INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Brasília
Campus Brasília

TECNOLOGIA EM SISTEMAS PARA INTERNET

André Luiz Gomes dos Santos
Brenda Lopes Miranda Teixeira
Luiz Henrique de Paiva Ventura

RELATÓRIO DE PRÁTICA INTEGRADA
DE
CIÊNCIA DE DADOS E INTELIGÊNCIA ARTIFICIAL

Brasília - DF

21/03/2021

Sumário

1. Objetivos

Para esta fase do projeto será realizada uma limpeza de dados, que tem por objetivo;

- Carregar o seu arquivo OVNIS.csv em um dataframe;
- Remover registros que tenham valores vazios (None, Unknown, ...) para City, State e Shape;
- Manter somente os registros referentes aos 51 estados dos Estados Unidos (Links para um site externo.);
- Remover variáveis irrelevantes para a análise (Duration, Summary e Posted);
- Manter somente os registros de Shapes mais populares (com mais de 1000 ocorrências);
- Salvar o dataframe final em um arquivo CSV com o nome "df_OVNI_limpo".

Também será acrescentado variáveis:

- Carregar o seu arquivo df_OVNI_limpo.csv (arquivo gerado após a limpeza de dados efetuada na atividade 5.7) em um dataframe
- Dividir o conteúdo da coluna Date/Time em duas novas colunas no mesmo dataframe e deletar a coluna Date / Time
- Fazer o mesmo procedimento para dias da semana. Será que existe um dia da semana com mais ocorrências de relatórios para OVNIs? Para descobrir isso, você deve criar uma nova coluna chamada weekdays
- Separar as variáveis mês (Month) e dia (Day). Desse modo, será possível refinar as pesquisas.
- Por fim, salvar o dataframe resultante em um arquivo .csv com o nome: 'df_OVNI_preparado.

2. Descrição do problema

Com alguns registros possuindo valores vazios, nessa etapa será necessário remover esses registros de valores vazios e manter registros somente dos cinquenta e um estados dos EUA. Também é necessário acrescentar variáveis tais variáveis auxiliam a solucionar o problema.

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:

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

3.1 Código implementado

Como de costume foi importado as bibliotecas.

Imagem 1 - ovnis.csv

```
[ ]  !pip install -U pandasql
     import pandas as pd
     import numpy as np
     import pandasql
```

1 - Carregar o seu arquivo OVNIS.csv em um dataframe

```
[ ]  #Carregar o seu arquivo OVNIS.csv em um dataframe;
     ovnis = pd.read_csv('OVNIS.csv')
     ovnis
```

Fonte: Própria

Nessa parte do código removemos os registros com valores vazios.

Imagem 2 - Valores Vazios

```
#Removendo os valores vazios e Unknown
ovnis.drop(ovnis.index[ovnis['City'] == 'None'], inplace = True)
ovnis.drop(ovnis.index[ovnis['City'] == None], inplace = True)
ovnis.drop(ovnis.index[ovnis['State'] == None], inplace = True)
ovnis.drop(ovnis.index[ovnis['Shape'] == None], inplace = True)
ovnis.drop(ovnis.index[ovnis['Shape'] == "Unknown"], inplace = True)
#Remove os valores NaN
ovnis['State'].dropna()
ovnis['Shape'].dropna()
ovnis['City'].dropna()

ovnis
```

Fonte: Própria

Conforme solicitado para manter os registros dos 51 estados, na imagem abaixo vai uma demonstração de como fica o código.

Imagem 3 - 51 estados

```
q = """
  SELECT * FROM ovnis WHERE
  STATE LIKE 'AL'
  OR STATE LIKE 'AK'
  OR STATE LIKE 'AZ'
  OR STATE LIKE 'AR'
  OR STATE LIKE 'CA'
  OR STATE LIKE 'CO'
  OR STATE LIKE 'CT'
  OR STATE LIKE 'DE'
  OR STATE LIKE 'DC'
  OR STATE LIKE 'FL'
  OR STATE LIKE 'GA'
  OR STATE LIKE 'HI'
  OR STATE LIKE 'ID'
  OR STATE LIKE 'IL'
  OR STATE LIKE 'IN'
```

Fonte: Própria

Após colocar os estados você deve colocar o comando:

eua_registros = pandasql.sqldf(q.lower(), locals())

eua_registros

Conforme a imagem abaixo 4 mostrada abaixo.

Imagem 4 - 51 Estados

```
  OR STATE LIKE 'WV'
  OR STATE LIKE 'WI'
  OR STATE LIKE 'WY'

"""

eua_registros = pandasql.sqldf(q.lower(), locals())
eua_registros
```

Fonte: Própria

Remover variáveis irrelevantes para a análise (Duration, Summary e Posted).

Imagem 5 - Removendo variáveis

```
#eua_registros
#ajuste de colunas sem espaço
eua_registros.rename(columns = lambda x: x.replace(' ', '_'), inplace=True)

#Trocar o nome da coluna
eua_registros['DateTime'] = eua_registros['Date_/_Time']
#ovnis

#Filtro de dados
q = """
SELECT DateTime, City,  State,  Shape
FROM eua_registros
"""

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

Fonte: Própria

Manter somente os registros de Shapes mais populares (com mais de 1000 ocorrências).

Imagem 6 - Shapes mais populares

```
n_shape = df1['Shape'].value_counts()
maioresque1000 = n_shape[n_shape > 1000]
maioresque1000
```

Fonte: Própria

Aqui agrupamos e filtramos os dados.

Imagem 7 - Filtro

```
#Filtro de dados, agrupar depois filtrar
q = """
SELECT *
FROM df1
WHERE Shape in ('Light', 'Circle', 'Triangle', 'Fireball', 'Sphere', 'Other', 'Oval', 'Disk', 'Formation', 'Changing', 'Cigar', 'Flash', 'Rectangle')

"""


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

Fonte: Própria

Por último geramos o arquivo csv.

Imagem 8 - Salvar data frame

```
#Gerar um arquivo csv
df2.to_csv('df_OVNI_limpo.csv')
```

Fonte: Própria

Agora vamos para a tarefa 5.8, esta tarefa requer o acréscimo de variáveis.

Importando as bibliotecas.

Imagem 9 - Imports

```
#!pip install -U pandasql
import pandas as pd
import numpy as np
#import pandasql
```

Fonte: Própria

Carregar o seu arquivo df_OVNI_limpo.csv.

Imagem 10- df_OVNI_limpo.csv.

```
ovnis = pd.read_csv('df_OVNI_limpo.csv')
ovnis
```

Fonte: Própria

Dividir o conteúdo da coluna Date / Time em duas novas colunas no mesmo dataframe e deletar a coluna Date / Time.

```
#Dividir o conteúdo
ovnis['Sight_Date'], ovnis['Sight_Time'] = ovnis['DateTime'].str.split(' ', 1).str

#Deletar coluna
ovnis.drop(columns=['DateTime'], inplace=True)

ovnis
```

Fonte: Própria

Dia da semana com mais ocorrências de relatórios para OVNIs

Imagem 11- Procedimento para dias da semana.

```
time =  pd.to_datetime(ovnis['Sight_Date'])
time = time.dt.dayofweek

dia_semana={0:'Segunda-feira', 1:'Terça-feira', 2:'Quarta-feira', 3:'Quinta-feira', 4:'Sexta-feira', 5:'Sábado', 6:'Domingo'}
ovnis['Sight_Weekday'] = time.map(dia_semana)
ovnis
```

Fonte: Própria

Separar as variáveis mês (Month) e dia (Day). Desse modo, será possível refinar as pesquisas.

Imagem 10- Month and Day

```
ovnis['Sight_Day'] = ovnis['Sight_Date'].str.split('/', expand = True)[1]
ovnis['Sight_Month'] = ovnis['Sight_Date'].str.split('/', expand = True)[0]
ovnis
```

Fonte: Própria

Por último é gerado o arquivo csv.

Imagem 10- Salvar o arquivo CSV Final

```
#Gerar um arquivo csv
ovnis.to_csv('df_OVNI_preparado.csv')
```

Fonte: Própria

4. Considerações Finais

A partir desse ponto, a participação de todos foi muito importante para concluirmos esse trabalho.

Referências

Dividir data (dia, mês, ano) em novas colunas - DataFrame Pandas. 2021. Disponível em: <https://pt.stackoverflow.com/questions/428236/dividir-data-dia-m%C3%AAs-ano-em-novas-colunas-dataframe-pandas> Acesso em: 22/03/2021