INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Brasília
Campus Brasília

TECNOLOGIA EM SISTEMAS PARA INTERNET

Matheus Nícollas de Souza Mota
Thiago Marinho da Silva Campos

RELATÓRIO DE PRÁTICA INTEGRADA
DE
CIÊNCIA DE DADOS E INTELIGÊNCIA ARTIFICIAL

Brasília - DF
18/09/2020

Sumário

1. Objetivos

O objetivo é acrescentar novas variáveis e modelar os dados para que se apresentem de forma ordenada e funcional para pesquisas.

2. Descrição do problema

O projeto consiste em reunir fatos interessantes relacionados a OVINIs, a partir de relatos realizados dentro de um período de vinte anos usando o site Nuforc. O desafio consiste em fazer a extração de dados de forma tabular, afinal, para que os dados possam ser analisados eles acabam se tornando tabelas.

O WebScraping consiste em extrair os dados formatados com tag's da linguagem HTML. Como iremos extrair vinte anos de dados, consultaremos 240 páginas web, uma por cada mês, por vinte anos, entre setembro 1997 e agosto de 2017.

Para fazer análises é necessário que os dados estejam apresentados de forma ordenada e de uma maneira funcional para que facilite a realização de pesquisas.

3. Desenvolvimento

O desenvolvimento do algoritmo foi feito na plataforma Google Collab, esta plataforma foi escolhida pois ao iniciar um notebook na mesma, as bibliotecas e dependências do Python são todas da nuvem.

Carregamos nosso arquivo df_óvnis_limpo.csv em um dataframe, fizemos a divisão do conteúdo da coluna date em duas novas colunas (date e time) no mesmo dataframe e deletamos a coluna data/time. Fizemos também o mesmo procedimento para dias da semana e criamos uma nova coluna chamada weekdays. Foi separado as variáveis mês e dia, assim poderemos refinar ainda mais as pesquisas.

Por fim salvamos os resultados do tratamento de dados em um arquivo csv.

4 Código implementado

```
import pandas as pd
import datetime as dt

#le o arquivo CSV
df = pd.read_csv("df_OVNI_limpo.csv", index_col=[0])

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

#converte a coluna Data/Time que era string para o tipo datetime
df['Date / Time'] = pd.to_datetime(df['Date / Time'])

#cria uma nova coluna Sight_Date que recebera so a data no formato de
string
df['Sight_Date'] = df['Date / Time'].dt.strftime('%m/%d/%Y')
#cria uma nova coluna Sight_Time que recebera a hora no formato de str
ing
df['Sight_Time'] = df['Date / Time'].dt.strftime('%H:%M')
#exclui a coluna Data/Time
df = df.drop(columns=['Date / Time'])

#plota o dataframe
df
```

```
print("Fazer o mesmo procedimento para dias da semana. Será que existe
um dia da semana com mais ocorrências de relatórios para OVNIs? Para d
escobrir isso, você deve criar uma nova coluna chamada weekdays.")

#converte o tipo da coluna Sight_Date para datetime
df['Sight_Date'] = pd.to_datetime(df['Sight_Date'])

#cria um array com os nomes da semana em portugues
dias = ['Segunda-feira', 'Terça-feira', 'Quarta-feira', 'Quinta-feira'
, 'Sexta-feira', 'Sábado', 'Domingo']

#criar uma nova coluna Sight_weekday que recebe os dias da semana repr
esentados por numeros de 0 a 6
df['Sight_Weekday'] = df['Sight_Date'].dt.dayofweek

#cria um for que insere o nome de todos os dias da semana baseado no i
ndice do vetor no qual sera fornecido pela coluna criada acima
for i in df.itertuples():
  df.Sight_Weekday[i.Index] = dias[df.Sight_Weekday[i.Index]]

#plota o dataframe
df
```

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

#cria uma nova coluna Sight_Day na qual recebe o dia separado da data
df['Sight_Day'] = df['Sight_Date'].dt.day
#cria uma nova coluna Sight_Month na qual recebe o mes separado da dat
a
df['Sight_Month'] = df['Sight_Date'].dt.month

#plota o dataframe
df
```

```
print("Por fim, salvar o dataframe resultante em um arquivo .csv com o nome: 'df_OVNI_preparado'.")

#gera arquivo CSV
df.to_csv("df_OVNI_preparado.csv")
```

Github: https://github.com/Prof-Fabio-Henrique/pratica-integrada-icd-e-iia-2020-1-g13-mmt

5 Considerações Finais

Com esta etapa do projeto podemos trabalhar no tratamento dos dados, preparand-os para uma etapa de análise com maior granularidade onde será possível mensurar e obter respostas claras a determinadas questões. Abaixo segue o resultado obtido.

	City	State	Shape	Sight_Date	Sight_Time	Sight_Weekday	Sight_Day	Sight_Month
1	Solomons Island	MD	Disk	1997-09-22	20:00	Segunda-feira	22	9
2	Garden Grove	CA	Rectangle	1997-09-19	00:00	Sexta-feira	19	9
4	Houston	TX	Disk	1997-09-15	00:00	Segunda-feira	15	9
5	Santa Fe	NM	Light	1997-09-15	20:00	Segunda-feira	15	9
6	Kent	WA	Sphere	1997-09-15	20:00	Segunda-feira	15	9
...	...	...	...	...	...	...	...	...
71895	Columbus (North)	GA	Fireball	2017-08-01	06:15	Terça-feira	1	8
71896	Corcoran	MN	Light	2017-08-01	02:45	Terça-feira	1	8
71897	Moreno Valley	CA	Other	2017-08-01	02:00	Terça-feira	1	8
71898	Bradenton	FL	Other	2017-08-01	01:00	Terça-feira	1	8
71900	Laurel	MD	Other	2017-08-01	00:00	Terça-feira	1	8

6 Referencias

PANDAS. Https://pandas.pydata.org/docs/. Disponível em: https://pandas.pydata.org/docs/. Acesso em: 19 set. 2020.

REQUESTS. Requests: HTTP for Humans™. Disponível em: https://requests.readthedocs.io/en/master/. Acesso em: 19 set. 2020.

SOUP, Beautiful. Beautiful Soup Documentation. Disponível em: https://www.crummy.com/software/BeautifulSoup/bs4/doc/. Acesso em: 19 set. 2020.

SILVA, Réulison. Como fazer um Web Scraping com Python. Disponível em: https://goomore.com/blog/web-scraping-python/. Acesso em: 19 set. 2020