INSTITUTO FEDERAL
Brasília
Campus Brasília

TECNOLOGIA EM SISTEMAS PARA INTERNET

RELATÓRIO DE PRÁTICA INTEGRADA DE CIÊNCIA DE DADOS E APRENDIZADO DE MÁQUINA PARA A SPRINT I

Davi Pereira Speck Alves
Marcus Vinicius da Silva Romero
Pedro Stewart Silva Borges
Vinícius dos Santos Evangelista Dias

Brasília - DF
25/06/2022

Sumário

Objetivos Gerais

- Extrair dados de forma tabular do site “Nuforc”;
- Explorar os dados extraídos a partir do “*web scraping*” com SQL;

Descrição do problema

Será feita uma extração de dados de forma tabular, pois, parte-se do pressuposto de que para analisar fatos relacionados a OVNIs do site Nuforc (*The National UFO Reporting Center*), eles acabam necessariamente se tornando tabelas. Para tanto, a partir da extração de vinte anos de dados, serão consultadas 250 *páginas webs* , uma por cada mês, por vinte anos, entre setembro de 1997 e agosto de 2017. Além disso, torna-se importante explorar os dados para ter um primeiro contato com a realidade do que se está investigando.

Desenvolvimento

O projeto será desenvolvido principalmente em Python, devido à alta gama de possibilidades que a mesma oferece para trabalhar com a manipulação de dados. Porém, mediante necessidade, certos pontos podem ser feitos a partir de outras linguagens também.

Tecnologias Utilizadas

- **Python**: Principal tecnologia do projeto, utilizada para manipulação de dados e *web scraping*, com juntamente com as bibliotecas “Requests”, “Beautiful Soup”, “Pandas”, “Pandasql”, “Matplotlib”, “Altair”, “Folium” e “Geopandas”;
- **Javascript**: Utilizado para filtrar por intervalo de tempo para estudo mais direcionado conforme solicitado. Para trabalhar com CSV no Javascript, foram utilizadas as seguintes bibliotecas: csv-parser e objects-to-csv, além da biblioteca nativa da linguagem para trabalhar com arquivos. Para montar os gráficos em Javascript, usamos a biblioteca ChartJS. A filtragem, contudo, foi feita manualmente sem uso de bibliotecas. Mais adiante, para o armazenamento dos dados em uma base MongoDB foi utilizada a biblioteca Mongoose.

Link do repositório no GitHub

https://github.com/infocbra/pratica-integrada-cd-e-am-2022-1-dmpv.gi

Sprint I

1 - Introdução e coleta de dados

1.1 - Realizando imports para desenvolvimento da Sprint I e fazendo requisição na página inicial dos *reports* de avistamentos

```
import requests
from bs4 import BeautifulSoup
import pandas as pd
import numpy as np
from pandas import DataFrame, Series
from datetime import datetime
#!pip install pandasql
import pandasql

# Requisitando dados da página de eventos por data
r = requests.get('https://nuforc.org/webreports/ndxevent.html')
```

1.2 - Montando arrays para criação do dataframe inicial

Os arrays criados possuem dados da tabela mais exterior dos *webreports* no site nuforc, contendo, respectivamente, data, com mês/ano, contagem de avistamentos e links para cada data em específico, assim como na página da requisição.

```
data = []

for index, link in enumerate(soup.find_all('a')):
    if index != 0 and link.text != 'UNSPECIFIED / APPROXIMATE':
      aux = datetime.strptime(link.text, '%m/%Y').strftime('%m/%Y')
      data.append(aux)
```

```
contagem = []

for index, link in enumerate(soup.find_all('td')):
    if (index % 2 != 0) and link.text != '376':
      contagem.append(int(link.text))
```

```
links = []

for index, link in enumerate(soup.find_all('a')):
    if index != 0 and link.get('href') != 'ndxe.html':
```

```
      links.append(str('https://nuforc.org/webreports/' +
link.get('href')))
```

1.3 - Função para criação de dataframe e posterior visualização

```
def create_dataframe(data, contagem, links):
  d = {
    "Data": data,
    "Contagem": contagem,
    "Links": links
  }

  df = DataFrame(d)

  return df
```

```
df = create_dataframe(data, contagem, links)
df.to_csv("ovnis.csv", index = False)
```

	Data	Contagem	Links
0	08/2017	421	https://nuforc.org/webreports/ndxe201708.html
1	07/2017	528	https://nuforc.org/webreports/ndxe201707.html
2	06/2017	430	https://nuforc.org/webreports/ndxe201706.html
3	05/2017	369	https://nuforc.org/webreports/ndxe201705.html
4	04/2017	423	https://nuforc.org/webreports/ndxe201704.html
...	...	...	...
235	01/1998	85	https://nuforc.org/webreports/ndxe199801.html
236	12/1997	76	https://nuforc.org/webreports/ndxe199712.html
237	11/1997	132	https://nuforc.org/webreports/ndxe199711.html
238	10/1997	116	https://nuforc.org/webreports/ndxe199710.html
239	09/1997	61	https://nuforc.org/webreports/ndxe199709.html

240 rows × 3 columns

1.4 - Delimitando .csv com intervalo de tempo solicitado (1997-2017)

Em JavaScript, com o arquivo .csv montado em Python na etapa anterior, foi definido um outro documento csv: "dados-filtrados.csv", constando neste o intervalo de datas solicitado. A função “verificaData” nada mais faz do que verificar por meio de estruturas condicionais se o ano é menor que 2017 e maior que 1997, bem como se o mês de 1997 é maior do que setembro e se o mês de 2017 é menor que agosto. Somente os resultados que retornam “true” são colocados

no array de objetos filtrados. Por fim, esse array é convertido para CSV utilizando-se a biblioteca “objects-to-csv”, a partir da função “toCsv”.

O código abaixo lê o arquivo OVNIS.csv, filtra-o e gera outro arquivo CSV chamado “dados-filtrados”.

```
let resultado = []; let filtrados = [];

fs.createReadStream('OVNIS.csv')
.pipe(csvParser())
.on('data', (data) => resultado.push(data))
.on('end', async () => {

  for (let i = 0; i < resultado.length; i++) {
    if (verificaData(resultado[i].Data)) {
      filtrados.push(resultado[i]);
    }
  }


  toCsv(filtrados, 'dados-filtrados');
});
```

1.5 - Construíndo .csv com tabelas vindas da coluna “links” determinada anteriormente

```
df = pd.read_csv('./dados-filtrados.csv')
array = df.values.tolist()

tabelas = []

for val in array:
    tabelas.append(pd.read_html(val[2]))

len(tabelas)
```

```
3    about 3 min.  white,small,quiet,smooth,cigar shaped but poin...   4/1/01
4       2-4 min.  White, cigar-shaped object flying over Dublin,...   4/1/01
..           ...                                                ...      ...
287   few seconds  A UFO mistaken for an aeroplane, caught on cam...   3/6/01
288     5 minutes  small black oval, moving extremely slowly in b...  5/24/05
289     2 minutes     Triangles seen flying in thin layer of clouds.   8/5/01
290        1 hour  well i was closeing the curtins when i noticed...   3/6/01
291     2min,4min      Three UFO's seen flying over the Klamath sky.  3/16/01

     Images
0       NaN
1       NaN
2       NaN
3       NaN
4       NaN
..      ...
287     NaN
288     NaN
289     NaN
290     NaN
291     NaN

[292 rows x 9 columns]],
[       Date / Time                City State         Country      Shape  \
 0    2/28/01 23:50    Caldicot (UK/Wales)   NaN  United Kingdom      Flash
 1    2/28/01 20:00             Vincennes    IN             USA      Light
```

```
frames = []

for val in tabelas:
  frames.append(val[0])

frames_concatenados = pd.concat(frames)

frames_concatenados.to_csv('todos_ovnis.csv', sep=',')
```

2 - Exploração

2.1 - Saber a quantidade de linhas, observações ou variáveis que foram coletadas.

```
# RESPOSTA: 100398

q1 = pd.read_csv('todos_ovnis.csv')

q = """
SELECT * FROM q1;
"""

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

...	...	...	...	...	...	...	...	...	...	...
100393	56	9/2/97 15:00	Zurich (Switzerland)	None	Switzerland	Disk	unknown	fast moving object merging from in back of swi...	1/28/99	None
100394	57	9/1/97 23:00	Albany	ME	USA	Light	2 min	Star makes right angle turns, speeds away.	2/1/07	None
100395	58	9/1/97 22:30	Carlsbad	CA	USA	Light	3-5 min	Round orb of light seen in back yard.between a...	12/12/09	None
100396	59	9/1/97 20:00	Pleasant View	TN	USA	Fireball	5 seconds	large white fireball or super flare...	10/31/03	None
100397	60	9/1/97 18:00	Woodmont	CT	USA	Disk	15minutes	walking my dog along the beach,I was taking pi...	2/22/02	None

100398 rows × 10 columns

2.2 - Quantos relatos ocorreram por estado em ordem decrescente?

```
q2 = pd.read_csv('todos_ovnis.csv')

q = """
SELECT COUNT(state), state FROM q2 GROUP BY state HAVING COUNT(state) > 1
ORDER BY count(state) DESC;
"""

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

	COUNT(state)	State
0	11470	CA
1	5618	FL
2	4925	WA
3	4181	TX
4	3899	NY
...	...	...
63	21	YT
64	20	PE
65	18	PR
66	5	YK
67	3	Ontario

68 rows × 2 columns

2.3 - Remover possíveis campos vazios (sem estado).

```
q3 = pd.read_csv('todos_ovnis.csv')
```

```
q = """
SELECT * FROM q3 WHERE state is null OR state = "" OR state = "None";
"""

# Estes são os campos a serem removidos
pandasql.sqldf(q, locals())
```

7237	81	10/11/97 23:30	Eslö, Scania (Sweden)	None	Sweden	None	10 minutes	During an IYCW (International Youth Camp Weeke...	1/28/99	None
7238	82	10/11/97 22:00	Hafnarfjordur (Iceland)	None	Iceland	Sphere	5 min	playing with a jet	6/12/08	None
7239	23	9/18/97 22:15	Doha (Qatar)	None	Qatar	Circle	10-15 sec	UFO sighting In a Gulf Desert	8/5/01	None
7240	48	9/10/97 16:00	Lierskogen (Norway)	None	Norway	Unknown	3 minutes	It looked as a pointed horseshoe. It was full ...	4/26/99	None
7241	56	9/2/97 15:00	Zurich (Switzerland)	None	Switzerland	Disk	unknown	fast moving object merging from in back of swi...	1/28/99	None

7242 rows × 10 columns

2.4 - Limitar a análise aos estados dos Estados Unidos.

```
q4 = pd.read_csv('todos_ovnis.csv')

q = """
SELECT * FROM q4 WHERE Country = "USA";
"""

pandasql.sqldf(q, locals()).head()
```

	Unnamed: 0	Date / Time	City	State	Country	Shape	Duration	Summary	Posted	Images
0	0	8/31/17 22:00	Norwalk	CT	USA	Light	Extremely brief	Bright green light zig-zagged in the sky and d...	9/5/17	None
1	1	8/31/17 22:00	Elizabeth	WV	USA	Light	13 seconds	((HOAX??)) Looked like a star that moved acros...	9/5/17	None
2	2	8/31/17 21:00	San Diego	CA	USA	Rectangle	30 seconds	Rectangle four white lights two red flashing l...	9/5/17	None
3	3	8/31/17 20:15	E. Rio Vista	CA	USA	Light	20 seconds	I'm a truck driver headed E on Hwy 12 just W o...	9/5/17	None
4	4	8/31/17 19:30	Magna	UT	USA	Sphere	30 minutes	Bright glowing, reflected surface. Sphere-like...	9/5/17	None

2.5 - Consulta por cidades, visando com o objetivo de saber quais contêm o maior número de relatos (cidades que apresentem ao menos 10 relatos).

```
q5 = pd.read_csv('todos_ovnis.csv')
```

```
q = """
SELECT COUNT(city), city FROM q5 GROUP BY city HAVING COUNT(city) > 10
ORDER BY count(city) DESC;
"""

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

	COUNT(city)	City
0	561	Phoenix
1	550	Seattle
2	482	Portland
3	476	Las Vegas
4	396	San Diego
...	...	...
1799	11	Wrightsville Beach
1800	11	Xenia
1801	11	Yorktown
1802	11	Zeeland
1803	11	unknown

1804 rows × 2 columns

2.6 - (Questão) Com o dado anterior, responder a seguinte pergunta: por que será que essa é a cidade que possui mais relatos?

Resposta: No ano de 1997, ocorreu um fenômeno que ficou conhecido como “Luzes de Phoenix” na cidade de Phoenix. É possível que tal evento, considerando a sua magnitude, tenha mexido com o imaginário popular, a ponto de a população em geral criar um viés de confirmação, interpretando qualquer avistamento como OVNI. Além disso, Phoenix está no estado do Arizona, que faz fronteira com o país México. Nessa divisa, pode existir um maior fluxo de aeronaves das forças aéreas de ambos os países.

2.7 - Fazer uma query exclusiva para o estado com maior número de relatos, buscando cidades que possuam um número superior a 10 relatórios.

```
q7_2 = pd.read_csv('todos_ovnis.csv')

q = """
SELECT COUNT(city) AS "Number of Reports", state, city, shape FROM q7_2
WHERE state = "CA" GROUP BY city HAVING COUNT(city)> 10 ORDER BY
COUNT(city) DESC;
"""

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

	Number of Reports	State	City	Shape
0	392	CA	San Diego	Light
1	382	CA	Los Angeles	Triangle
2	244	CA	Sacramento	light
3	224	CA	San Jose	Triangle
4	197	CA	San Francisco	Triangle
...	...	...	...	...
250	11	CA	Nipomo	Light
251	11	CA	Pismo Beach	Triangle
252	11	CA	Seal Beach	Light
253	11	CA	West Los Angeles	Sphere
254	11	CA	Wildomar	Disk

255 rows × 4 columns

2.7.1 - Enfatizar a cidade, a quantidade de relatos e formato do objeto não identificado

```
q7_1 = pd.read_csv('todos_ovnis.csv')

q = """
SELECT COUNT(state), state FROM q7_1 GROUP BY state ORDER BY COUNT(state)
DESC;
"""

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

	COUNT(state)	State
0	11470	CA
1	5618	FL
2	4925	WA
3	4181	TX
4	3899	NY
...	...	...
81	1	South Carolina
82	1	Southern District
83	1	VI
84	1	Washington, DC
85	0	None

86 rows × 2 columns

Considerações finais

Na sprint, foi trabalhado *a* web *scraping*, mesmo não sendo uma parte complexa da programação, houve a necessidade de estudar a documentação antes de qualquer prática. Por isso, houve um aprendizado imenso ao descobrir as diversas possibilidades de chegar a um mesmo objetivo, utilizando inclusive as bibliotecas que auxiliam na execução do termo no Python.

Referências

1. PyData Organization. **pandas documentation**. “pydata.org”, 2022. Disponível em: https://pandas.pydata.org/docs/. Acesso em: 26/06/2022
2. Reitz, Kenneth. **Requests HTTP For Humans**. Requests, 2022. Disponível em: https://requests.readthedocs.io/en/latest/. Acesso em: 26/06/2022
3. Richardson, Leonard. **Beautiful Soup Documentation**. Crummy, 2022. Disponível em: https://www.crummy.com/software/BeautifulSoup/bs4/doc/. Acesso em: 26/06/2022
4. Openbase, Inc. **Beautiful Soup Documentation**. Openbase, 2022. Disponível em: https://openbase.com/python/pandasql. Acesso em: 26/06/2022
5. Story, Rob. **Folium**. “python-visualization”, 2013. Disponível em: https://python-visualization.github.io/folium/. Acesso em: 08/07/2022
6. Mafintosh. **CSV Parser**. Disponível em: <https://github.com/mafintosh/csv-parser>. Acesso em: 02/08/2022
7. Antonbot. **Objects To CSV**.. Disponível em: <https://github.com/anton-bot/objects-to-csv>. Acesso em: 02/08/2022