INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Brasília
Campus Brasília

TECNOLOGIA EM SISTEMAS PARA INTERNET

Daniel Evangelista Pereira
Ribson Coelho Cardoch Valdés
Douglas Seidi Shibata

RELATÓRIO DE PRÁTICA INTEGRADA DE CIÊNCIA DE DADOS E INTELIGÊNCIA ARTIFICIAL

19/08/2020

Brasília - DF

2020

Sumário

1. Objetivos

Está etapa do projeto tem como finalidade a exploração dos dados utilizando a linguagem SQL no script do notebook em python, que foram coletados na fase anterior por meio de web scraping, utilizando a linguagem python e gerando um arquivo Comma-separated values(CSV). Que será utilizado para a exploração. E o código estará armazenado no repositório do Github com a utilização do versionamento de código do git.

2. Descrição do problema

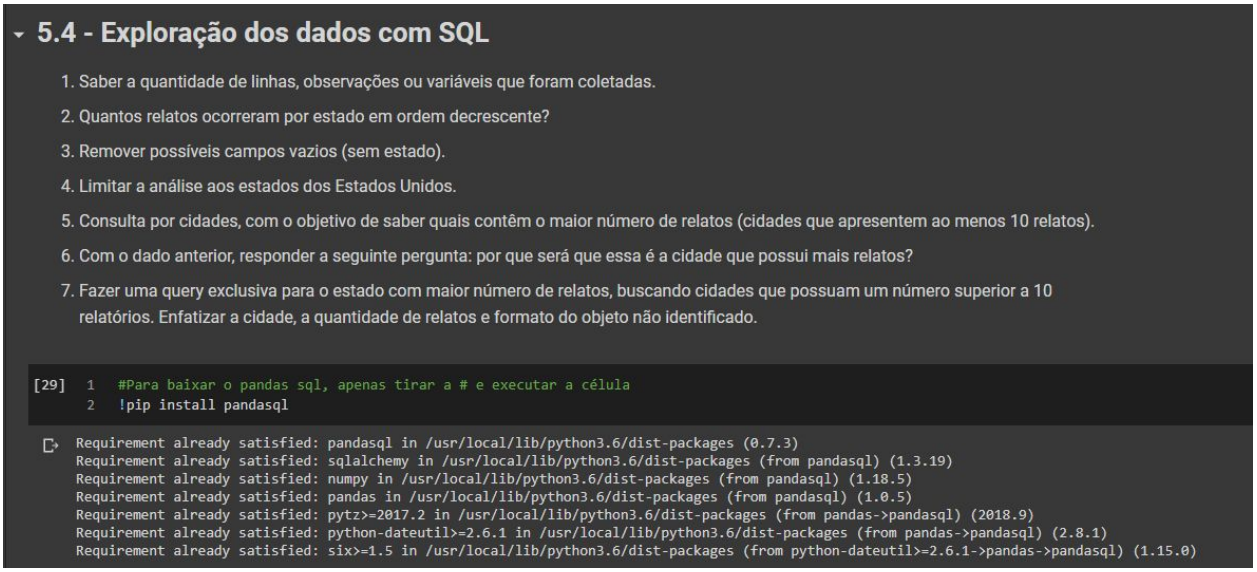
Ao iniciar o desenvolvimento do projeto podemos destacar a forma da coleta dos dados e como estão distribuídos no site e a partir desta análise inicial, foi visto que havia uma grande quantidade de dados em diversas páginas que precisar ser extraídas do site para realizar a coleta de dados e armazenar em DataFrames para que se possa gerar um arquivo csv para começar o estudo dos dados na primeira etapa do projeto. E a partir dos dados que foram gerados e armazenados no csv será feita a exploração de dados com a utilização da linguagem sql para a melhor filtragem dos dados.

3. Desenvolvimento

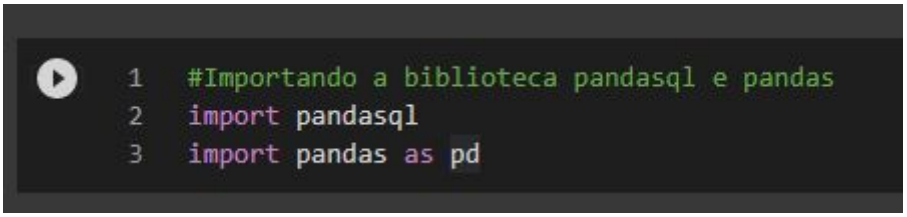
As tecnologias utilizadas para a elaboração da primeira parte do projeto na fase de coleta de dados consiste na linguagem de programação python com o auxílio de suas bibliotecas, como o requests para fazer a requisição do site e retornar o status code, a biblioteca BeautifulSoup para a leitura das páginas em Hyper Text Markup Language(HTML) e para o armazenamento de dados foi utilizada a biblioteca pandas, e o pandasql para que seja possível executar códigos SQL para fazer consultas no dataframe de ovnis, por meio do notebook e o ambiente de desenvolvimento Google Colab.

3.1 Código implementado

5.4 - Exploração de dados com SQL



Baixando a biblioteca pandasql para a utilização de códigos SQL na exploração de dados



Importação das bibliotecas pandasql e pandas para exploração de dados com sql e a leitura e geração de arquivos csv, respectivamente.

```
[25]  1  ovnis = pd.read_csv('OVNIS.csv')
      2  ovnis
```

	Date / Time	City	State	Shape	Duration	Summary	Posted
0	9/30/97 20:00	Nova Scotia (Canada)	NS	Light	8-10 seconds.	Single light resembling a star, but moving spu...	10/30/06
1	9/22/97 20:00	Solomons Island	MD	Disk	10 minutes	Close up at twilight, Stationary UFO.	8/5/09
2	9/19/97	Garden Grove	CA	Rectangle	4 mins.	Around 6:30 PM I was walking through a Vons Pa...	12/1/19
3	9/18/97 20:15	Panama City	FL	Unknown	30 seconds	Looked like stars in the sky so far up/moveing...	3/13/12
4	9/15/97 00:00	Houston	TX	Disk	5 minutes	Beautiful silver-colored flying saucer about t...	7/19/10
...	...	...	...	...	...	...	...
71896	8/1/17 02:45	Corcoran	MN	Light	Still going	Small light south west of Minneapolis maneuver...	8/4/17
71897	8/1/17 02:00	Moreno Valley	CA	Other	10 seconds	I was looking out the front windshield and loo...	8/4/17
71898	8/1/17 01:00	Bradenton	FL	Other	<20 seconds	I was walking my dog about 1am on August 1, 20...	5/9/19
71899	8/1/17 00:00	Springdale	AR	NaN	1 hour	Glowing flying people . seven of them flying l...	2/13/20
71900	8/1/17	Laurel	MD	Other	NaN	It was an alien project level 1 federal ran on...	6/25/20

71901 rows × 7 columns

Leitura do arquivo OVNIS csv e armazenando na variável ovnis

```
[26]  1  #Saber a quantidade de linhas, observações ou variáveis que foram coletadas.
      2  print("Quantidade de Linhas coletadas: ",len(ovnis))

   Quantidade de Linhas coletadas:  71901
```

Imprimindo a quantidade de linhas que foram coletadas na atividade anterior

```
      1  #Quantos relatos ocorreram por estado em ordem decrescente?
      2  estados =ovnis.State.value_counts()
      3  estados.sort_values(ascending=False)

CA    7911
FL    4352
WA    3225
TX    2882
NY    2824
      ...
NF      21
YT      14
PE       9
NT       7
SA       4
Name: State, Length: 64, dtype: int64
```

Declarando a variável estados e contando os estados por valores, ou seja, por siglas que apresentam na base de dados e depois organizando as siglas por ordem crescente.

```
#Remover possíveis campos vazios (sem estado).
ovnis.State.dropna()
ovnis.State.sort_values(ascending=True).unique()
```

```
array(['AB', 'AK', 'AL', 'AR', 'AZ', 'BC', 'CA', 'CO', 'CT', 'DC', 'DE',
       'FL', 'GA', 'HI', 'IA', 'ID', 'IL', 'IN', 'KS', 'KY', 'LA', 'MA',
       'MB', 'MD', 'ME', 'MI', 'MN', 'MO', 'MS', 'MT', 'NB', 'NC', 'ND',
       'NE', 'NF', 'NH', 'NJ', 'NM', 'NS', 'NT', 'NV', 'NY', 'OH', 'OK',
       'ON', 'OR', 'PA', 'PE', 'QC', 'RI', 'SA', 'SC', 'SD', 'SK', 'TN',
       'TX', 'UT', 'VA', 'VT', 'WA', 'WI', 'WV', 'WY', 'YT', nan],
      dtype=object)
```

Removendo os possíveis campos vazios da base de dados.

```
#Limitar a análise aos estados dos Estados Unidos.
q = """
    SELECT * from total where State LIKE '%AK%' OR State LIKE '%AL%'
OR State LIKE '%AR%'
OR State LIKE '%AZ%'
OR State LIKE '%CA%'
OR State LIKE '%CO%'
OR State LIKE '%CT%'
OR State LIKE '%DE%'
OR State LIKE '%FL%'
OR State LIKE '%GA%'
OR State LIKE '%HI%'
OR State LIKE '%IA%'
OR State LIKE '%ID%'
OR State LIKE '%IL%'
OR State LIKE '%IN%'
OR State LIKE '%KS%'
OR State LIKE '%KY%'
OR State LIKE '%LA%'
OR State LIKE '%MA%'
OR State LIKE '%MD%'
OR State LIKE '%ME%'
OR State LIKE '%MI%'
OR State LIKE '%MN%'
OR State LIKE '%MO%'
OR State LIKE '%MS%'
OR State LIKE '%MT%'
OR State LIKE '%NC%'
OR State LIKE '%ND%'
OR State LIKE '%NE%'
OR State LIKE '%NH%'
OR State LIKE '%NJ%'
OR State LIKE '%NM%'
OR State LIKE '%NV%'
OR State LIKE '%NY%'
OR State LIKE '%OH%'
OR State LIKE '%OK%'
OR State LIKE '%OR%'
OR State LIKE '%PA%'
OR State LIKE '%RI%'
OR State LIKE '%SC%'
OR State LIKE '%SD%'
OR State LIKE '%TN%'
OR State LIKE '%TX%'
OR State LIKE '%UT%'
OR State LIKE '%VA%'
OR State LIKE '%VT%'
OR State LIKE '%WA%'
OR State LIKE '%WI%'
OR State LIKE '%WV%'
OR State LIKE '%WY%'
    """
# Executa o seu comando SQL e retorna um dataframe
just_us = pandasql.sqldf(q.lower(), locals())
just_us
```

Limitando os estados para apenas estados norte americanos e executando o comando sql.

	Date / Time	City	State	Shape	Duration	Summary	Posted
0	9/22/97 20:00	Solomons Island	MD	Disk	10 minutes	Close up at twilight, Stationary UFO.	8/5/09
1	9/19/97	Garden Grove	CA	Rectangle	4 mins.	Around 6:30 PM I was walking through a Vons Pa...	12/1/19
2	9/18/97 20:15	Panama City	FL	Unknown	30 seconds	Looked like stars in the sky so far up/moveing...	3/13/12
3	9/15/97 00:00	Houston	TX	Disk	5 minutes	Beautiful silver-colored flying saucer about t...	7/19/10
4	9/15/97 20:00	Santa Fe	NM	Light	2-3 minutes	Saw white dot of light moving in zig-zag motio...	11/9/17
...	...	...	...	...	...	...	...
64892	8/1/17 02:45	Corcoran	MN	Light	Still going	Small light south west of Minneapolis maneuver...	8/4/17
64893	8/1/17 02:00	Moreno Valley	CA	Other	10 seconds	I was looking out the front windshield and loo...	8/4/17
64894	8/1/17 01:00	Bradenton	FL	Other	<20 seconds	I was walking my dog about 1am on August 1, 20...	5/9/19
64895	8/1/17 00:00	Springdale	AR	None	1 hour	Glowing flying people . seven of them flying l...	2/13/20
64896	8/1/17	Laurel	MD	Other	None	It was an alien project level 1 federal ran on...	6/25/20

64897 rows × 7 columns

Resultado da consulta feita para filtrar apenas estados norte americanos.

```
[35]  1  just_us.to_csv('dados_usa.csv')
```

Gerando um arquivo csv dos dados que foram gerados anteriormente, com a filtragem de apenas estados americanos.

```
[ ]  1  cidades_usa =just_us.City.value_counts()
     2  cidades_usa[cidades_usa >=10]
     3  cidades_usa.head()
```

```
Phoenix      366
Las Vegas    338
Seattle      323
Portland     318
San Diego    272
Name: City, dtype: int64
```

Contando a quantidade de caso por cidades dos estados unidos, e mostrando apenas os estados que têm mais de 10 ocorrências. E mostrando apenas os 5 mais recorrentes.

```
#Fazer uma query exclusiva para o estado com maior número de relatos, buscando cidades que possuam um número superior a 10 relatórios.
#Enfatizar a cidade, a quantidade de relatos e formato do objeto não identificado.

# Executa o seu comando SQL e retorna um dataframe
query = '''
 SELECT CITY,Count(City) as 'Numero de Ocorrencias', Shape FROM just_us  group by city,shape  having count(City)>=10 order by  count(City) desc
'''
pandasql.sqldf(query.lower(), locals())
```

	City	numero de ocorrencias	Shape
0	Seattle	75	Light
1	Las Vegas	74	Light
2	Phoenix	72	Light
3	San Diego	71	Light
4	Myrtle Beach	66	Light
...	...	...	...
564	Virginia Beach	10	Sphere
565	Visalia	10	Light
566	Wellington	10	Light
567	Woodbridge	10	Light
568	York	10	Circle

569 rows × 3 columns

Identificando os formatos dos objetos não identificados por cidade e quantidade de ocorrência do formato que foi relatado.

4. Considerações Finais

Nesta parte de exploração de dados, houve estudos das bibliotecas utilizadas e o modo que as consultas foram realizadas por meio da linguagem de consulta estruturada. No processo de desenvolvimento houve alguns empecilhos, como a forma de executar alguns comandos SQL para que se obtenha o exato resultado e a remoção de dados nulos, houve uma pequena dificuldade, pois acredita-se que alguns campos não estejam nulos, mas está escrito com sem valor.

Referências

ANDRADE, Andrew. Análise Exploratória de Dados.**Escola de Dados**.[s.d.]. Disponível em:<https://escoladedados.org/tutoriais/analise-exploratoria-de-dados/>.Acesso em: 13, Setembro de 2020.

GEROLA, Letícia. 5 pythons hacks para a exploração do dados.**Medium**, 2020. Disponivelem:<https://medium.com/joguei-os-dados/4-python-hacks-para-explora%C3%A7%C3%A3o-de-dados-8c89931b6d1f>. Acesso em: 13, Setembro de 2020.

CARACIOLO, Marcel. **Introdução a análise exploratória com python e pandas**. 2013.(1h32m25s).Disponivel em: <https://www.youtube.com/watch?v=vIJwq6QjZL8>. Acesso em: 14, Setembro de 2020.