**CURSO DS EM PRODUCAO**

Sumário

[BONUS 01: PYTHON E GIT PARA DATA SCIENTISTS 2](#_Toc75278653)

[B.1. - Vídeo 1: O que você vai aprender nesse bônus? 2](#_Toc75278654)

[B.1. - Vídeo 2: O que e um ambiente virtual do Python? 2](#_Toc75278655)

[B.1. - Vídeo 3: A Importância do Git no Desenvolvimento de Projetos de Data Science. 3](#_Toc75278656)

[B.1. - Vídeo 4: Explorando o GitHub. 4](#_Toc75278657)

[B.1. - Vídeo 5: As 7 IDEs mais famosas para python. 6](#_Toc75278658)

[B.1. - Vídeo 6: Setup no Linux. 6](#_Toc75278659)

[MODULO 01: ENTENDENDO O PROBLEMA DE NEGOCIO 6](#_Toc75278660)

[M.1. - Vídeo 2: Entenda o problema de negocio com 4 perguntas: 6](#_Toc75278661)

[M.1. - Vídeo 4: Método de Gerenciamento CRISP-DS: 7](#_Toc75278662)

[MODULO 02: A DESCRICAO DOS DADOS 9](#_Toc75278663)

[M.2. - Vídeo 1: Analise Descritiva dos Dados: 9](#_Toc75278664)

[M.2. - Vídeo 2: Analise Como Lidar com os Dados Faltantes: 10](#_Toc75278665)

[M.2. - Vídeo 3: Estatísticas Descritiva: 10](#_Toc75278666)

[MODULO 03: FEATURE ENGENEERING 12](#_Toc75278681)

[M.3. - Vídeo 1: O mapa Mental de Hipóteses: 12](#_Toc75278682)

[M.3. - Vídeo 2: Construindo o MindMap de Hipóteses: 13](#_Toc75278683)

[M.3. - Vídeo 3: Feature Engineering: 14](#_Toc75278684)

[M.3. - Vídeo 4: Filtrando Variáveis: 14](#_Toc75278685)

[MODULO 04: ANALISE EXPLORATORIA DE DADOS 15](#_Toc75278686)

[M.4. - Vídeo 1: Os 3 Objetivos da Analise Exploratoria de Dados: 15](#_Toc75278687)

# 

# BONUS 01: PYTHON E GIT PARA DATA SCIENTISTS

## B.1. - Vídeo 1: O que você vai aprender nesse bônus?

Vídeo 1: Instalando o Python no Windows -> Vídeo 2: Ambiente Virtual no Python -> Vídeo 3: Git e GitHub -> Vídeo 4: As IDE para Python -> Vídeo 5: Configurando o ambiente completo no Linux.

**4 principais ferramentais do data scientist:**

1. Gerenciador das versões do python:

- Gerenciamento de diferentes versões do Python. ( ajuda a manipular diferentes versões do python).

1. Ambiente virtual do Python:

- Organizar dependências de projetos. (Organizam as classes, os pacotes, as variáveis de ambiente, todos os sistemas do projeto). (Serve para que outras pessoas usem as mesmas bibliotecas q eu usei, caso essas outras pessoas utilizem o meu projeto).

1. IDE (Ambiente Desenvolvimento Integrado):

- Software para criar os códigos, scripts.

1. Git e GitHub:

- Versionamento e Armazenamento.

**Instalando o Python no Windows:**

Instalar anaconda.

## B.1. - Vídeo 2: O que e um ambiente virtual do Python?

Um ambiente virtual, único e individual que armazena todas as dependências do projeto (tudo aquilo utilizado para construir o projeto. Ex: IDE’s, as classes, os pacotes, as variáveis de ambiente, todos os sistemas do projeto).

Para outras pessoas poderem utilizar o seu projeto e importante passar o projeto e o ambiente do projeto.

Passar quais bibliotecas e quais versões dessas bibliotecas foram utilizadas. Para isso gera-se um arquivo chamado **requirements.txt**. Neste arquivo, cada linha e uma biblioteca com suas respectivas versões. Ex: pandas == 1.0

Numpy == 1.2

Streamlit == 1.5

Etc...

Cria-se uma pasta chamada .venv e coloca o arquivo **requirements.txt** nesta pasta, em seguida coloca a pasta dentro da pasta do projeto.

**Instalando o ambiente virtual no windows:**

Abre o anaconda, vai em environments, create, nomeia o virtual env, especifica qual pacote de python criar (ex: 3.7), depois clicar em create.

## B.1. - Vídeo 3: A Importância do Git no Desenvolvimento de Projetos de Data Science.

**Git:**

Um sistema de controle de versão de arquivos. Ex: Um software q foi da versão 2 para versão 3, porem a versão 3 mostrou muitos problemas e o melhor a fazer e voltar a versão 2. O git serve para isso, para auxiliar de maneira bem mais fácil o retrocesso de versões de softwares.

Também ajuda na forma como se trabalha colaborativamente em um time. Ajuda a diversas pessoas que trabalham no mesmo projeto a trabalhar de forma mais organizada.

Exemplo:

Projeto de um e –commerce, onde tem um site q vende produtos.

Este projeto e dividido em 3 partes:

Front – end: onde os clientes visualizam e navegam no site (layout);

Back – end: conexão entre o front – end e o banco de dados;

Bancos de dados: armazena todos os dados do projeto.

Dentro do front-end alguns elementos são construídos para ter o front –end pronto:

HTML Layout construído;

Banners (verifica a facilidade de usuabilidade do site pelos clientes);

Request API (acoes feitas pelos clientes no layout do site).

3 pessoas diferentes, cada uma trabalha em uma área do front-end.

* Cenário sem o Git: cada um trabalha individualmente no seu projeto, enquanto um adiciona coisas o outro remove coisas, sem haver comunicação do q cada um faz, gerando trabalho jogado fora, desperdiçado. Com isso a chance de problemas no código e alta.
* Cenário com o Git: cria uma linha do tempo de projeto. Então todas as alterações no projeto e updates no projeto são postados na time line. Essa time line e chamada de máster branch. Cara pessoas também cria uma branch secundaria especifica pra seu elemento. E assim q uma pessoa termina a primeira versão do seu projeto, ela salva essa versão na máster branch para que as outras pessoas vejam e utilizem esta versão. E assim e feito sucessivamente por todos, o que acarreta na não geração de erros.

## B.1. - Vídeo 4: Explorando o GitHub.

**GitHub:**

Um serviço web que oferece diversas funcionalidades extras aplicadas ao git. Uma plataforma que por meio do git, permite vc construir projetos e deixa esses projetos mais visuais.

**Criando projetos no GitHub:**

Clica em repositor;

Clica em new;

Digita o nome do repositório;

Digita uma descrição;

Coloca como publico;

Check box do readme;

Add.gitignore: coloca como python;

Add license: coloca MIT;

Create repository.

**Colocando projetos no GitHub:**

Vai em git for Windows e baixa o git;

Abre a pasta repos (que e a pasta q estão todos os projetos), botão direito dentro da pasta e abre o git bash here, ou abre o git bash digita ls – l e depois digita cd repôs/ para acessar a pasta repos;

No git bash digite git config --global user.name “IgorQueiroz23031988”;

#### Depois digita git config --global user.email “igor.queiroz2303@gmail.com”;

Abre github, e vai no projeto criado;

Cilca em clone or download e copia o link https;

Volta ao git bash digita git clone e cola o https;

Abre o anaconda;

Abre o virtual env criado;

Instal o jupyter notebook;config

Launch jupyter notebook;

Abre o virtual env criado no jupyter notebook;

E começa a criar o projeto;

Volta no git bash

Digita cd e o nome da virtual env;

Pra add no repositório do github digite git add mais o nome do projeto criado no jupyter notebook;

Git commit -m ‘initial commit’;

Git push origin master.

(para remover um projeto da pasta repos pelog it bash e so digitar rm –r e o nome do repositorio)

(ls – l verifica o que tem dentro da pasta repôs)

**Clonando um repositório já pronto:**

Para copiar um projeto já pronto do github para seu coputador, deve-se seguir esses passos

Acesse o projeto do repositorio do github;

Cilca em clone or download;

Abre o git bash;

Git clone mais o link https;

O arquivo requirements.txt mostra todas as bibliotecas usadas neste projeto;

Abre o anaconda;

Vai em environments;

Clica em import;

Cria um novo ambiente virtual;

Na aba specification file coloque o requirements.txt, que esta dentro do repositório clonado;

Para acessar este arquivo, dentro do repositorio que esta dentro de repôs, modifique o tipo de arquivo para pip requirements, que o arquivo aparecera;

Open;

Import;

Este processo instala todas as bibliotecas dentro do requirements.txt.

**Para selecionar o ambiente virtual no prompt de comando do anaconda:**

(base) username % conda env list

(base) username % conda activate project-env

(project-env) username ~ % conda list

Update index no anaconda enviroments/ data-science-em-producao

Criando arquivos vazios .py = type nul > nome\_do\_arquivo.py

Para fazer o freeze com as versões das bibliotecas pip list --format=freeze > requirements.txt

**Para instalar bibliotecas:**

Abre o anaconda;

Vai em environment;

Seleciona o ambiente virtual;

Vai em not installed;

Digita o nome da biblioteca;

Seleciona a biblioteca;

Clica em apply.

**Para instalar outras bibliotecas q não estão no anaconda:**

XGBoost : <https://www.lfd.uci.edu/~gohlke/pythonlibs/>

Busca o lonk XGBoost

Download o pacote

No git bash digita Pip install xgboost-0.90-cp38-cp38-win\_amd64.whl

Boruta

No git bash conda install –c conda-forge boruta\_py

## B.1. - Vídeo 5: As 7 IDEs mais famosas para python.

**O que são IDEs?:**

Integrated Development Environment. Software q tem diversas funções para trabalhar com python.

* Spyder;
* VS Code ( Visual Studio Code) uma das mais novas;
* Google Colab, em cloud, consegue rodar deep learning;
* Jupyter Notebook, roda partes do script independentemente;
* Atom;
* Sublime.
* Pycharm.

## B.1. - Vídeo 6: Setup no Linux.

# MODULO 01: ENTENDENDO O PROBLEMA DE NEGOCIO

## M.1. - Vídeo 2: Entenda o problema de negocio com 4 perguntas:

Não saia resolvendo o problema, sempre questione e tente entender mais sobre o problema.

* Entender: A motivação

Qual e o contexto? R: O CFO requisitou essa solução durante uma reunião de resultados mensais.

* Entender: A causa Raiz do Problema

Porque fazer uma previsão de vendas? R: Investimento em reformas das lajas

* Entender: Quem e o dono do problema

Quem será o Stakeholder (o mais interessado)? R: O CFO

* Entender: O formato da solução

Qual e o formato da solução?

-Granularidade(por data, mês, ano, produto)?R: Vendas diárias por loja, nas próximas 6 semanas.

- Tipo de Problema (classificação, previsão, clusterizacao )?R: Problemas de predição

- Potenciais métodos (rede neural, algoritmo de clusterizacao, outros)? R: Time Series, Regressão e Neural Networks.

- Formato da Entrega (dashboard, csv, imagem, anexo, e-mail, outros)? R: Predições acessadas via celular.

**Entendendo os dados disponíveis:**

<https://www.kaggle.com/c/rossmann-store-sales>

## M.1. - Vídeo 4: Método de Gerenciamento CRISP-DS:

Cross – Industry Process - Data Science.

Método cíclico de desenvolvimento.

**Benefícios:**

* 1º Ciclo Crisp Completo:

- 1 Versão End – to End da solução. ( já consegue entregar valor para o negocio)

- Velocidade na entrega de valor.

- Mapeamento de todos os possíveis problemas em todas as etapas.

**Etapas:**

O ciclo CRISP – DS

Questão de negocio -> Entendimento do negocio -> Coleta dos dados -> Limpeza de dados -> Exploração dos dados -> Modelagem dos dados -> Algoritmo de Machine Learning -> Avaliação do algoritmo -> Modelo em produção.

Questão de negocio: Alguém passa um problema. Ir atrás dos problemas. Fazer mapeamento dos processos das empresas. Identificar os problemas de cada setor.

Entendimento do negocio: Explicação no Vídeo 2.

Coleta dos dados: Uso de SQL. Juntar todos os dados em uma grande tabela.

Limpeza de dados: Verificar dados.

Exploração dos dados: Entendem-se as variáveis que impactam o problema q esta sendo modelado. Exemplo: no modelo de previsão de vendas, nesta etapa e necessário entender o q puxa as vendas pra cima e pra baixo, quais são os fatos q influenciam na venda. Testes estatísticos são aplicados pra verificar as correlações entre estes atributos.

Modelagem dos dados: Aqui se prepara os dados para ensinar os algoritmos de ML. Separar dados em treinamento e teste, fazer encoding de variáveis categóricas (transforma as variáveis categóricas em numéricas), transformar dados, etc.

Algoritmo de Machine Learning: Aplicar o algoritmo sobre os dados.

Avaliação do algoritmo: Criar um conjunto de erros que fazem sentindo para problema, para ter uma visão da acurácia dos modelos. Ex: acurácia, tabela verdade, curva rock, etc. Se a desempenho já esta como esperada, já pode ser usada pelas pessoas.

Modelo em produção: Se a desempenho do modelo já esta como esperada, já pode ser usada pelas pessoas, já pode colocar o modelo em produção.

**Preocupação deste método não e a acurácia do modelo em produção, mas sim terminar o ciclo, entregar valor para negocio minimamente e descobrir todos os possíveis problemas q podem acontecer durante o desenvolvimento da solução.**

**Depois de terminar o ciclo, volta-se para o Entendimento do negocio.**

**CRISP – DS no desafio da Rossmann:**

**- Questão do negocio:**

Qual e o valor das vendas de cada loja nas próximas 6 semanas?

**- Entendimento do negocio:**

* Qual e a motivação?

- A previsão de vendas foi requisitada pelo CFO em uma reunião mensal sobre os resultados das lojas.

* Qual e a causa Raiz do Problema?

- Dificuldade em determinar o valor do investimento para reformas de cada loja.

* Quem e o dono do problema?

- Diretor financeiro (CFO) da Rossmann.

* Qual e o formato da solução?

- Granularidade: Previsão de vendas diárias por loja, nas próximas 6 semanas, 42 dias.

- Tipo de Problema: Problemas de predição (previsão de venda)

- Potenciais métodos: Series temporais, Regressão.

- Formato da Entrega: O valor total de vendas no final de cada loja na sexta semana ( vai ver a coluna com o ID da loja e outra coluna com o valor total de cada loja); Como essas vendas acontecem dia a dia; Checar esses valores pelo celular.

**Update do Progresso do Projeto:**

Informar o CFO em q etapa esta o projeto.

Tem como estimar o tempo do projeto com isso.

# MODULO 02: A DESCRICAO DOS DADOS

## M.2. - Vídeo 1: Analise Descritiva dos Dados:

**Update do Progresso do Projeto Rossmann:**

Nesta etapa já se sabe a **questão de negocio**, o **entendimento do negócio**, e já foi feito a **coleta dos dados**.

Agora começaremos a **limpeza dos dados**, que envolve a descrição dos dados.

**A Motivação da Descrição dos Dados:**

“O quão desafiador e o problema q estou lidando?”.

* Quantidade de dados:

- Eu tenho os recursos corretos para trabalhar? R: pc normal suporta 10 mil linhas por exemplo.

- Para 1 trilhão de linhas eu preciso de ferramentas como: Servidores, Clusters, (Spark, Hadoop) linguagens q processam dados massivamente.

* Tipos de variáveis:

- Quais são os tipos de variáveis?

- Quais são as porcentagens numéricas, categóricas, temporal? Isso guia vc a escolher as técnicas de analise.

* Quantidade de dados faltantes:

- Qual e o voluma de NA?

- Qual e a razão de faltar dados? Culpa do sistema ou do input manual (erro humano)?

* Resumo Geral dos dados:

- Estatística descritiva: Usado para ter uma noção de grandeza dos dados; noção de limites mínimos e máximos das variáveis; qual e o range; qual e a mediana e outros.

**Descrição dos dados no python:**

Feito no jupyter notebook

**Check NA:**

3 maneiras de tratar os NA:

* Primeira: joga as linhas fora (rápido e fácil, porem e prejudicial para quantidade muito grande, perdendo informação e prejudicando a desempenho do modelo).
* Segunda: usando algoritmos de ml, tem como substituir as linhas vazias por outros valores baseados no comportamento da coluna, ex: calcular a mediana ou a media e substituir os NAs pela mediana ou media; ou usar ml para prever os valores q podem ser colocados no lugar dos NAs.
* Terceira: Entendendo o negocio. Descobrir pq o NA ta aki? (mais importante)

## M.2. - Vídeo 2: Analise Como Lidar com os Dados Faltantes:

**Substituição dos dados vazios em python:**

Feito no jupyter notebook

## M.2. - Vídeo 3: Estatísticas Descritiva:

**Numerical Attributes:**

Serve para ganhar conhecimento de negocio e para conseguir detectar erros. Existem duas grandes métricas dentro da estatística descritiva:

* Métricas de tendência central: resumem os dados em um único numero. Ex: media e mediana.
* Métricas de dispersão: Dizem se os dados estão muito concentrados perto da media ou muito dispersos das mesmas. Ex: Variância, desvio padrão, mínimo e máximo, range, **Skew e Kurtosis.**

A Skew fala como q e a deformação da minha distribuição em relação a normal (distribuição forma de sino). Deformação mais pra esquerda a skew e positiva, deformação mais par direita a skew e negativa.

A Kurtosis fala sobre a concentração dos dados. Quanto maior a Kurtosis positiva significa q tem mais dados concentrados, então tem um pico muito alto. Com os dados com um pico menor, ou seja, mais dispersos, significa que a Kurtosis e negativa.

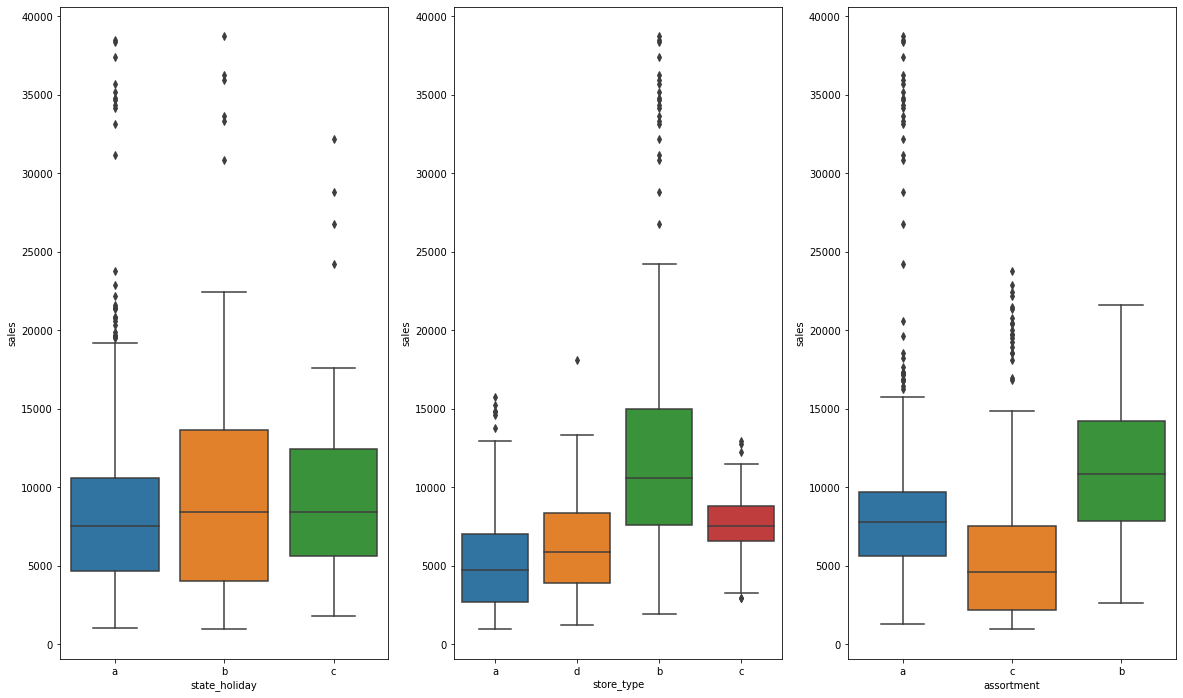
**Exemplo de uso das Estatísticas Descritivas:**

Se caso existir algum valor de venda negativo, na hora de verificar o valor mínimo de venda na Estatística Descritiva vai aparecer valor negativo, o q significa erro para este tipo de coluna.

**Categorical Attributes:**

Para categorical attirbutes usa-se o boxplot, onde mostra todas as medidas de dispersão e central em um lugar só.

**Funcionamento do boxplot:**



A linha do meio do quadrado colorido e a 50º percentil ou seja e a mediana;

A linha de baixo do quadrado colorido e a 25º percentil;

A linha de cima do quadrado colorido e a 75º percentil;

As linhas extremas são 0º e 100º percentil.

Este retângulo e dividido em 4 quartis:

O 1º quartil esta entre os percentis 0 e 25;

O 2º quartil esta entre os percentis 25 e 50;

O 3º quartil esta entre os percentis 50 e 75;

O 4º quartil esta entre os percentis 75 e 100.

Tudo fora dos valores mínimos e máximos são considerados outlires, q são considerados como 3 vezes o desvio padrão.

**EStatisticas Descritiva em python:**

Feito no jupyter notebook

# MODULO 03: FEATURE ENGENEERING

## M.3. - Vídeo 1: O mapa Mental de Hipóteses:

**Update do Progresso do Projeto Rossmann:**

Nesta etapa já se sabe a **questão de negocio**, o **entendimento do negócio**, já foi feito a **coleta dos dados** e a **limpeza dos dados**.

Agora começaremos a **Exploração dos dados**, que tem 3 principais tarefas:

* A primeira parte que e Feature Engineering (criação e derivação das variáveis).
* A segunda parte: Criar uma lista de hipóteses.
* A terceira: Validar esta lista de hipóteses.

Agora trataremos da **Feature Engineering.**

**Mapa Mental de Hipóteses:**

Roteiro que mostra quais são as analises que vc precisa fazer para poder validar as hipóteses, e para validar as hipóteses e fazer as analises, quais variáveis vc precisa derivar.

3 elementos que constituem o mapa mental de hipóteses:

1. Fenômeno: Aquilo q vc esta tentando medir ou tentando modelar:

- Qual fenômeno estou modelando? Ex: Vendas, detecção de objetos em imagem, clusterizacao de clientes para criação de personas, etc...

1. Agentes: São todas aquelas entidades que de alguma forma impactam o seu fenômeno.

- Quem são os agentes que atuam sobre o fenômeno de interesse? Ex: clientes, lojas, produto, etc...

1. Atributos dos Agentes: Descreve-se cada agente com certos atributos.

- Qual a descrição dos clientes? EX: cliente tem idade, escolaridade, estado civil, numero de filhos, frequência de ida a loja, etc...

**O objetivo do mapa e derivar uma lista de hipóteses.**

1. Lista de Hipóteses:

- Hipóteses para validar com os dados. E feita uma analise dos dados para poder validar essas hipóteses, e cada hipótese validada ou descartada e chamado **de insights.**

**- A criação de insights e gerado através da surpresa e através da contraposição da crença do CEO.**

**Criação do mapa mental de hipóteses:**

<https://coggle.it/>. Site utilizado para criar mapas mentais de hipóteses.

**A Motivação da Feature Engineering:**

“Ter as variáveis **disponíveis** para **estudo** durante a Analise Exploratória de Dados.”.

Criar variáveis no meio da analise exploratória de dados e ruim, pois a estrutura dos código fica totalmente bagunçado, dificultando o entendimento do código para outras pessoas e ate pra si mesmo.

## M.3. - Vídeo 2: Construindo o MindMap de Hipóteses:

**Como escrever as Hipóteses?**

Hipóteses são apostas, e o que vc acha sobre alguma coisa.

Ex: Lojas de **Maior** **Porte** deveriam **Vender** **Mais**.

1. Porte e um atributo da entidade loja. Vender e a variável resposta, que e fenômeno e ser modelado.
2. As palavras azuis são as apostas.

Lojas de **Maior** **Sortimento** deveriam **Vender** **Mais**.

1. Sortimento e um atributo da entidade loja. Vender e a variável resposta, que e fenômeno e ser modelado.
2. As palavras azuis são as apostas.

Lojas de **Mais** **Competidores por perto** deveriam **Vender** **Menos**.

1. Competidores por perto e um atributo da entidade loja. Vender e a variável resposta, que e fenômeno e ser modelado.
2. As palavras azuis são as apostas.

**Lista Final de hipóteses:**

Aqui se prioriza as hipóteses.

Para isso verifica-se se tem ou não o dado disponível no dataset para fazer a hipótese.

Caso com essas hipóteses o modelo tenha um desempeno ruim, ai sim se cria novos dados para aplicar as novas hipóteses.

## M.3. - Vídeo 3: Feature Engineering:

**Como derivar features?**

Feito no jupyter notebook

## M.3. - Vídeo 4: Filtrando Variáveis:

**A motivação de filtragem de variáveis:**

“Restrições de Negocio”

As vezes o projeto não da certo pq não se deu atenção necessária as restrições de negocio.

**Filtrar Variáveis x Selecionar Variáveis:**

1. Filtragem de Variáveis:

- Restrições de acordo com o negocio. Ex: Variáveis que não podem ser usadas para aplicação do modelo de ML.

- Dados Disponíveis no momento. Ex: Se quer uma previsão para daqui a 3 meses, porem a variável ‘x’ não tem esses dados a partir de agora ate daqui a 3 meses, então não se pode utilizar esta variável pois ela não tem informações futuras.

1. Seleção de Variáveis:

- Variáveis mais relevantes para o modelo. Ex: Verificar quais variáveis tem alta correlação com a variável resposta (output), e utiliza-las no modelo.

**Filtragem de variáveis:**

Feito no jupyter notebook

# MODULO 04: ANALISE EXPLORATORIA DE DADOS (EDA)

## M.4. - Vídeo 1: Os 3 Objetivos da Analise Exploratória de Dados:

**Update do Progresso do Projeto Rossmann:**

Nesta etapa já se sabe a **questão de negocio**, o **entendimento do negócio**, já foi feito a **coleta dos dados,** a **limpeza dos dados** e **Exploração dos dados (Feature Engineering).**

Agora começaremos o restante da **Exploração dos dados**.

**A Motivação da Analise Exploratória de Dados:**

“ Como as variáveis impactam o fenômeno? E Qual a forca desse impacto?”

Mede o imacto das variaveis em relação a variável resposta. E muitas vezes tentar quantificar este impacto.

**Quais são os 3 Objetivos da EDA?**

1. Ganhar experiência de negocio: Entender como o negocio funciona, o comportamento que o negocio tem através dos dados.
2. Validar hipóteses de negocio (Insights): Surpreender e/ou alterar a crença do CEO.
3. Perceber variáveis que são importantes para o modelo: Quais variáveis impactam o fenômeno.

**Analise Exploratória de Dados (EDA):**

Constituída de 3 tipos de analises:

1. Analise Univariada: Onde vc se importa basicamente só com a variável.

- Como e essa variável?

- Quais são seus valores de (min, max, distribuição, range, etc...)

1. Analise Bivariada: Analisa uma única variável em relação a variável resposta.

- Como que a variável impacta a resposta?

- Impacta significa correlação e validação de hipóteses.

- Qual e o impacto dessa variável na variável resposta?

1. Analise Multivariada: Analisa varias variáveis de uma vez em a variável resposta, e tbm entre elas mesmas.

- Como as variáveis se relacionam?

**Analise Exploratória de Dados (EDA), codificação:**

Feito no jupyter notebook.

**Variável Resposta (output) - Distribuição Normal :**

Quanto mais normal a variável resposta, melhor o algoritmo vai performar.

Mais normal significa que a dispersão da variável e em forma de sino e bem centrada, ou seja, em um gráfico de dispersão o sino fica bem no meio do gráfico.

**Variável Resposta (output) - Distribuição Poason :**

Pesquisar online

## M.4. - Vídeo 2: Analise Exploratória de Dados – Parte 1:

**Analise Exploratória de Dados (EDA), codificação:**

Feito no jupyter notebook.

## M.4. - Vídeo 3: Analise Exploratória de Dados – Parte 2:

**Analise Exploratória de Dados (EDA), codificação:**

Feito no jupyter notebook.

## M.4. - Vídeo 4: Analise Exploratória de Dados – Parte 3:

**Analise Exploratória de Dados (EDA), codificação:**

Feito no jupyter notebook.

## M.4. - Vídeo 5: Analise Exploratória de Dados – Conclusão:

**Analise Exploratória de Dados (EDA), codificação:**

Feito no jupyter notebook.

## M.4. - Vídeo 6: Analise Multivariada:

**A Motivação da Analise Multivariada:**

Para obter um bom modelo sempre deve-se tentar reduzir a complexidade do próprio modelo.

Quanto maior a dimensionalidade do dataset maior e a complexidade do modelo.

Para reduzir essa dimensionalidade/complexidade deve-se encontar variáveis com o mesmo tipo de informação. Para fazer isso usa-se a correlação entre estas variáveis. Se estas variáveis são altamente correlacionada, pode-se eliminar uma delas e menos assim o nível de informação se mantem.

Ou seja, A analise multivariada:

Ajuda a olhar as variáveis que são correlacionadas e, portanto podem ser retiradas para diminuir a dimensionalidade do dataset, assim diminuindo sua complexidade.

**Variáveis Numéricas:**

1. **Correlação de Pearson:**

Correlação entra a variável e a variável resposta.

Método de pearson somente usado em variáveis numéricas.

Quanto mais próximo de 1 ou -1 mais alta e a correlação, quanto mais próximo de 0 mais fraca e a correlação. Quanto maior a correlação mais impacto a variável tem sobre a variável resposta, significando que, quanto maior a correlação melhor pode ser a performance do modelo.

Se a correlação for linear, o valor tende a ser 1 ou -1. Quanto mais próximo de 0 menos linear é a correlação.

Variável linear geralmente com dois valores, sim ou não, tem ou não tem, 0 ou 1, etc.

- Tbm ocorre correlação entre as variáveis não resposta.

**Variáveis Categoricas:**

1. **Correlação de Cramer V:**

Correlação entra a variável e a variável resposta.

Método de cramer V somente usado em variáveis catgoricas.

Vai de 0 a 1;

Quanto mais perto do 0 menor a correlação;

Quanto mais perto do 1 maior a correlação.

- Tbm ocorre correlação entre as variáveis não resposta.

# MODULO 05: PREPARACAO DOS DADOS

## M.5. - Vídeo 1: Os 3 Tipos de Preparação dos Dados:

**Update do Progresso do Projeto Rossmann:**

Nesta etapa já se sabe a **questão de negocio**, o **entendimento do negócio**, já foi feito a **coleta dos dados,** a **limpeza dos dados** e **Exploração dos dados.**

Agora começaremos a **Modelagem dos dados**.

**A Motivação da Preparação dos Dados:**

“O aprendizado da maioria dos algoritmos de ML e facilitado **com dados numéricos, na mesma escala**.”

E necessário fazer o encoding de variáveis categóricas, ou seja, transforma-las em numéricas.

E e necessário uma normalização das escalas das variáveis, para que todas as variáveis fiquem no mesmo range, e para q não haja beneficio para as variáveis com maiores range, na hora de fazer o modelo ML.

**Os 3 tipos de preparação dos dados:**

1. **Normalização:** Rescala o centro para 0 com desvio-padrão igual a 1.

- Funciona muito bem em variáveis com distribuição normal.

1. **Rescaling:** Rescala para o intervalo entre 0 e 1.

- E utilizado em variáveis que **NÃO** tem distribuições Gaussianas, ou seja, **NÃO** tem distribuição normal.

1. **Transformação:** Conversão de features categóricas para numéricas (encoding).

- Transformação de natureza. EX: Meses que vao do 1 ao 12 transformado em uma forma cíclica.

**Normalização:**

Xnew = (x1 – mean) / std

mean = media

std = desvio padrao

**Normalização dos dados em python:**

Feito no jupyter notebook.

## M.5. - Vídeo 2: Rescaling de Variáveis Numericas:

**Rescaling (Min – Max Scaler):**

Xnew = (xi – mean) / (xmax – xmin)

Mean = media

Super sensivel para outliers.

Para resolver este problema dos outliers, usa-se os intervalos dos quartis. Esta técnica e nomeada de Robust Scaler.

**Rescaling (Robust Scaler):**

Xnew = (xi – Q1(x)) / (Q3(x) – Q1(x))

Q1(x)) = 1º quartil

Q3(x)) = 3º quartil

As distribuições das variáveis, aqui ficam mais apertadas.

Variáveis com outliers muito forte: aplica-se o Min-Max Scaler;

Variáveis sem outliers muito forte: aplica-se o Robust Scaler.

**Rescaling dos dados em python:**

Feito no jupyter notebook.

## M.5. - Vídeo 3: Encoding de VAriaveis Categoricas:

Convertendo uma variável categórica em numérica, mantendo o conteúdo de informação.

**Tipos de Transformação Encoding:**

Dicas para encontrar o melhor encoding para as variáveis que eu estou trabalhando:

1. Tenha o feeling dos dados. Faca a AED para ter esse feeling. Conheca cada variavel.
2. Teste.
3. **One Hot Encoding:** Cria novas colunas, uma para cada valor da coluna original. Coloca 1 para a coluna valor daquela linha, e resto 0 para todos os outros valores. Ex: Coluna cor, que tem 3 valores: verde, azul e vermelho. A linha 1 mostra cor verde, então cria-se 3 colunas no dataset, cada coluna sendo uma cor, e coloca 1 na coluna verde e 0 nas colunas azul e vermelho.

-Simples de aplicar, mas cria muitas novas colunas no dataset, aumentando a dimensionalidade do dataset, prejudicando a aprendizagem do modelo, tornando-o overfitado.

-Usado em variáveis q apresentam ideia de estado, exemplo feriado, onde tem um estado de dias regulares e estados de dias de feriado, onde cada um demonstra um comportamento diferente.

1. **Label Encoding:** Troca os valores da variável por numero. Ex: Valores: vermelho, verde e azul viram 1, 2 e 3, respectivamente.

- Funciona muito bem em variáveis q são apenas nomes, ex: cor, onde não tem uma escala, são nomes separados.

1. **Ordinal Encoding:** Igual ao label, porem respeita uma ordem de escala. Ex: Temperatura que vai da mais baixa (frio) ate a mais alta (muito quente). Então esse encoding coloca uma numeração crescente para cada valor de temperatura, onde frio e 1 e muito quente e 4 por exemplo.
2. **Target Encoding:** Funciona muito bem para varias que tem muitos mas muitos valores.

-Calcula a media de vendas para cada um dos valores da variável categórica, e substitui o valor da variavel por esse numero calculado. Ex: a media de venda do valor verde foi 0.8, então substitui-se verde por 0.8.

1. **Frequence Encoding:** Igual ao target encoding, porem em vez de media, faz-se a contagem de vendas cada valor.

- Usado para marcar de carro por exemplo.

1. **Embedding Encoding:** Método implícito dentre das redes neurais profundas.

- Muito usado em NLP, e deep learning.

**Transformação Encoding dos dados em python:**

Feito no jupyter notebook.

**IMPORTANTE:**​O link da Referencia sobre os tipos de Encoding para variáveis categóricas foi alterado. O novo link é esse:  
[http://contrib.scikit-learn.org/category\_encoders/](http://contrib.scikit-learn.org/category_encoders/" \t "_blank)

## M.5. - Vídeo 4: Transformação da Natureza das Variáveis:

**Grandeza x Natureza:**

1. **Grandeza:** Tem como objetivo trazer a variável resposta com uma distribuição mais próxima possível de uma normal, para melhorar a acurácia do modelo de ML.

-Tipos de Transformação de Grandeza:

1. **Logarithm Transformation:** Transforma distribuições com skew, tanto pra esquerda quando pra direita, em uma distribuição mais próxima da normal.
2. **Box-Cox Transformation:** Faz a mesma coisa q a anterior. De modo diferente.
3. **Cube-Root Transformation:** Faz a mesma coisa q a anterior. De modo diferente.
4. **Square-Root Transformation:** Faz a mesma coisa q a anterior. De modo diferente.
5. **Sine and Cosine Transformation:** Faz a mesma coisa q a anterior. De modo diferente.
6. **Natureza** Tem como objetivo trazer a natureza real dos seus dados dentro do conjunto de dados. Ex: Serve para variáveis de valores cíclicos, exemplo meses, q tem valores cíclicos de 1 a 12 e depois volta para o 1. Porem variáveis assim mostra um problema, a distancia de 1 a 12 não e a mesma de 1 a 2, pois o ciclo vira no 12º mês. Então a transformação por natureza tenta concertar esse problema, tornando a variável com valores cíclicos no dataset, ou seja, colocando a distancia de 12 a 1 igual a 1 para 2.

**-** Coloca os meses em um circulo, para ter a distancia entre meses iguais para todos os meses. Como se faz isso no dataset? Cria-se duas novas colunas para a variavel mês, uma sendo o seno e a outra sendo o coseno.

**Transformação por natureza e grandeza dos dados em python:**

Feito no jupyter notebook.

# MODULO 06: FEATURE SELECTION

## M.6. - Vídeo 1: A Motivação da Seleção de Variáveis:

**Update do Progresso do Projeto Rossmann:**

Nesta etapa já se sabe a **questão de negocio**, o **entendimento do negócio**, já foi feito a **coleta dos dados,** a **limpeza dos dados** e **Exploração dos dados.**

Agora começaremos a **Modelagem dos dados ( Feature Selection)**.

**A Motivação da Seleção de Variáveis:**

“A explicação mais simples sobre um fenômeno observado, deveria prevalecer sobre explicações mais complexa.” ( Occam’s Razor)

**Principio da Navalha de Occam:**

Se vc tem dois modelos q representam o mesmo fenômeno, de preferencia pra aquele que e mais simples, pois provavelmente o modelo mais simples generaliza melhor.

**Variáveis Colineares:**

Para vc tornar mais simples a aprendizagem em ML, vc precisa utilizar modelos mais simples, ou seja, com menos variáveis no dataset.

Porem existem variáveis que explicam a mesma parte do fenômeno, essas variáveis são chamadas colineares.

## M.6. - Vídeo 2: Os Metodos de Feature Selection:

**Os Tipos de Seleção de Variáveis:**

1. **Seleção Univariada (Filter Methods):** Classifica as variáveis como relevantes e não relevantes para o dataset.

- Tem como objetivo identificar quais das variáveis explicam a maior porção do fenômeno, ou explicam melhor o fenômeno.

- Variáveis que tem uma porção pequena de explicação podem ser excluídas.

- Para fazer essa identificação, a Seleção Univariada utiliza o **Coeficiente de Correlação** (explicado no modulo 4). Quanto mais próximo de 1 ou -1 relevância alta, mais próximo de 0 relevância baixa. Este coeficiente explica a intensidade da relação entre duas variáveis.

- Correlação 1 (positiva). Quanto mais os anos crescem mais as vendas crescem.

- Correlação -1 (negativa). Quanto mais o preço aumenta, mais diminuem as vendas.

- Tipos de correlação para os tipos de variáveis:

Variável Resposta

|  |  |  |
| --- | --- | --- |
| Variaveis Preditoras | **Numérica** | **Categórica** |
| **Numérica** | **Correlacao de Pearson** | **Linear Discriminant Analysis (LDA)** |
| **Categórica** | **Anova** | **Chi-Square / Cramer’s V** |

- Vantagens: Método simples, rápido e fácil entendimento.

- Desvantagens: Não considera a influencia entre as variáveis. Onde só correlaciona a variável preditora sozinha com a variável resposta. Isso pode apresentar problema, pois, esta variável sozinha pode apresentar uma correlação bem fraca, porem essa variável combinada com outra variável preditora pode apresentar uma correlação bem alta com a variável resposta.

1. **Seleção por Importância (Embedded Methods):** Classifica as variáveis como relevantes e não relevantes para o dataset.

- Tem como objetivo identificar quais das variáveis explicam a maior porção do fenômeno, ou explicam melhor o fenômeno.

- Variáveis que tem uma porção pequena de explicação podem ser excluídas.

- Tbm conhecido como método embutido. O método de seleção por importância e um subproduto do próprio aprendizado de alguns algoritmos. Ex: No Random Forest, para fazer o aprendizado de ML, o próprio algoritmo seleciona as melhores vaiáveis para este processo. Outros algoritmos que fazem a mesma coisa são os algoritmos de regressão Lasso e Ridge.

- No método de regressão, o algoritmo da um peso para cada uma das variáveis, e este peso da uma noção da importância das variáveis. Variáveis com peso alto tem grande importância para o modelo.Variáveis com peso baixo são quase que irrelevante para o modelo.

- Random Forest: Usa um método chamado Gini Impurity. Esse algoritmo seleciona as variáveis e divide o dataset original em dois outros datasets. Nesta divisão, O impurity seleciona as variaveis que demonstra uma maior homogeneidade entre os dois datasets, ou seja, escolha as variáveis dos dois datasets para que os dois dadasets se assemelham o maior percentual possível.

- Ex: O Random Forest seleciona a variável preço, e divido o dadaset em dois datasets, dataset 1 onde tem os preços baixos e dataset 2 onde tem os preços altos. Se esses dois datasets forem homogêneos, ou seja, parecidos, em relação a distribuição, quantidade de valores, etc, significa que esta variável possui alta relevância para o modelo. SE esses dois datasets forem diferentes nessas características mencionadas acima, significa que esta variável não e relevante para o modelo.

- Logicamente que existem cálculos matemáticos, como entropia, que dizem se os datasets criados são homogêneos ou não.

- A entropia da um percentual para cada variável, então quanto maior a entropia da variável mais relevante ela e para o modelo.

- Com isso a entropia cria uma lista de maior relevância para menor relevância das variáveis.

- Os algoritmos de regressão Lasso e Ridge tem mais ou menos a mesma ideia de entropia e criação de lista de relevância. Nesses dois cria-se uma lista com as variáveis com peso mais próximos de 1 ou -1, as variáveis mais próximos de 0 tem relevância quase nula. Porem invés de entropia usa-se o peso das variáveis.

- Vantagens: Considera a influencia entre as variáveis. Onde pode combinar variáveis para correlacionar com a variável resposta e demonstrar uma maior correlação. Acredito que seja mais confiável tbm.

- Desvantagens: Método mais complicado, demorado, pesado computacionalmente e de difícil entendimento.

1. **Seleção por SubSet (Wrapper Methods):** Classifica as variáveis como relevantes e não relevantes para o dataset.

- Tem como objetivo identificar quais das variáveis explicam a maior porção do fenômeno, ou explicam melhor o fenômeno.

- Variáveis que tem uma porção pequena de explicação podem ser excluídas.

- Método totalmente diferente dos outros dois.

- Processo de seleção por SubSet:

- Selecione uma única variável aleatoriamente.

\_ Com essa variável mais a variável resposta, treina-se um modelo de ML.

- Calcula-se a performance do modelo, em termos de acurácia ou erro.

- assume-se que essa variável mais a variável resposta conseguiu uma acurácia de 62%.

- Agora faz-se a pergunta: a performance aumentou?

- Agora adiciona uma nova variável aleatória junta a primeira variável e a variável resposta.

- Treina-se o modelo ML novamente.

- Calcula-se a performance do modelo, em termos de acurácia ou erro, novamente.

- Vamos dizer que agora a acurácia e de 65%.

- Agora se faz a pergunta novamente: a performance aumentou?

- Nesse caso sim. Então mantem a variável.

- Agora adiciona uma terceira variável aleatória, repete todo o processo, e verifica se a acurácia aumentou.

- Nesse caso, a acurácia manteve 65%.

- Agora se faz a pergunta novamente: a performance aumentou?

- Nesse caso não. Então remove essa terceira variável adicionada por ultimo.

- Repete esse processo para todas as variáveis do dataset.

- No final, tem-se um subset somente com as variáveis mais relevantes.

- Um algoritmo utilizado para fazer este processo chama-se **Boruta**.

- Funcionamento do **Boruta**:

- No dataset original, este algoritmo seleciona duas variáveis aleatórias.

- Ele cria variáveis shadow, que são copias das variáveis originais, porem com as linhas embaralhadas.

- Essa mistura remove a correlação entre as variáveis originais e as shadows.

- Agora treina-se o modelo, que geralmente e um random forest.

- Com esse modelo, encontra-se a importância de cada variável, tanto das originais quanto das shadows.

- Imaginado o cenário que: v1 tem 0.6, v2 tem 0.2, s1 tem 0.03 e s2 tem 0.4.

- Agora ele verifica qual variável shadow e a mais importante, que e a s2 com 0.4.

- Agora ele compara a importância de cada variável original com a variável shadow s2.

- Se a variável original tiver uma importância maior que a variável shadow s2, coloca-se 1 para esta variável original, caso contrario coloca-se 0 pra variável original tbm.

- Aqui, por exemplo, como a importância de v1 e mairo q v2, então v1 coloca-se 1 e v2 coloca-se 0.

- repete-se esse processo de embaralhamento para v1, v2, s1 e s2 varias vezes, para criar uma tabela de v1 e v2 chamada Success Count.

-Com essa tabela Success Count, te como fazer uma distribuição binomial para v1 e v2, com o intuito de achar o p-value de v1 e v2, neste caso v1 com 0.125 e v2 com 0.007.

- Esse p-value ajuda a determinar se uma hipótese e aceita ou rejeitada.

- Depois de calcular o p-value das variáveis originais, constrói-se a distribuição binomial e seta um limite do p-value.

- Se o p-value da variável original for menor que o p-value limite, nesse caso 0.01 por exemplo, confirma-se que a variável e importante, se for maior confirma-se que a variável não e importante, logo rejeitada.

- Neste caso v2 e aceita e v1 e rejeitada.

- Faz-se esse processo todo com todas as variáveis do dataset ate ter um dataset reduzido com a variáveis mais relevantes.

- Vantagens:. Acredito que seja mais confiável tbm.

- Desvantagens: Método mais complicado, muito demorado, muito pesado computacionalmente e de difícil entendimento.

## M.6. - Vídeo 5: Selecionando Variáveis com o Boruta:

Feito no jupyter notebook.

Quando da discrepância na analise de importância entre a analise exploratória de dados e o boruta, no primeiro ciclo do crisp, se da preferencia ao boruta, nos ciclos seguintes adiciona a variável q foi dita como importante na EDA. Exemplo: o ano e relevante na EDA, mas não e relevante no boruta, então primeiro cria-se um modelo de ML sem a coluna ano, e no segundo ciclo adiciona o ano pra verificar se a acurácia melhorou ou não.

# MODULO 07: MODELOS DE MACHINE LEARNING

## M.7. - Vídeo 1: Machine Learning Supervisionado:

**Update do Progresso do Projeto Rossmann:**

Nesta etapa já se sabe a **questão de negocio (Modulo 1)**, o **entendimento do negócio (Modulo 1)**, já foi feito a **coleta dos dados (kaggle),** a **limpeza dos dados(Módulos 2 e 3)** , **Exploração dos dados(Modulo 4)**, e **Modelagem dos dados (Módulos 5 e 6)**.

Agora começaremos a **Algoritmos de Machine Learning (Modulo 7).**

**A Motivação dos Modelos de Machine Learning:**

“Aprender o comportamento de um fenômeno (nesse caso vendas) com as variáveis disponíveis e então generaliza-lo para o futuro.”

**Tipos de variáveis:**

Variável é a característica de interesse que é medida em cada elemento da amostra ou população. Como o nome diz, seus valores variam de elemento para elemento. As variáveis podem ter valores numéricos ou não numéricos.

Variáveis podem ser classificadas da seguinte forma:

**Variáveis Quantitativas:** são as características que podem ser medidas em uma escala quantitativa, ou seja, apresentam valores numéricos que fazem sentido. Podem ser contínuas ou discretas.

**Variáveis discretas:** características mensuráveis que podem assumir apenas um número finito ou infinito contável de valores e, assim, somente fazem sentido valores inteiros. Geralmente são o resultado de contagens. Exemplos: número de filhos, número de bactérias por litro de leite, número de cigarros fumados por dia.

**Variáveis contínuas**, características mensuráveis que assumem valores em uma escala contínua (na reta real), para as quais valores fracionais fazem sentido. Usualmente devem ser medidas através de algum instrumento. Exemplos: peso (balança), altura (régua), tempo (relógio), pressão arterial, idade.

**Variáveis Qualitativas (ou categóricas)**: são as características que não possuem valores quantitativos, mas, ao contrário, são definidas por várias categorias, ou seja, representam uma classificação dos indivíduos. Podem ser nominais ou ordinais.

**Variáveis nominais**: não existe ordenação dentre as categorias. Exemplos: sexo, cor dos olhos, fumante/não fumante, doente/sadio.

**Variáveis ordinais**: existe uma ordenação entre as categorias. Exemplos: escolaridade (1o, 2o, 3o graus), estágio da doença (inicial, intermediário, terminal), mês de observação (janeiro, fevereiro,..., dezembro).

As distinções são menos rígidas do que a descrição acima insinua.

Uma variável originalmente quantitativa pode ser coletada de forma qualitativa.

Por exemplo, a variável idade, medida em anos completos, é quantitativa (contínua); mas, se for informada apenas a faixa etária (0 a 5 anos, 6 a 10 anos, etc...), é qualitativa (ordinal). Outro exemplo é o peso dos lutadores de boxe, uma variável quantitativa (contínua) se trabalhamos com o valor obtido na balança, mas qualitativa (ordinal) se o classificarmos nas categorias do boxe (peso-pena, peso-leve, peso-pesado, etc.).

Outro ponto importante é que nem sempre uma variável representada por números é quantitativa.  
O número do telefone de uma pessoa, o número da casa, o número de sua identidade. Às vezes o sexo do indivíduo é registrado na planilha de dados como 1 se macho e 2 se fêmea, por exemplo. Isto não significa que a variável sexo passou a ser quantitativa!

Exemplo do ursos marrons (continuação):  
No conjunto de dados ursos marrons, são qualitativas as variáveis sexo (nominal) e mês da observação (ordinal); são quantitativas contínuas as demais: idade, comprimento da cabeça, largura da cabeça, perímetro do pescoço, perímetro do tórax, altura e peso.

**Os Tipos de Tarefas dos Algoritmos:**

**Tarefa Supervisionada:** Alguém te passa os atributos e dizem que esses atributos são de uma determinada coisa. Ex: Passam-te o numero de rodas, as dimensões, o numero de assentos e falam que esses atributos pertencem a um carro. Logoquando se depara com esses atributos vc saberá q e um carro.

**Modelo de Classificação:** Classifica a variável resposta em classes. Ex: Sim ou não; Carro, moto ou ônibus; Verde, vermelho, azul; etc... **O que e medido na classificação e se o valor predito e sim ou não, certo ou errado. Para representar isso se usa a acurácia, que mede a quantidade de acertos da predição em relação ao valor real.**

**Modelo de Regressão:** Aqui e uma variável real, vai de – infinito ate infinito. E variável resposta numérica continua. EX: preço, altura, temperatura, etc.. **O que e medido na regressão e o quão longe a sua predição esta do valor real. Para representar isso se usa as métricas de erro.**

**Series Temporais:** Modelos de ML preparados para regressão ao longo do tempo e chamado de **time series.** Modelos de predição (previsão) preveem os valores do futuro baseados em valores do passado.

## M.7. - Vídeo 2: Machine Learning Nao-Supervisionado:

**Tarefa Não-Supervisionada:** Vc não sabe que coisas são, pois ninguém te disse o que e, e nem disseram quais atributos são relacionados a essa coisa.

**Agrupamento/Clusterizacao:** Aqui agrupa- se essas coisas, em grupos chamados clusters, por similaridade de tamanho, cor, formato e outras características.

## M.7. - Vídeo 3: Machine Learning Reinforcement Learning (Semi-Supervisionados):

**Funcionamento do Reinforcement Learning:** Aprende durante a execução.

Ação & Recompensa;

Agente & Ambeinte.

**Agente** faz uma **ação** no **ambiente**, e o **ambiente** responde o **agente** com uma **recompensa**, podendo ser positiva ou negativa.

Este ciclo de aprendizado se repete ate o agente conseguir receber somente recompensas positivas em todo o percurso.

Exemplo de Como funciona:

NoNetflix, o pessoal de marketing cria varias capas diferentes para uma determinada serie, com o intuito de chamar a atenção do usuário pra ver essa serie.

Após criar essas capas, o time de data analysis quer saber qual das capas funciona melhor, ou seja, chama mais atenção do usuário.

Para isso, usa-se o RL.

Primeiro passo:

O agente e criado, e ele tem duas fases: fase 1 - exploration e fase 2 - exploitation.

Fase 1 – Exploration: Pega qualquer uma das capas aleatoriamente, mostra para o usuário e coleta feedback.

Esta coleta de feedback da-se pela seguinte maneira: se o usuário clicar nesta capa aleatória, esta capa recebe uma recompensa positiva + 1, se o usuário não clicar na capa, a mesma recbe recompensa negativa 0.

Fase 2 – Exploitation: depois de um tempo, passa-se para este tipo de exploração, porem baseado em um aprendizado passado. Onde já se sabe quais capas q não chamam à atenção dos usuários, e o agente as exclui.

## M.7. - Vídeo 4: Os 5 Melhores Modelos de Machine Learning:

1. **Modelo de Media (Average Model):** Serve para ser uma base comparativa para performance dos outro modelos. Ex: se o modelo de media, que e inferior oas outros modelos, ter uma acurácia de 80%, e um deep learning ter uma acurácia de 60%, significa que a acurácia do deep learning não e boa.
2. **Regressão Linear:** Usa-se este algoritmo, que são mais simples. Se este modelo tiver uma boa performance, assume-se que o dataset tem um comportamento linear. Entao pode-se utilizar modelos mais simples que trazem boas performances para poder usar como modelo final. Caso contrario, significa que o dataset e complexo, então precisa-se de algoritmos não lineares como random forest, SVM e redes neurais.
3. **Regressão Linear Regularizado:** Tem um parâmetro a mais do que a regressão linear, geralmente chamado de Lasso e Ridge. E geralmente performa um pouco melhor que o linear.
4. **Random Forest Regressor:** Modelo baseado em arvore, adaptado para regressões.
5. **XGBoost Regressor:** Modelo baseado em arvore, adaptado para regressões. Utiliza um método chamado begging e boosting, para selecionar as featrures mais rapidamente e mais relevantes. Geralmente com performance melhor.

## M.7. - Vídeo 6: O Método de Cross - Validation:

Metodo utilizado para dividir o dataset em treino e teste. Dividi-se o dataset para treinar o algoritmo de ML, onde a parte de treino do dataset e onde o algoritmo aprende o dataset e a parte de teste e onde o algoritmo testa o que aprendeu.

Existe diversos métodos de dividir o dataset, como por exemplo, dividir o dataset entre 70 % para treino e 30% para teste, ou no caso desse projeto, que pegou as ultimas 6 semanas como teste e as semanas anteriores como treino. Porem esse métodos de divisão são ruins, os dados das ultimas 6 semanas podem ser tendenciosos para bom ou para ruim.

Por isso o melhor e mais utilizado e o cross validation.

**Cross - Validation tem duas formas de aplicação:**

1. Quando se faz uma atividade de regressão ou classificação.
2. Quando se faz uma atividade de time series ou series temporais.

**Como funciona o Cross – Validation para atividade de regressão e classificacao:**

Divide-se o dataset em diversas partes, por exemplo 10 partes, numeradas de 1 a 10.

Usa-se a parte 1 para teste, as outras 9 para treino e calcula a performance;

Depois usa-se a parte 2 para teste, as outras 9 para treino e calcula a performance;

Depois usa-se a parte 3 para teste, as outras 9 para treino e calcula a performance;

Faz-se isso ate chegar a decima parte.

Depois de treinar e testar o algoritmo 10 vezes, faz a media dos 10 valores de performances do algoritmo.

A performance real do modelo e o resultado dessa media mais ou menos o desvio padrão das 10 partes.

OBS: Aqui se chama a parte de teste como validação.

**Como funciona o Cross – Validation para atividade de time series:**

A única diferença desse tipo para com o anterior e que se te que respeitar a ordem cronológica dos dados.

Por exemplo: No primeiro, dividi-se os dados em 10 partes, então pode-se pegar qualquer parte desses 10 para colocar como teste, pode –se por exemplo colocar a parte 3 do dataset como teste como primeira validação.

Já no time series a ordem deve ser respeitada. Por exemplo, se dividiu os dados em 10 partes, então a parte 1 tem que ser a primeira validação, a parte 2 a segunda validação, a parte 3 a terceira validação, e assim sucessivamente.

Porem tem um adendo. Aqui faz –se o treinamento da parte 1, somente com a parte 1 e depois valida, depois pega-se a parte 1 e 2 e depois valida, depois pega-se a parte 1, 2 e 3 e depois valida, assim sucessivamente.

Geralmente a validação tem o mesmo tamanho.

EX: No caso desde problema, a validação será de 6 semanas, e nas parte de treino dividira o restante das semanas em um úmero k de partes iguais.

# MODULO 08: HYPERPARAMETER FINE TUNING

## M.8. - Vídeo 1: As 3 Estrategias para Fine Tuning - Parte 1:

**Update do Progresso do Projeto Rossmann:**

Nesta etapa já se sabe a **questão de negocio (Modulo 1)**, o **entendimento do negócio (Modulo 1)**, já foi feito a **coleta dos dados (kaggle),** a **limpeza dos dados(Módulos 2 e 3)** , **Exploração dos dados(Modulo 4)**,  **Modelagem dos dados (Módulos 5 e 6) e Algoritmos de Machine Learning (Modulo 7).**.

Agora começaremos a **Algoritmos de Machine Learning (Modulo 8).**

**A Motivação do Fine Tuning:**

“Encontrar o conjunto de parâmetros que maximiza o aprendizado do modelo.”

**As 3 Estrategias para Fine Tuning:**

1. **Random Search:** Define valores para cada um dos hiperparametros aleatoriamente. Por exemplo: E programado pra rodar 10 vezes, e em cada vez e escolhido valores aleatórios para cada hiperparametro. No final o conjunto com melhor performance e escolhido.

-Vantagens: Fácil de implementar e super-rápido de rodar e baixo custo.

- Desvantagens: Pode usar o mesmo conjunto de valores mais de uma vez; Pode nunca conseguir encontrar o melhor conjunto de valores que maximiza o aprendizado do modelo. Pode encontrar somente os máximos locais e não os gerais.

1. **Grid Search:** Define todas as combinações possíveis de valores que os hiperparametros podem assumir.

-Vantagens: Realmente encontra o melhor conjunto de valores que maximiza o aprendizado do modelo.

- Desvantagens: Demora muito pra rodar.

1. **Bayesian Search:** Define os valores para os hiperparametros seguindo a teoria de Bayes. Tenta encontrar o melhor conjunto baseado no aprendizado anterior.

-Vantagens: Define os valores para os hiperparamentros baseado em aprendizados passados; mais rápido que o grid e mais lento que o random.

- Desvantagens: Complexo de entender.

**Aplicação do random search no pyhton:**

Feito no jupyter notebook.

# MODULO 09: INTERPRETACAO E TRADUCAO DO ERRO

## M.9. - Vídeo 1: Interpretação dos Erros:

**Update do Progresso do Projeto Rossmann:**

Nesta etapa já se sabe a **questão de negocio (Modulo 1)**, o **entendimento do negócio (Modulo 1)**, já foi feito a **coleta dos dados (kaggle),** a **limpeza dos dados(Módulos 2 e 3)** , **Exploração dos dados(Modulo 4)**,  **Modelagem dos dados (Módulos 5 e 6) e Algoritmos de Machine Learning (Modulo 7 e 8).**

Agora começaremos a **Avaliação do** **Algoritmo (Modulo 9).**

**A Motivação da Traducao do Erro:**

“Como entender a performance do modelo e dizer para o CEO o quanto $ esse modelo vai trazer para a empresa?”

**Performance do Modelo de Regressão:**

1. MAE (Mean Absolute Error): diminui o valor real pelo valor predito, soma o modulo de tudo ( ignora o sinal de menos) e divide pelo numero de pontos existentes, gerando a media de todos esses valores.

- Atribui peso igual para todos os erros.

- Robusto na presença de outliers.

- Facil entendimento pelo time de negócios.

- Bom para usa-lo como report para o time de negócios.

1. MAPE ( Mean Absolute Percentage Error): diminui o valor real pelo valor predito e divide esse valor pelo valor real, soma o modulo de tudo ( ignora o sinal de menos) e divide pelo numero de pontos existentes, gerando a media de todos esses valores, e no final multiplica por 100 para ter o valor em percentual.

- esse valor diz o quanto se esta errado percentualmente, informa o quão longe se esta do valor real, na media, em porcentagem.

- Muito usado na divulgação de resultados.

- Não pode ser usado se a variável resposta contem 0.

1. RMSE (Root Mean Squared Error): diminui o valor real pelo valor predito, eleva o resultado ao quadrado, soma todos os valores e divide pelo numero de pontos existentes, gerando a media de todos esses valores, no final tira a raiz quadrada dessa media.

- Atribui MAIOR PESO em erros MAIORES.

- Sensível na presença de outliers.

- Ideal para medir a performance de modelos de ML.

- usa-se o RMSE para medir a performance do modelo de ML, por causa dos outliers, e para reportar para o time de negócios, usa-se o MAE e o MAPE, pois não consideram muito o problema do outlier.

1. MPE (Mean Percentage Error): Seu objetivo e verificar se o modelo esta subestimando ou superestimando as predições. Diminui o valor real pelo valor predito, soma os valores e divide pelo numero de pontos existentes, gerando a media de todos esses valores e no final multiplica por 100 para ter o valor em percentual. Se esse valor for negativo significa que o modelo esta superestimando as previsões, se o valor for positivo significa que o modelo esta subestimando as predições.

- Não pode ser usado para medir performance do modelo.

- Útil para entender se o modelo esta subestimando ou superestimando.

- Não pode ser usado se a variável resposta contem 0.

**Como usar os Erros na Performance do Modelo de Regressão:**

RMSE e MPE são métricas usadas para melhoria de performance do modelo.

**Como usar os Erros na Performance do Modelo sobre o Negocio:**

MAE, MAPE e MPE e MPE são métricas usadas para report para o time de negocio.

**Performance do Modelo sobre o Negocio:**

1. Comparação com o Status quo: Compara modelos já existente na empresa com o novo modelo de ML implementado pelo cientista de dados. Fazendo a diferença entre o Mae e o MAPE do modelo atual e o MAE e MAPE do modelo anterior.
2. Incremento de Receita: Feito no jupyter notebook.

# MODULO 10: DEPLY DO MODELO EM PRODUCAO

## M.10. - Vídeo 1: A Arquitetura do Modelo em Produção:

**Update do Progresso do Projeto Rossmann:**

Nesta etapa já se sabe:

1. **Questão de negocio (Modulo 1): previsão de vendas nas próximas 6 semanas**;
2. **Entendimento do negócio (Modulo 1)**: requisição vinda do CFO, pois ele precisa determinar um valor de budget para cada uma das lojas com o intuito de fazer reforma em cada uma delas. Porem ele não sabe o valor exato que ele precisa colocar nesse budget. Então ele quer saber uma previsão de vendas para daqui a 6 semanas para cada uma das lojas, para saber o quanto de receita cada loja terá, então quanto ele separara para esta reforma.
3. **Coleta dos dados (kaggle):** Download do dataset no site do kaggle.
4. **Limpeza dos dados (Módulos 2 e 3):** Faz-se a descrição dos dados, ondese tem um resumo geral dos dados e se sabe o tamanho do problema. Também se cria novas variáveis a partir das variáveis originais, chamado de feature engineering. Por ultimo se faz a filtragem das variáveis baseado no negocio.
5. **Exploração dos dados (Modulo 4)**: Faz-se a EDA, onde se tem 2 objetivos, primeiro entender o negocio do ponto de vista dos dados, segundo encontrar ou ter um sentimento de quais variáveis são relevantes para o modelo de ML. Foram feitas analises univariadas, bivariadas e multivariadas, e também foram levantadas hipóteses. Foi verificado quais hipóteses foram validadas. Insights gerados.
6. **Modelagem dos dados (Módulos 5 e 6):** Preparação dos dados, onde se transforma variáveis categóricas em numéricas, se rescala a variáveis, e transformação de natureza e da variável resposta. E também se faz uma filtragem do dados pelo boruta com o intuito de verificar quais variáveis são relevantes para o modelo de ML. No final se comparou e fez um merge das variáveis relevantes encontradas no boruta e na EDA.
7. **Algoritmos de Machine Learning (Modulo 7 e 8):** Implementação dos algoritmos e verificação de qual algoritmo e o melhor para este problema através do cross validation, que faz a medição real da performance do algoritmo.
8. **Avaliação do** **Algoritmo (Modulo 9):** Analise as métricas de erro que mede a performance do modelo. Também informa quais métricas são usadas para medir a performance do modelo, e quais usadas para passar para o time de negócios. Por ultimo foi explicado como informar os resultados do modelo para o time de negócios e para o CFO, ou seja, quanto de dinheiro as lojas venderão daqui a 6 semanas.
9. A partir daqui se decide se vamos para um segundo ciclo, alterando algumas coisas para tentar melhorar a performance do modelo de ML. Apesar de ter algumas coisas a se fazer como: reduzir a performance de algumas lojas pois estão muito alta; adicionar a variável ano no modelo de ML; excluir variáveis preditoras que tem correlação acima de 0.6; resolver a hipótese h5, etc... Porem para finalizar o curso iremos direto para o **modelo em produção.**

Agora começaremos o **Modelo em Produção (Modulo 10).**

E bom colocar o modelo em produção no primeiro ciclo para pegar feedback dos resultados e como esta a usabilidade do modelo na plataforma online.

Esta etapa e feita pelo **Machine Learning Engineer**, porem e bom um data scientist saber também.

**A Motivação do Modelo em Produção:**

“Tornar as predições do modelo acessível para qualquer consumidor.”

**Termos mais Comuns:**

1. **API** = Application Programming Interface.
2. **Request** = Requisicao / Chamada.
3. **Endpoint** = URL / Porta da API, endereço do website.
4. **Deploy** = Implementacao / Publicacao. Tira o projeto do ambiente local e coloca no ambiente de nuvem que acessado por todos.
5. **Ambiente Local** = Seu computador.
6. **Ambiente de Produção** = Servidores na nuvem que fornecem um serviço / dados. Ou seja, um site de loja online, onde o cliente navega clicando em varias coisas e comprando.
7. **Ambiente de Desenvolvimento** = Uma copia idêntica do ambiente de produção na nuvem controlado e não disponível online, com o intuito de se deixar fazer melhorias nesse mesmo ambiente, que caso essa melhorias dão certo, são copiadas para o ambiente de produção.

**Ciclo de Desenvolvimento:**

2º Teste no Ambiente de Desenvolvimento

1º Desenvolvimento Local 3º Publicação no Ambiente de Produção

**O que e uma API?**

Assistir de novo pq não entendi nada.

**Lista de Tarefas:**

1. Criar uma Classe com as Limpeza, Transformações e Encoding. Ira se chamar Rossmann.py.
2. Criar a API. Ira se chamar Handler.py.
3. Criar um script para testar o API.

## M.10. - Vídeo 2: A Criação da Classe em Produção:

**Implementando a Classe Rossmann:**

Criar uma Classe com as Limpeza, Transformações e Encoding. Ira se chamar Rossmann.py.

Feito o Jupyter notebook.

**# funcionamento da api:**

# carrega o modelo em memoria com o endpoint ativo, quando recebe a requisicao ela pega os dados originais, prepara esses dados,

# passa para o modelo, o modelo faz a predisao, depois de feito a predicao, a mesma e devolvida e anexada aos dados originais

# e devolve pra quem pediu, ou seja, e enviado para o endpoint.

**Funcionamento de todo o processo do modelo em producao:**

# aki se faz o modelo, trabalha nas transformacoes do dataset, faz o eda, e tudo mais, depois coloca o modelo salvo em producao,

# e por fim se faz uma requisicao de fora via api, o modelo recebe essa requisicao, roda, fornece a predicao e devolve via api.

No API tester o dado original é as vezes tratado e sempre transformado em json. Depois de transformado em json é passado para o API Handler, juntamente com o seu tipo de dado, que é json.

No API Handler o dado original, que esta em json, é transformado em dataframe, tratado, também é feito a predição e por ultimo transformado em json novamente (todo esse processo é através de funções criadas na classe rossmann). Após todo esse processo, o dado (em json) tratado e predito, é enviado para o endpoint, cujo qual é plotado pelo API Tester. Este processo é todo feito em forma de ciclo, onde começa e termina no API Tester.

## M.10. - Vídeo 5: Deploy do Modelo ML no Heroku:

1. **Servidor Gratuito na Nuvem.**

**Tarefas para Deploy no Heroku:**

1. Criar uma conta gratuita no heroku
2. Instalar o cliente do heroku (mac, Linux, Windows)
3. Logar na conta do heroku via terminal (heroku login)
4. Criar 3 arquivos:

* Procfile
* Requirements.txt
* git init

# MODULO 11: CRIANDO O BOT NO TELEGRAM

## M.10. - Vídeo 1: A Arquitetura do Telegram Bot:

**A Motivação do Telegram Bot:**

“Impressionar o Recrutador com um Produto de Dados Real. Construído com Todos os Passos um Projeto de Data Science..”