazenamento de destino. Pode também chamar um Lambda customizado para transformação do fluxo de dados cuja saída será entregue ao serviço de destino

Data Analytics

Permite executar consultas SQL diretamente nos dados de streaming recebidos

Útil para monitoramento e disparo de notificações e alarmes

#### Video Streams

Facilita o streaming de vídeos para dentro da AWS com base em fontes como webcams, câmeras de segurança ou embutidas em carros ou babá eletrônica por exemplo, bem como drones, satélites e outras possibilidades

Pode ter como consumidor por exemplo uma instância EC2 contendo uma aplicação que utiliza o Rekognition para fazer a análise do conteúdo de vídeo recebido em tempo real

O MSK permite a migração de aplicações baseadas em Kafka

## Capítulo 6

### Ferramentas de preparação de dados

Ter os dados crus em armazenados no S3 não é o suficiente para aplicar um modelo de ML. A preparação ou pré-processamento dos dados consiste em atividades como

* Junção de diferentes fontes de dados em uma única como por exemplo realizar o Join via SQL da tabela de clientes com a tabela de pedidos
* Conversão de formato de arquivo e tipos de dados
* Remoção de valores nulos
* Escalonamento de dados em diferentes dimensões
* Aplicação de labels ou categorização

### Redshift Spectrum

Outro serviço analítico para consultas interativas diretamente em dados no S3. Porém necessita e recomendado para cenários quando já há um cluster do Redshift em operação

## Capítulo 7

Uma vez que os dados são coletados e passam por pré-processamento é preciso realizar a chamada engenharia de Feature que consiste em tornar os atributos crus, no caso de dados tabulares, em colunas já refinadas para o formato de valor que melhor funciona para cada tipo de modelo. Isso consiste geralmente em converter dados textuais categóricos em valores numéricos, remover outliers (Registros com valores atípicos)

### Técnicas para dados tabulares

* Remover linhas/registros duplicados
* Normalização
  + Coloca os valores numéricos de diferentes colunas em uma mesma escala de geralmente entre 0 e 1, isso se faz necessário devido ao funcionamento interno de alguns algoritmos. Por exemplo em um conjunto de dados relacionados a preços de imóveis podemos ter uma coluna de m2 do terreno/construído e outra para o número de quartos, por trabalharem em faixas numéricas muito distintas a normalização se faz necessária
* Padronização
  + Outro tipo de escalonamento de valores numéricos
* Detecção de valores atípicos
  + Quando há poucos exemplos de outliers não é recomendado os reter pois reduzem a capacidade de generalização do modelo
  + As estratégias para lidar são
    - Remoção pura e simples
    - Transformação Logarítmica
* Codificação de colunas categóricas
  + Codificação de rótulos
    - Consiste em atribuir e mapear um equivalente numérico para cada valor distinto de uma enumeração textual. Por exemplo para cada nome de dia da semana atribuir um número de 1 até 7, a possível desvantagem é caso o algoritmo derive ou interprete alguma ordem de grandeza entre os valores numéricos arbitrários atribuídos
  + One-hot-encoding
    - Nessa abordagem cada valor distinto possível de uma coluna textual se transforma em uma coluna adicional de flag booleana no dataset onde 1 representa valor presente e 0 ausente. O problema possível é o número massivos de colunas adicionais impactando performance no caso de atributos com muitos valores distintos como CEP
  + Codificação binária
    - Para coluna com dois possíveis valores distintos com sim ou não, atribuí um número para cada opção
* Lidar com valores ausentes
  + Conjuntos de dados onde há registros com valores pendentes para um ou mais atributos. Estratégias são
    - Coletar manualmente os dados pendentes para cada indivíduo, o que pode ser custoso ou inviável
    - Eliminar coluna caso seu valor esteja pendente na maioria ou quantidade significativa dos registros, isso também vai depender da relevância dela para o problema em questão
    - Imputação arbitrária utilizando média, valor default ou ocorrência mais frequente caso hajam poucos ausentes e isso não vá inserir viés aos dados
    - Aplicar um modelo de ML intermediário anterior para prever o valor da coluna ausente para os registros em questão, elimina o viés, mas pode ser custoso
* Equilibrar a amostragem no caso de desbalanceamento
  + Em detecção de fraudes por exemplo geralmente há muito mais exemplos de ocorrências não fraudulentas do que fraudulentas. Para equilibrar o insumo de aprendizado do modelo há as seguintes estratégias
    - Coletar mais dados e exemplos dos registros minoritários, que assim como a pesquisa e preenchimento manual de colunas ausentes pode ser custoso e inviável
    - Downsampling
      * Reduz arbitrariamente o número de registros majoritários
    - Upsampling
      * Replica o número de registros minoritários aumentando sua expressidade no dataset
    - SMOTE
      * Técnica que usa aprendizagem não supervisionada para gerar exemplos de casos minoritários
    - Aplicação de pesos ou ponderação
* Divisão de dados
  + Fragmentação dos dados em base de treinamento, validação e teste
  + Deve ocorrer previamente aos processos anteriores a fim de evitar vazamento de dados

### Técnicas para dados desestruturados ou de séries temporais

* Aumento de dados
  + Para imagens consiste em introduzir uma maior variedade de exemplos de uma mesma imagem por meio de cópias com edições arbitrárias como aplicação de rotação, zoom, alteração de cores, dentre outras técnicas. A biblioteca Keras provê utilidade nesse quesito
* Divisão por janelas de período em séries temporais
  + Para séries temporais o teste consiste em na técnica de backtesting, onde a validação ocorre com base em uma amostra de período de exemplo onde temos certeza sobre os valores. Por exemplo para preços de ações será usado um mês anterior ou passado
  + A divisão entre dados de treinamento e teste é diferente para séries temporais, pois um período é relacionado a outro e portanto uma divisão arbitrária por porcentagem não é possível
* Aplicação de pré-processamento específico para NLP

### Ferramentas de engenharia de feature na AWS

Para aplicar as técnicas acima mencionadas é possível usar o Glue e EMR com código customizado ou DataBrew e Data Wrangler como ferramentas low code com transformações embutidas predefinidas e visualização gráfica

## Capítulo 8

KNN

Constrói um índice com as distâncias entre os registros (Pontos de dados), para uma nova ocorrência calcula o vizinho mais próximo para derivar o rótulo no caso de classificação

Para o seu uso recomendasse reduzir a dimensionalidade dos dados (Número de colunas ou atributos), isto pois, o tamanho do modelo em memória corresponde ao volume dos dados em que ele foi treinado

SVM

Dado um conjunto de registros cria uma faixa limitadora ou fronteiras entre as classes presentes

Dados textuais

Podem ser documentos em texto puro, postagens em redes sociais, conversas em ferramentas de chat, dentre outros. São atendidos por modelos que visam dentre outros propósitos classificação de documentos, análise de sentimento e melhoria de ferramentas de atendimento autônomas

O SageMaker oferece o BlazingText que é uma otimização do algoritmo Word2vec que é voltado para algumas tarefas de NLP mencionadas no paragrafo acima. Diferente do Comprehend que é totalmente gerenciado, pode gerar modelos implantáveis fora da aws, o que traz maior flexibilidade

Análise de imagens

O SageMaker oferece um algoritmo de aprendizagem supervisionada baseado em ResNet para classificação de imagens

Treinar um modelo do 0 pode ser interessante para casos de uso com contexto extremamente específico e com muitos dados. Já quando há poucos dados, o negócio está buscando por insights rápidos ou o caso de uso não tem grandes especificidades é recomendado o uso de aprendizagem por transferência que consiste no reuso de um modelo treinado para uma tarefa em outra semelhante

Aprendizagem não supervisionada

PCA

Usado para redução de dimensionalidade como passo na engenharia de características e também como modelo final para problemas de aprendizagem não supervisionada

Implementação embutida também oferecida pelo SageMaker

K-means

Algoritmo de aprendizagem não supervisionada utilizado para problemas de agrupamento

RCF

Variação de árvores de decisão, porém em abordagem não supervisionada e dedicado a problemas de detecção de anomalias

LDA (Mais simples) e NTM (Mais performático)

Usados para sumarização e extração de tópicos ou assuntos a partir de texto puro como documentos

Aprendizagem por reforço

Visa simular um problema ou comportamento em um ambiente. Há a figura de um agente que performa um conjunto predefinido de ações e com base em penalizações e recompensas que refinam gradualmente sua política de comportamento ou ação

Cenários comuns

* Simular comportamentos no mercado financeiro
* Entregas autônomas e organização e gestão de galpão logístico
* Controle da CPU em jogos digitais

Em tempo de desenvolvimento geralmente os scripts de ML são testados por meio de células avulsas de Notebooks. É recomendado com o avanço do processo migrar para código Python passível de ser colocado em sistemas de controle de versão como o Git

Monitoramento

Muitas métricas e logs relacionados aos serviços de IA mencionados até aqui integram nativamente ao CloudWatch, o que nos permite monitoramento ativo ou passivo por meio de alarmes

As ações a nível de usuário com relação aos serviços são também gravadas no CloudTrail

Podemos também usar o EventBridge para reagir a eventos relacionados aos nossos processos dentro dos serviços de ML como por exemplo ao finalizar um job de treinamento ou completar uma revisão humana de previsão no A2I

O SageMaker oferece capacidades de depuração para os Jobs de treinamento

HPO

Otimização de hiper parâmetros consiste em executar vários Jobs de treinamento com parametrizações distintas no intuito de encontrar o melhor modelo e métricas para um dado problema

## Capítulo 9

Gestão de experimentação

Parte intrínseca do processo de ML é a experimentação que consiste para um mesmo problema aplicar variações com diferentes abordagens de pré-processamento, amostras de dados, algoritmos e parâmetros e o uso de métricas de avaliação e técnicas de teste distintas. Dado isso, todos os serviços discutidos até então possuem funcionalidades nativas de organização para acompanhamento de diferentes experimentações com chamadas de API do tipo list e describe para obter detalhes sobre cada projeto, o mesmo também é possível no console diretamente

Métricas e visualização

* TP ou verdadeiro positivo
* TN ou verdadeiro negativo
* FP ou falso positivo
* FN ou falso negativo

As métricas acima dizem respeito a um problema de classificação binária, onde verdadeiro são as classes corretamente identificadas como tal pelo modelo e falso são os erros, ou seja, classes com determinado rótulo que foram interpretadas erroneamente como o outro. Positivo e negativo são contextuais ao problema, no que tange ao tipo de classe com que se está lidando, um exemplo clássico seria previsão de diagnóstico de uma doença onde a classe sim seria o positivo e o não seria o negativo, nesse caso então, um verdadeiro positivo seria um paciente doente corretamente classificado como tal e um falso negativo seria um paciente doente erroneamente identificado como saudável

* Matriz de confusão
  + Uma tabela que mostra a distribuição de classificações para cada classe. Cada linha representa uma das classes presentes distintas e cada coluna representa a quantidade de vezes em que a dada classe foi interpretada como cada uma das classes possíveis. Em um modelo assertivo se espera que a maioria das ocorrências para cada linha esteja na intersecção da classe com a coluna dela própria (Por exemplo em 80% dos casos um goleiro foi classificado como goleiro e no restante foi classificado erroneamente dentre outras posições de linha)
* Precisão
  + Porcentagem de previsões positivas corretas no total
* Recall
  + Porcentagem de previsões positivas corretas com relação ao total de exemplos positivos disponíveis
* Taxa de falsos positivos
  + Porcentagem de falsas previsões positivas
* Pontuação F1
  + Média entre precisão e Recall
* Curva ROC
* Curva AUC
  + Serve para comparar modelos e entender qual o melhor para determinado caso de uso
* Curva de precisão e recall
* MAE
* MSE
* RMSE

O sobre ajuste ocorre quando o modelo performa bem com os dados de treinamento ou os decora, porém não generaliza bem para dados não vistos como o conjunto de validação

No SageMaker existirá um dashboard com métricas relevantes para cada um dos tipos de algoritmo embutidos

## Capítulo 10

Uma vez que temos um modelo de ML treinado pronto para validar o próximo passo é ou expor uma API para possibilitar previsões avulsas ou submeter um batch de registros específicos para inferência

Vários dos serviços de ML mencionados provem funcionalidade e modelos embutidos para cenários comuns e abstratos, mas também permitem criação de modelos customizados com base em insumo de dados específico ou contextual do problema. Por serem serviços gerenciados a hospedagem de modelos próprios oferece pouca flexibilidade com relação a customização de detalhes de infraestrutura e segurança, caso isso seja necessário podemos usar ou o sagemaker ou implantar diretamente nos serviços de computação de propósito geral como EC2 ou Fargate

Cada serviço possui suas nomenclaturas com relação ao artefato de modelo gerado, no caso do Forecast temos preditores e no caso do Personalize temos soluções/campanhas por exemplo

## Capítulo 11

Um modelo construído tem pouca utilidade se não for integrado de fato com um sistema e ter suas previsões utilizadas

* Integração com sistemas on-premise

Pode ser necessário devido a vários motivos como sistemas críticos legados de organizações que operem em servidores locais, setores críticos podem ter resistência ou limitação regulatória que impeçam migração completa para nuvem, ausência de conhecimento ou viabilidade para migração para cloud por diversos motivos ou sistemas específicos com necessidade de baixíssima latência e sem acesso à internet como em chão de fábrica

Para tal cenário de sistemas híbridos a AWS fornece várias ferramentas

* + Outposts
    - Permite executar serviços da AWS como EC2 e S3 com base em servidores locais
  + Wavelength
    - Suporta redes 5G
  + Local Zones
    - Implanta infraestrutura para acesso de baixíssima latência em zonas industriais
  + Snowball
    - Para transferência ou migração de dados para cloud ou execução de computação na ponta
  + RDS, ECS e outros serviços permitem executar tais serviços com infraestrutura local por meio de virtualização
  + Direct Connect
    - Permite a conexão e comunicação de uma rede local com sua VPC na AWS de forma privada
  + Serviços específicos para transferência e backup de dados locais na nuvem
  + Serviços de IoT que permitem execução na ponta
  + SageMaker Edge Manager
    - Provê dashboard para visualização de desempenho dos modelos nos dispositivos na ponta

Nesse panorama híbrido são possíveis várias mesclas e variações no que tange a divisão entre nuvem e local das etapas de treinamento, implantação e monitoramento

* Integração com sistemas em cloud

Nesse caso a AWS fornece serviços e ferramentas que dão suporte a todas as etapas do ciclo de vida de ML e que possuem forte integração em si, o que facilita todas as etapas

* Integração com Front-end

Para aplicações do tipo Full Stack (Com interface web ou aplicativo móvel) que utilizam modelos ML como insumo no backend existem as seguintes ferramentas

* + Amplify
  + Device Farm
  + Pinpoint
  + Location
  + SageMaker Neo

## Capítulo 12

A AWS estabelece pilares de excelência operacional visando aplicações que funcionem de maneira confiável e escalável em ambiente de nuvem. Eles também deve ser considerados levando em conta as especificidades para projetos e aplicações de ML

* Tudo como código
  + Usar IaC com o CloudFormation para prover um ambiente de desenvolvimento replicável para os cientistas de dados
  + Usar o Systems Manager para guardar parametrizações gerais utilizadas ao longo de todo o processo de ML
  + Aplicar o paradigma de CAC que consiste em aplicar controle de versão em recursos de configuração
  + Usar também Pipeline as Code
* CI/CD

Automatizar builds frequentes e execuções de testes, bem como mesclar e integrar código continuamente

Não é possível seguir as melhores práticas de DevOps de uma maneira restritiva em um processo de ML devido a sua característica intrinsecamente experimental

A engenharia de características, teste de modelos e código inicial em células de notebooks podem ser feitas inicialmente com SageMaker Studio e em seguida ao obter alguma estabilidade e confiabilidade com conjuntos de dados menores passar a utilizar versionamento com Git ou ferramenta análoga, isso trará capacidade de reprodução, rastreabilidade e capacidade de reverter alterações

Uma outra diferença com relação a desenvolvimento de software tradicional é a necessidade de controlar versão não só do código como também dos datasets por razões regulatórias de interpretação e explicação e de performance. Para isso existe o SageMaker Feature Store

Também é preciso manter um rastreio de versões de pacotes e dependências de apoio utilizadas como bibliotecas e para isso é comum criar imagens Docker que podem ser armazenadas e ter controle de versão no ECR e CodeArtifact

Já a entrega contínua diz respeito a implantação e liberação de versões de software de maneira automatizada nos diferentes ambientes

O CodeDeploy pode ser utilizado para construção do código de pré-processamento e execução de testes e após isso gerar o container que fará o treinamento do modelo. Como o processo de implantação de ML é raramente complemente automático é possível usar o CodePipeline em conjunto com Step Functions para criar etapas de validação e aprovação humana para tanto acurácia e detalhes técnicos do modelo gerado como verificações regulatórias específicas do nicho de negócio

O processo de MLOps tipicamente envolve disponibilizar um modelo para prover previsões através de requisições a um endpoint remoto

Pode usar Sagemaker Pipelines

* Monitoramento contínuo

Após implantado é preciso que monitoremos ao longo do tempo o modelo de ML com relação a sua acurácia e performance. Isso porque pode ocorrer a derivação dos dados que é natural na maioria dos casos de uso devido ao dinamismo de por exemplo, a mudança de preferência de clientes por certos produtos, caso isso não seja levado em consideração a performance do modelo irá se degradar devido a disparidade dos dados reais com os dados no qual foi treinado e as previsões perderão valor

O modelo também pode se degradar devido a mudanças conceituais ou técnicas que não foram previstas e que podem ser levadas em consideração a partir da existência de um adequado monitoramento

Ferramentas do SageMaker para esse propósito são o Ground Truth para gerar novas classificações assertivas contra as quais podemos comparar a saída de nosso modelo e Model Monitor que notifica aos surgirem problemas de qualidade do modelo

O monitoramento também envolve rastrear aspectos de infraestrutura e segurança

O CloudTrail serve para monitorar chamadas de API aos serviços ou alterações e interações feitas por outros usuários neles

* Melhoramento contínuo

Consiste em iterar o processo de ML com novas experimentações ou como resposta a deteriorações detectadas no monitoramento

A melhoria também pode ocorrer como resposta a novas intuições de negócio ou contexto, inovações técnicas e acadêmicas como novas pesquisas

## Capítulo 13

A AWS trabalha com um sistema de responsabilidades compartilhadas no que tange a segurança e para projetos e aplicações ML esse pilar também deve ser considerado. A depender dos componentes arquiteturais (Serviços e infraestrutura escolhidos) nosso escopo no que tange a gestão de segurança irá variar, no modelo compartilhado é descrito que a AWS se responsabiliza pela infraestrutura, localizações físicas e software que dão suporte a seus serviços enquanto cabe aos clientes se responsabilizar pelas aplicações criadas dentro da nuvem e os dados ali trocados

Ao usar o EC2 que é um serviço de computação de propósito geral para trabalhar com ML seremos responsáveis por atualizações de OS e manutenção de firewall enquanto ao usar um serviço gerenciado como o SageMaker isso fica abstraído e faremos controles de mais alto nível usando por exemplo VPC, KMS e IAM

Os seguintes princípios de design devem ser considerados no que tange a segurança

* Proteção dos dados
  + Como a maioria dos processos e serviços de ML trabalham com dados armazenados no S3 a criptografia em repouso que é recomendada, se torna possível tanto do lado cliente quanto servidor com diferentes opções para gestão das chaves de criptografia. Quanto a criptografia em trânsito para tráfego entre VPCs e entre centros de dados da AWS é garantido pelo provedor, já a nível de aplicação o cliente deve usar protocolos como TLS, é possível obrigar criptografia em trânsito para movimentações no S3 por meio de política no IAM
* Isolamento do ambiente de computação
  + Evitar que os recursos alocados de computação como contêineres no ECS tenham exposição a internet sem necessidade, caso um dado container seja um recurso interno e não diretamente usado pelo usuário final podemos o isolar em uma rede privada e endpoints de VPC para interação com os serviços que o suportam, assim toda a comunicação não saí da rede AWS. O suporte a essa funcionalidade em muitos casos é uma premissa para um serviço seja homologado para uso em uma empresa
* Controles fino de acesso
  + Podemos usar o IAM para implementar políticas de acesso do tipo RBAC por função ou ABAC por atributos e criar contas da AWS distintas para cada projeto, ambiente, time ou LOB (Linha de negócio)
* Auditoria e logs
  + Uma boa prática é colocar alarmes para monitorar performance e acurácia dos modelos expostos via API pelo CloudWatch (Monitoramento operacional das cargas de trabalho e aplicações)
  + Determinadas empresas podem ter times dedicados para auditoria e conformidade a fim de verificar aspectos como aplicação de boas práticas da indústria e de TI em geral, bem como aspectos de segurança e outras possíveis regulamentações específicas
  + É possível exportar logs do CloudTrail (Chamadas de API e atividades de usuário) de várias regiões e contas para um S3 centralizado
* Escopo de compliance
  + Um administrador pode usar o AWS Config para criar regras de conformidade para a configuração dos serviços e receber alertas no caso de mudanças nos recursos que quebrem a conformidade
  + O GuardDuty utiliza ML para detectar atividades suspeitas e anomalias nos logs providos pelas ferramentas previamente citadas
  + O AWS Artifact provê documentações relacionadas a regulamentações e regras de conformidade para várias industrias e tipos de aplicações como PCI para o domínio financeiro

Ambientes seguros do SageMaker

Autenticação pode ser feita por meio do IAM, SSO e Identity Federation

Quanto a proteção dos dados é possível utilizar tanto encriptação do lado cliente ou servidor

Pode usar endpoints de VPC possíveis a partir do PrivateLink para comunicação com os outros serviços envolvidos no processo

Se integra ao CloudWatch provendo métricas predefinidas úteis, bem como possibilidade de criar alarmes customizados

Possui regras possíveis de conformidade relacionadas ao uso do serviço em si para se especificar usando o Config

É possível usando políticas a nível de organização que se aplicam a todas as contas optar por não compartilhar ou armazenar dados com os serviços de IA citados até aqui, o que geralmente é feito por padrão

## Capítulo 14

Apesar do caráter experimental do processo de ML, uma vez publicado um modelo em produção são esperados os mesmos critérios de confiabilidade, escalabilidade e resiliência a falhas com relação a uma aplicação de software tradicional

Gerenciamento de mudança

Diz respeito a antecipar, planejar e testar as aplicações em situações de mudança de escala ou outros requisitos de negócio

* Usar IaC sempre que possível para automatizar, replicar e reverter alterações. Sistemas de versionamento de código e dados, bem como ferramentas de automação de implantação de software tornam a gestão de mudanças mais robusta e são quase uma necessidade em arquiteturas e processos atuais
* A instrumentação para aspectos como invocações e uso/saúde de recursos é necessária para cargas de trabalho envolvendo modelos de ML
* Prover um ambiente tão próximo quanto possível do produtivo para testes

Gestão de falhas

Usar as capacidades do S3 como versionamento e trava de deleção é recomendado para os artefatos de modelo, bem como utilizar múltiplas Azs e regiões nas implantações conforme necessário ao caso de uso. Ao usar serviços de IA gerenciados de alto nível aspectos relacionados são abstraídos de nós, enquanto ao trabalhar com os serviços de computação e o SageMaker ficarão a nosso cargo

## Capítulo 15

### Eficiência performática

Privilegiar sempre que possível os serviços gerenciados e Serverless para reduzir a preocupação e responsabilidades do time para com gestão de infraestrutura

#### Seleção

* + De recursos computacionais, armazenamento e rede devem ser orientados a dados e para isso também é necessário conhecimento das opções de serviços disponíveis na AWS
  + Para casos de uso em que há serviços com funcionalidades sobrepostas ou semelhantes testar sempre ambas opções, iniciando pelo serviço mais específico e gerenciado e passando para o mais customizável comparando custos e adequação ao problema
  + Anotar e referenciar a performance e conveniência de diferentes métodos e serviços ao longo de todo o ciclo de vida de ML é preciso para se alcançar o melhor resultado
  + Usar serviços como Debugger e Profiler do SageMaker para observar aspectos de performance
  + Usar IaC para comparar diversas formações diferentes de um ambiente de teste
  + Os processos de ML como preparação dos dados e treinamento usam com frequência arquivos provenientes do S3, esse então deve ser um passo implícito que afete a performance com arquivos massivos ou em grande volume, por isso considere revisitar as classes de armazenamento do S3 e se preciso o uso de EFS ou FSx
  + Reavaliar as escolhas ao longo do tempo ao aferir se o sistema é mais demandante em termos de memória, processamento ou rede

#### Revisão

* + Revisar as escolhas e implementações ao longo do tempo por mudanças na demanda, requisitos, adição de novos serviços, dentre outros fatores será muito mais fácil ao adotar os paradigmas de EaC (Tudo como código) e observação de métricas compatíveis por meio de um processo de revisão de performance que possibilite um ciclo iterativo de PDCA nesse aspecto

#### Monitoramento

#### Compensações

* + Otimizações em performance geralmente vem com um Trade-off de aumento de custo e complexidade em todo o processo que deve ser analisado com relação ao custo benefício de cada decisão

## Capítulo 16

Otimizações de custo

É preciso ter um time com consciência e atenção aos custos devido ao caráter experimental e potencialmente massivo do processamento envolvendo ML. O Cost Explorer permite prever custos com base no histórico

O Budgets pode ser utilizado para criar alertas de custos excessivos através dos quais é possível remediar rapidamente a situação

No Organizations é possível destrinchar os custos entre diferentes contas e projetos

Priorizar serviços gerenciados e Serverless bem como bibliotecas de código aberto para mitigar custos

Usar sempre a abordagem de iniciar com o mínimo possível em termos de alocação e escolhas, foque na simplicidade e escale conforme necessário

Identifique quais modelos tem uso pontual com previsões em batch e quais precisam estar disponíveis em tempo real por meio de Endpoint

Sempre deletar recursos ociosos e se atentar a recursos subutilizados

## Capítulo 17

Novos serviços e atualizações em serviços existentes. São sempre feitos com base em feedback e solicitações dos usuários

### Amazon Monitron

* + Detecção de anormalidades em maquinário industrial usando sensores e ML
  + Solução de ponta a ponta, dispensa a necessidade do próprio usuário integrar os dados dos sensores

### Lookout for Vision

* + Usado para treinar modelos para detectar anomalias em imagens com o intuito de, por exemplo, identificar produtos defeituosos antecipadamente

### Lookout for Metrics

* + Detecta anomalias em dados operacionais ou de negócio
  + Tem conexão nativa com os principais serviços de armazenamento mencionados

### Lookout for Equipment

* + Para prever falhas em equipamentos por meio de sensores
  + Detecção de comportamento anormal em equipamentos utilizando sensores e técnicas de ML

### Panorama

* + Provê dispositivo e SDK para implantar fluxos de visão computacional na ponta

### DevOps Guru

* + Usa ML e integração a vários serviços relacionados a aplicações para detectar problemas operacionais em aplicações

### HealthLake

* + Serviço dedicado a regulamentação HIPAA para que empresas de saúde e casos de uso relacionados a ciências biológicas coloquem seus dados em conformidade

# Cola rápida

* Usar Rekognition para cenários de análise de vídeo e imagem sem conhecimento prévio de machine learning. Geralmente descartar como opção quando enunciado mencionar necessidade de modelo customizado
* Converter texto em áudio: Polly
* Converter áudio em texto: Transcribe
* Tradução de texto entre diferentes idiomas: Translate
* FAQ/Q&A em documentos corporativos/Mecanismo de busca corporativo: Kendra