

TECNOLOGIA EM SISTEMAS PARA INTERNET

**André Luiz Gomes dos Santos
Brenda Lopes Miranda Teixeira**

**RELATÓRIO DE PRÁTICA INTEGRADA
DE
CIÊNCIA DE DADOS E APRENDIZADO DE MÁQUINA**

Brasília - DF

05/03/2020

Sumário

1. Objetivos	3
2. Descrição do problema	3
3. Desenvolvimento	3
3.1 Código implementado	3
3.1.1 Base de dados	4
4. Considerações Finais	8
Referências	10

1. Objetivos

Na última fase do projeto foi solicitado aos grupos realizar um modelo para estimar a quantidade de visualizações que obteremos no nos anos posteriores. Tendo em vista alguns objetivos listados:

- Recuperação de dados;
- Ordenar as observações de forma ascendente temporalmente;
- Construção dos conjuntos de Treinamento e Teste;
- Investigar os parâmetros para discriminar o melhor modelo;
- A última etapa é realizar uma previsão utilizando o melhor modelo.

2. Descrição do problema

Depois de ter conhecido a The National UFO Reporting Center e ter passado por toda a trajetória com as sprints, chega-se na reta final. Agora é hora de procurar uma solução para aperfeiçoar a quantidade de visualizações para os próximos anos.

3. Desenvolvimento

Este trabalho está sendo desenvolvido usando um Script Python por ser uma linguagem orientada a objetos é bastante maleável, o grupo G2 está utilizando a plataforma Google Colaboratory, assim todos podem modificar e acrescentar o código quando necessário. Todos os códigos estão sendo disponibilizados no github.

Link para o github:

<https://github.com/Prof-Fabio-Henrique/pratica-integrada-icd-e-iam-2020-2-grupo-2>

3.1 Código implementado

Abaixo será mostrada imagens do código já implementado com descrições.

Como sempre, começamos importando as bibliotecas.

Imagem 1 - Importação

Importando as bibliotecas

```
[ ] #!pip install -U pandasql
import pandas as pd
import numpy as np
import pandasql
import matplotlib
import matplotlib.pyplot as plt
import datetime
from datetime import datetime
```

Fonte: Própria

Foi realizada a Leitura do "df_OVNI_preparado.csv" e criando o dataframe.

Imagem 2- Leitura CSV

```
#Leitura CSV
ovnis = pd.read_csv('df_OVNI_preparado.csv')

ovnis
```

Fonte: Própria

Nesse tópico Converteremos a coluna em data.

Imagem 3- Coluna/ Data

```
#Converter a coluna em data
ovnis['Date'] = pd.to_datetime(ovnis['Sight_Date'])
ovnis
```

Fonte: Própria

Ordenar as observações de forma ascendente temporalmente (da observação mais antiga para a observação mais recente).

Imagem 4- Ordenar datas

```
#Ordenar datas
dados_phoenix = ovnis[ovnis['City']=='Phoenix']
dados_phoenix.sort_values(by=['Date'], inplace=True, ascending=True)
dados_phoenix
#print(type(dados_phoenix))
```

Fonte: Própria

Aqui é realizado um filtro de dados.

Imagem 5 - Filtro

```
q = """
SELECT Date, Count(*) as Views
FROM dados_phoenix
group by Sight_Day, Sight_Month
order by Date
"""

df1 = pandasql.sqldf(q, locals())
df1
```

Fonte: Própria

Visualização dos dados em forma de Série Temporal.

Imagem 6 - Filtro

```
#Observar o gráfico em barras da série temporal para o ano x de forma a
#investigar como se comporta a distribuição das visualizações

#Análise do ano: 2001

#Carregar o seu arquivo OVNIS.csv em um dataframe;
temporal = pd.read_csv('df_OVNI_preparado.csv')
#temporal

#Filtro de dados
q = """
SELECT City, Sight_Month, COUNT(*) AS total
FROM temporal
WHERE City = 'Phoenix'
GROUP BY Sight_Month
"""

df2 = pandasql.sqldf(q, locals())
df2
```

Fonte: Própria

Chegamos na parte do gráfico em barras

Imagem 7- Barras

```
#gráfico em barras
valoresA = df2['Sight_Month']
valoresB = df2['total']

#a = ['Filme 0', 'Filme 1','Filme 2', 'Filme 3','Filme 4', 'Filme 5']
# Plota as barras
plt.bar(valoresA, valoresB, label = valoresA)

#Título
plt.title("Ano 2001", fontdict={'fontsize':20})

#dimensões do grafico
plt.rcParams['figure.figsize'] = (10,10)

#Grade
plt.grid(axis='y')
```

Fonte: Própria

Imagem 8 - Editar nomes

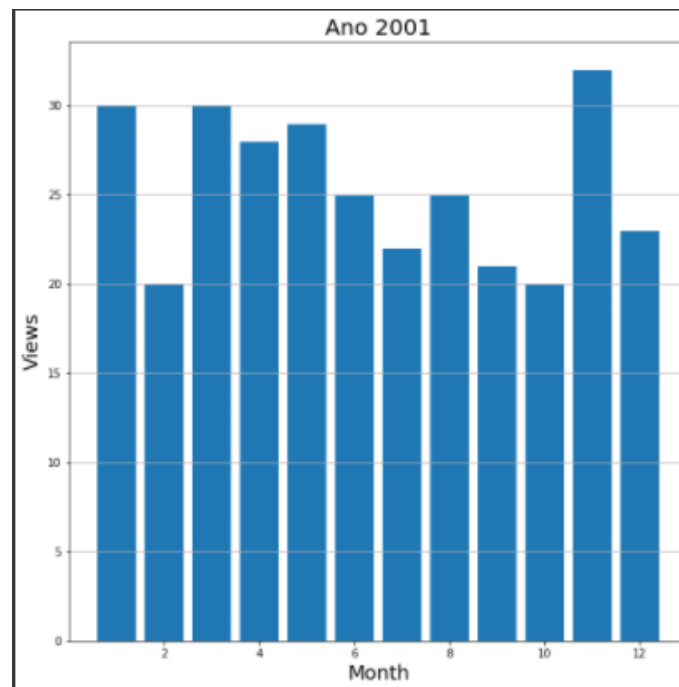
```
#Editar nomes
plt.ylabel('Views', fontsize = 18)
plt.xlabel('Month', fontsize = 18)

#Melhor local
#plt.tight_layout();
# plt.savefig('figure.png', dpi=600, bbox_inches='tight')
plt.show()
```

Fonte: Própria

Ao fim do código o gráfico fica assim, como mostrado na imagem abaixo.

Imagem 9 - Gráfico



Fonte: Própria

A seguir fazemos o código para a coluna do ano e filtros de dados
Imagem 9 - Coluna do ano

```
#Coluna ano
ovnis['Year'] = ovnis['Sight_Date'].str.split('/', expand = True)[2]
#ovnis

#Filtro de dados
q = """
SELECT City, Sight_Day, Sight_Month, Year
FROM ovnis
WHERE City = 'Phoenix'

"""

df2 = pandasql.sqldf(q, locals())
df2
```

.Fonte: Própria

Quantidade de visualizações por ano

Imagem 10 - Por ano

```
q = """
SELECT Year, COUNT(*) AS total
FROM df2
GROUP BY Year
order by Year
"""

agrupar = pandasql.sqldf(q, locals())
agrupar
```

.Fonte: Própria

magem 11 - Gráfico de linha

```
anos = ['1999', '2001', '2003', '2005', '2006', '2007', '2008', '2009', '2010', '2011', '2012', '2013', '2014', '2015', '2016', '2017']
valores = [1, 1, 1, 5, 21, 12, 31, 25, 15, 13, 28, 21, 35, 36, 36, 24]

# anos = agrupar.Year
# valores = agrupar.total

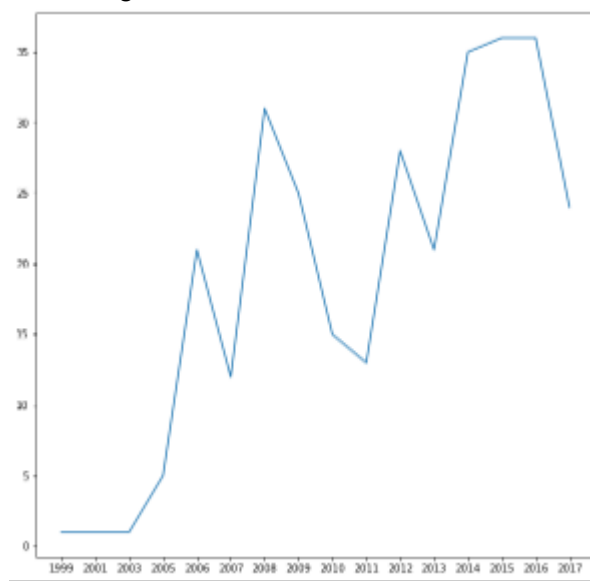
matplotlib.pyplot.plot(anos, valores)

matplotlib.pyplot.show()
```

.Fonte: Própria

O gráfico de linha fica assim:

Imagem 12 - Gráfico de linha Pronto



.Fonte: Própria

E por fim vamos salvar o arquivo em formato CSV.

Imagem 12 - CSV

```
#Gerar um arquivo csv
dados_phoenix.to_csv('df_OVNI_fim.csv')
```

.Fonte: Própria

4. Considerações Finais

Com o fim de semestre, muita correria, dificultou muito o encontro do grupo e o tempo necessário para fazer. Mas com compromisso foi finalizado mais uma etapa. Agradeço aos professores: Diego e Fábio, por mais um projeto que chega ao fim graças a Deus.

Referências

Matheus. Yuri. **Matplotlib uma biblioteca Python para gerar gráficos interessantes**. 2018.
Disponível em: <<https://www.alura.com.br/artigos/criando-graficos-no-python-com-a-matplotlib>>
Acesso em: 25-03-2021,