Bom Negócio - Anúncio de Vendas de Carros

Karen Yasmin de Oliveira Vicente

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Pós Graduação em Ciência de Dados e Big Data

PUC Minas

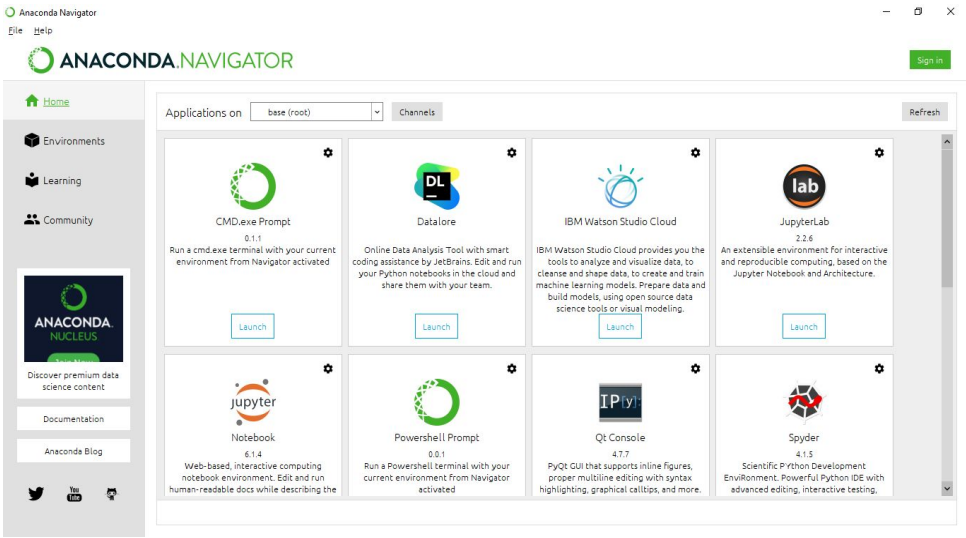
Definição do Problema

- Auxiliar e identificar os melhores negócios para os usuários
- Para clientes e futuros compradores
- Classificar os anúncios de venda de carro como um Bom Negócio.
- Realizar uma Análise Exploratória e Modelagem Preditiva para explorar e classificar os anúncios.
- Exibir em anúncios de lojas de carros

Ferramentas Utilizadas

Python

- Anaconda Navigator
- Juptyer Notebook



PUC Minas

Coleta de Dados

PUC Minas

Coleta de Dados

API

- Site: webmotors.com.br

Datasets:

- Anúncios
- Valores da Tabela Fipe

PUC Minas

Coleta de Dados

script.py

```
#coding: utf-8

import json

from urllib.request import urlopen # Faz a requisição no servidor e
obtem a resposta
import urllib.error

# Definindo variáveis da url de busca
urlBase = 'https://www.webmotors.com.br/api/';
urlDetails = urlBase + 'detail/car/'

print('Obtendo os dados, aguarde!');

# Pegando dados
data = [];
for i in range(1, 500):
    url = urlBase +
'search/car?url=https://www.webmotors.com.br/carros%2Fsp%3Festadoci
dade%3DS%25C3%25A3o%2520Paulo%26tipoveiculo%3Dcarros&actualPage='+s
tr(i)

    # Exibir erro caso tenha problemas para obter os dados
    try:
        data += json.load(urlopen(url))['SearchResults'];
        print('Dados da Page: %s obtidos com sucesso, aguarde a
criação do arquivo!' % i)
```

PUC Minas

Coleta de Dados

Dataset: Anúncios


```
[
  {
    "UniqueId": 34283122,
    "Specification": {
      "Title": "LAND ROVER DISCOVERY SPORT 2.0 16V TD4 TURBO DIESEL HSE 4P AUTOMÁTICO",
      "Make": {
        "id": 23,
        "Value": "LAND ROVER"
      },
      "Model": {
        "id": 3516,
        "Value": "DISCOVERY SPORT"
      },
      "Version": {
        "id": 346910,
        "Value": "2.0 16V TD4 TURBO DIESEL HSE 4P AUTOMÁTICO"
      },
      "YearFabrication": "2018",
      "YearModel": 2018,
      "Odometer": 23000,
      "Transmission": "Automática",
      "NumberPorts": "4",
      "BodyType": "Utilitário esportivo",
      "VehicleAttributes": [
        {
          "Name": "Aceita troca"
        },
        {
          "Name": "Todas as revisões feitas pela agenda do carro"
        },
        {
          "Name": "Único dono"
        },
```


Dataset: Tabela Fipe


```
][
]     {
          "Fipe": 206373.0,
          "UniqueId": 34283122
      },
]     {
          "Fipe": 31520.0,
          "UniqueId": 9532158
      },
```


PUC Minas

Processamento/ tratamento dos dados

PUC Minas

Processamento/tratamento dos dados

Bibliotecas Utilizadas

```
import json
import pandas as pd # lib pandas
import numpy as np # lib numpy
import datetime # lib datetime

import matplotlib.pyplot as plt # lib para utilização dos gráficos
import seaborn as sns # lib para exibir dados estátisticos

# Necessário para visualização automática dos gráficos no Jupyter
%matplotlib inline

from pathlib import Path
from urllib.request import urlopen # Faz a requisição no servidor e obtem a resposta
from pandas import json_normalize # package for flattening json in pandas df

# lib sklearn
from sklearn.model_selection import train_test_split # lib para definir os dados de treino e de teste
from sklearn.neighbors import KNeighborsClassifier # lib classe KNeighborsClassifier - Classificação
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn import preprocessing # Import LabelEncoder
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import RandomForestClassifier # lib classe RandomForestClassifier - Classificação
```

Processamento/tratamento dos dados

Unindo os Datasets

```
#Unindo os datasets infos detalhes do carro
df_cars = df_cars.join(df_cars_fipe.set_index('UniqueId')[['Fipe']], on='UniqueId')
df_cars.head()
```

PUC Minas

Processamento/tratamento dos dados

Removendo colunas irrelevantes

```
del df_cars['Channels'] # Coluna Channels - Canal de origem da
venda
del df_cars['ListingType'] # Coluna Tipo de Listagem
del df_cars['ProductCode'] # Coluna Código Produto
del df_cars['PhotoPath'] # Coluna PhotoPath
del df_cars['FipePercent'] # Coluna FipePercent
del df_cars['LongComment'] # Coluna LongComment - comentários do
vendedor

del df_cars['VipAutopago'] # Coluna VipAutopago

# Group - Media
del df_cars['Media.Photos'] # Coluna Media.Photos
del df_cars['Media.Videos'] # Coluna Media.Videos
```

Processamento/tratamento dos dados

Renomeando Colunas

```
df_cars.rename({'Specification.Make.Value': 'Make',
                'Specification.Model.Value': 'Model',
                'Specification.Version.Value': 'Version',
                'Specification.YearModel': 'YearModel',
                'Specification.YearFabrication': 'YearFabrication',
                'Specification.Odometer': 'Odometer',
                'Specification.Armored': 'Armored',
                'Prices.Price': 'Price',
                'Prices.OldPrice': 'OldPrice',
                'Specification.Color.Primary': 'Color',
                'Specification.Armored': 'Armored',
               }, axis=1, inplace=True)
```

Processamento/tratamento dos dados

Padronizando Tipos das Colunas

```
# Convertendo o ano de Float64 para Int64 (obs: o ano modelo está
vindo como float 1 casa decimal)
df_cars['YearModel'] = df_cars['YearModel'].astype('int64')

# Convertendo dados que estão como object para string
df_cars['Make'] = df_cars['Make'].astype('string')
df_cars['Model'] = df_cars['Model'].astype('string')
df_cars['Armored'] = df_cars['Armored'].astype('string')
```

Análise/exploração dos dados

PUC Minas

Análise/exploração dos dados

Análise coluna *GoodDeal (Bom Negócio)*

	Bom Negócio	Quantidade	Porcentagem
0	Pode não ser um Bom Negócio	10466	93.371398
1	Bom Negócio	743	6.628602

PUC Minas

Análise/exploração dos dados

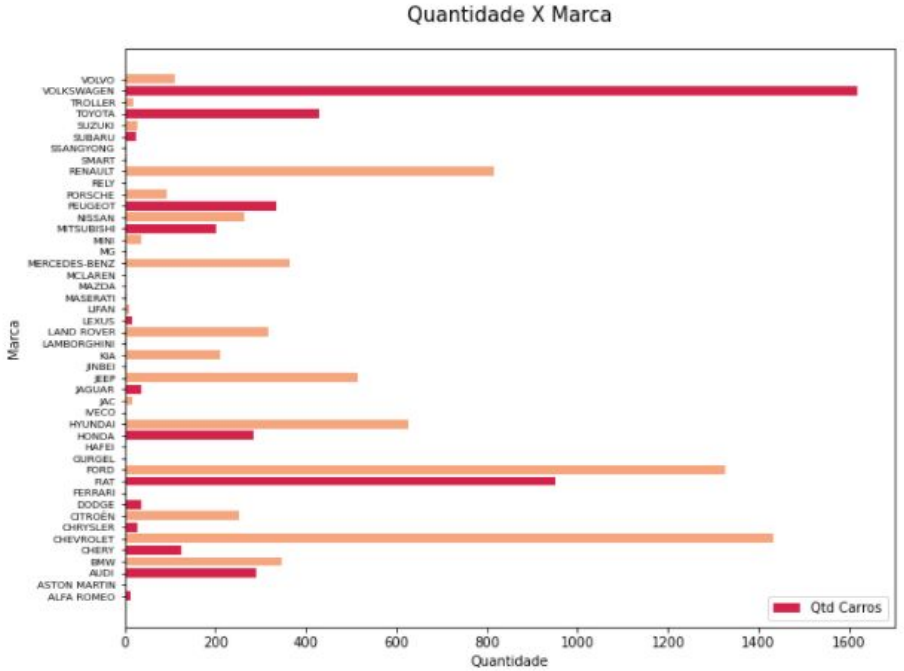
Análise coluna *GoodDeal (Bom Negócio)*

- **Relação com a coluna *PriceFipeOk:*** verificamos que foram encontrados 743 registros como GoodDeal, onde todos os registros estão com o Preço Fipe Ok.
- **Relação com a coluna *OdometerRecommended (<15km)*:** foram encontrados 281 registros com a quilometragem recomendada recebendo um valor verdadeiro e considerados um Bom Negócio.
- **Relação com a coluna *OldPrice*:** não foi encontrado nenhum registro onde o preço inicial do anúncio tenha sido ajustado por um menor preço.

PUC Minas

Análise/exploração dos dados

Gráfico Agrupamento: Marca e Quantidade

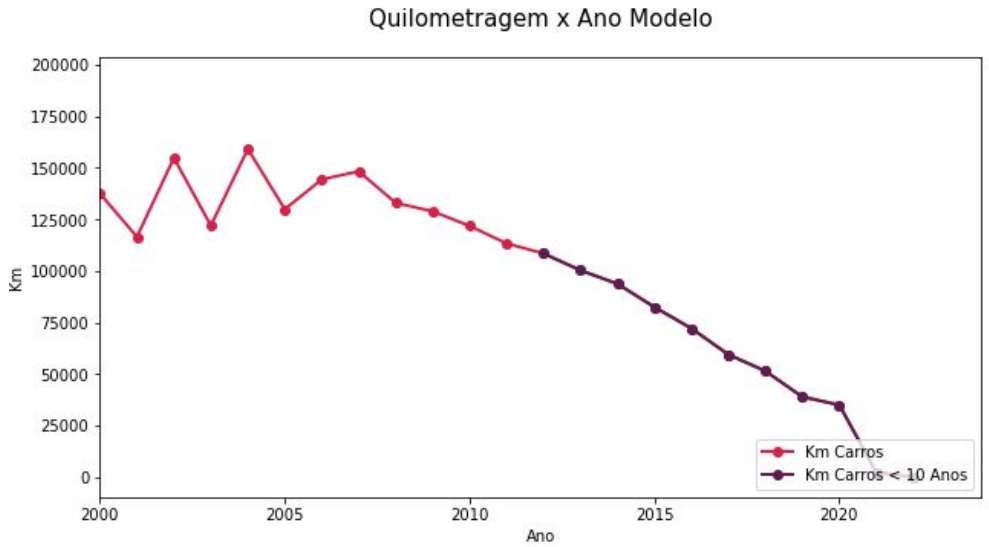


PUC Minas

Análise/exploração dos dados

Gráfico Agrupamento

quilometragem dos carros



PUC Minas

Análise/exploração dos dados

Gráfico Agrupamento valor Fipe OK e Fipe não OK das 10 maiores marcas

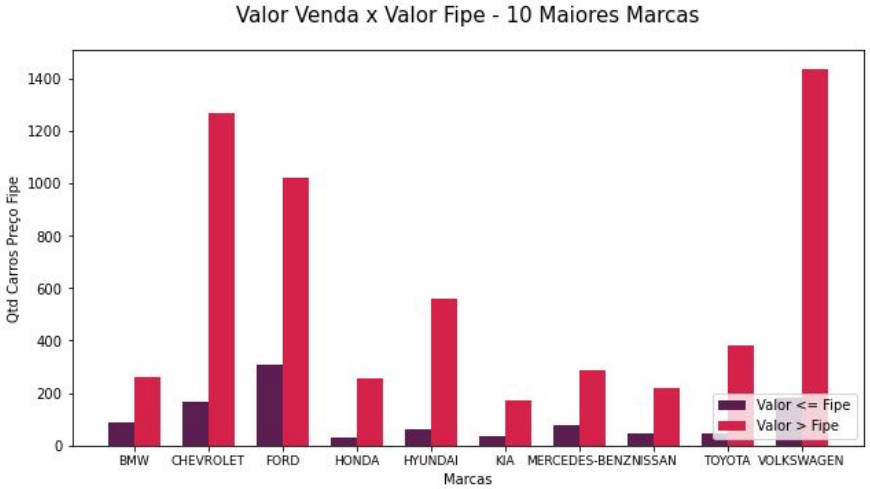
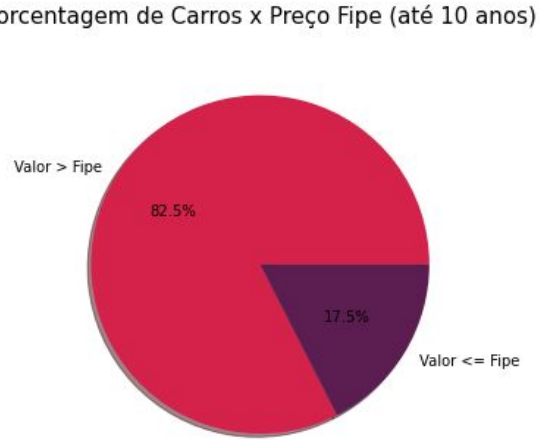


Gráfico Análise valor Fipe OK e Fipe não OK em porcentagem



PUC Minas

Criação de Modelo ML

PUC Minas

Criação de Modelo ML

Definindo variáveis: X e Y

```
# Selecionando todas as colunas exceto a coluna “Class”
X = df_cars_class.drop('Class', axis=1)

# Selecionando apenas valores da coluna Class
y = df_cars_class['Class']
```

Dividindo a base de treino e de teste

```
# Divide o DataFrame em teste e treino
# 70% treino
# 30% teste
train_X, test_X, train_y, test_y = train_test_split(X, y,
train_size=0.70, test_size=0.30, stratify=y)
```

PUC Minas

Criação de Modelo ML

KNN (K-Nearest Neighbor)

```
Confusion Matrix:
[[3103   37]
 [ 139   84]]
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.99      0.97      3140
           1       0.69      0.38      0.49       223

    accuracy                           0.95      3363
   macro avg       0.83      0.68      0.73      3363
weighted avg       0.94      0.95      0.94      3363

Accuracy: 0.9476657746060065
```

PUC Minas

Criação de Modelo ML

Random Forest Classifier

```
Confusion Matrix:
[[3100   40]
 [ 101  122]]
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.99      0.98      3140
           1       0.75      0.55      0.63       223

    accuracy                           0.96      3363
   macro avg       0.86      0.77      0.81      3363
weighted avg       0.95      0.96      0.95      3363

Accuracy: 0.9580731489741302
```

PUC Minas

Interpretação dos Resultados

PUC Minas

Interpretação dos Resultados

Gráfico Resultados KNeighborsClassifier

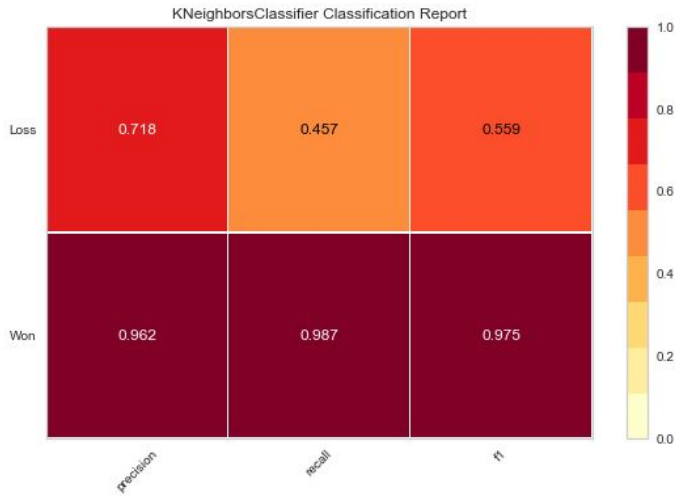
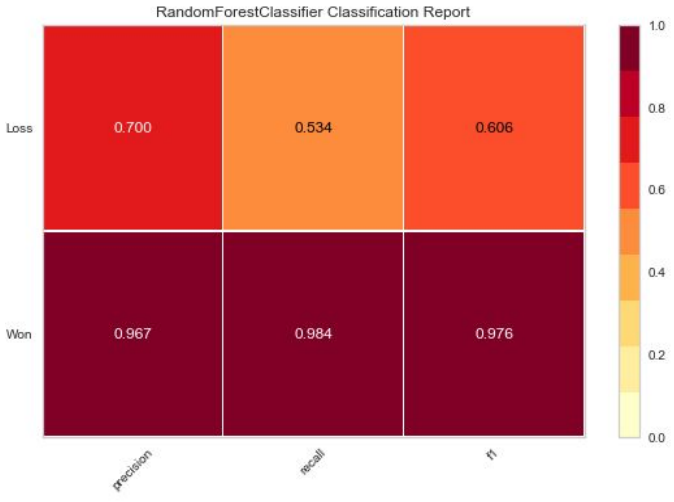
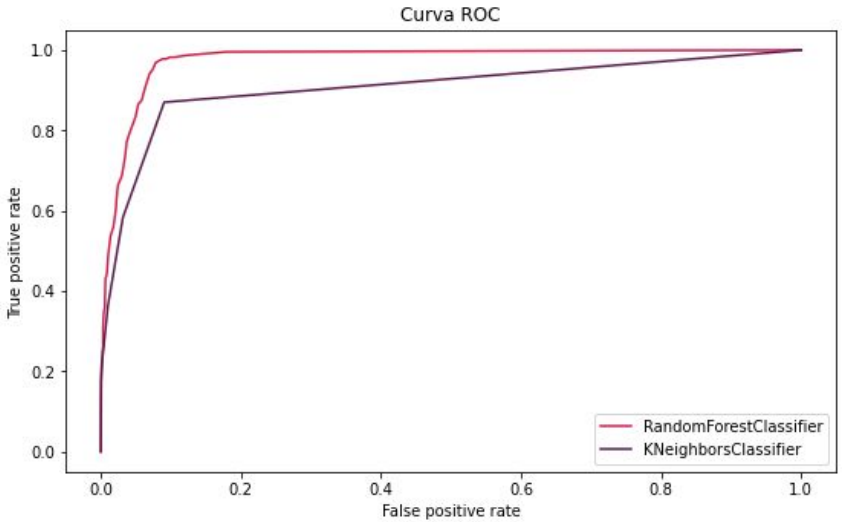


Gráfico Resultados RandomForestClassifier



PUC Minas

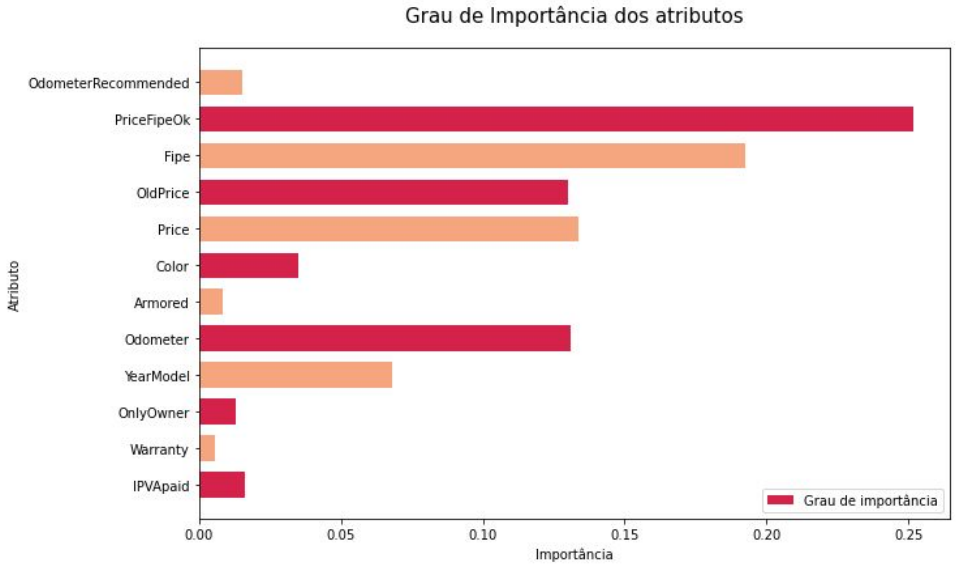
Interpretação dos Resultados



PUC Minas

Comunicação dos Resultados

Comunicação dos Resultados



PUC Minas

Comunicação dos Resultados

Título: Bom Negócio - Anúncio de vendas de carros		
Problema Analisar o dataset de anúncios de vendas de carros da Webmotors e investigar atributos relacionados ao atributo "Bom Negócio."	**Resultados e Previsões** Avaliar os atributos relacionados a atribuição verdadeira do atributo "Bom Negócio", com a finalidade de tentar prever e classificar os atributos de maior importância e assim atribuir verdadeiro ou falso para o atributo "Bom Negócio"	**Aquisição de Dados** Os dados de ambos os datasets data-cars.json e data-cars-fipe.json) foram coletados do site da Webmotors.
Modelagem Realizado análises no dataset coletado, tanto de forma gráfica quanto análise descritiva dos dados utilizando a biblioteca *Pandas* em *Pyrthon*. Desta forma foi possível identificar um dataset adequado para aplicar modelo de classificação de ML.	**Avaliação do Modelo** Para avaliação dos resultados obtidos no modelo de classificação, foram avaliados a Matriz de Confusão e o Relatório de Classificação conforme o notebook em Python no diretório deste projeto.	**Preparação dos Dados** Após a união dos datasets, os dados foram tratados, as colunas foram renomeadas, os dados duplicados foram removidos e dados desnecessários para a análise também foram removidos.

PUC Minas

Links

- Link para o vídeo: https://drive.google.com/file/d/11WKz7NBY41jevE6eWWklSHIDl62sa6TK/view?usp=sharing
- Link Github: https://github.com/karenyov/TCC_PUC_BigData.

PUC Minas

Obrigada!