INSTITUTO FEDERAL
Brasília
Campus Brasília

TECNOLOGIA EM SISTEMAS PARA INTERNET

Davi Soares Luna
Jorge Kayodê Lima Trindade
Marcos Rian Tomé de Oliveira
Nathália Teixeira Guimarães

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

Brasília - DF

03/08/2021

Sumário

1. Objetivos

Analisar, tratar e explorar dados do site The National UFO Reporting Center relativos a objetos voadores não identificados avistados por pessoas nos Estados Unidos.

2. Descrição do problema

A The National UFO Reporting (NUFORC) é uma organização nos Estados Unidos (EUA) que investiga o aparecimento de OVNIs (objetos voadores não identificados) e/ou contatos com extraterrestres.

A NUFORC foi fundada em 1974 e já catalogou mais de 90.000 relatos de OVNIS ao longo de sua existência, conforme a Wikipedia. Em seu site, eles disponibilizam os dados de todos esses relatos gratuitamente e sem compromisso com a veracidade, já que não conseguem averiguar todas as ocorrências.

Além disso, o Centro fornece um número de telefone de linha direta 24 horas para que qualquer pessoa possa relatar atividades de OVNIs ocorrendo em sua cidade (Wikipédia).

Esse trabalho tem como objetivo extrair e explorar os dados disponibilizados no site da NUFORC fazendo uso de ferramentas de (aprendizado de máquina e ciência de dados) linguagem de programação.

3. Desenvolvimento

Na etapa de **coleta** e **exploração** dos dados no site NUFORC, o grupo G2DJMN fez uso da plataforma online Google Collaboratory devido à sua facilidade de programação online e conjunta pelos membros do grupo.

O código foi desenvolvido em um Script Python e foram utilizadas técnicas como o **Web Scraping**, além de serem usadas bibliotecas como a **request** (para execução de requisições HTTP), **Pandas** (para armazenar, limpar e salvar os dados em forma de tabela) durante a fase de *coleta*.

Posteriormente foram usadas as bibliotecas **pandasql**, **MatplotLib** e **Seaborn** para fazer a *exploração* desses dados em SQL.

3.1 Pacotes e .CSV da pagina HTML

Nesta primeira parte do código foi feita a importação dos pacotes a serem usados ao longo do Script Python.Também foi feita a leitura da página do NUFORC e guardado seu o conteúdo na variável dataHTML. Em seguida, foi criado o arquivo 'file.csv' com o conteúdo desta página.

Figura 1 - Importação das Bibliotecas e Geração do arquivo .CSV a partir da página HTML

```
1  #Os imports não são finais
2  import pandas as pd
3  import sqlite3
4  import numpy as np
5  import requests
6
7  #Para gerar o arquivo .csv sem tratamento a partir da página .html
8  dataHTML = pd.read_html('http://www.nuforc.org/webreports/ndxevent.html')
9  dataHTML[0].to_csv('file.csv')
```

Fonte: Própria

3.2 Coleta

Aqui foi criada uma lista com as colunas 'Reports' e 'Count'. Depois estas colunas foram usadas na criação do banco de dados 'newData'. A partir disso, delimitamos que nosso banco de dados 'newData' usará somente o conteúdo indexado entre [45:285].

Figura 2 - Tratamento e Intervalo do DataFrame

```
12 #Para fazer o tratamento do DataFrame do pandas:
13 keep_col = ['Reports','Count']  #para o tratamento do arquivo com os index para o .csv
14 newData = dataCSV[keep_col]
15
16 #E agora para deixar apenas os dados do intervalo de tempo desejado
17 newData = newData.iloc[45:285]
18
```

Fonte: Própria

Nesta parte do código, foi criado outro arquivo .CSV ‘OVNIS_Index.csv’ que será usado como index durante o armazenamento dos dados coletados na página HTML.

Figura 3 - Criação de novo arquivo .CSV a partir do DataFrame tratado

```
19  #Gerar o novo arquivo usado como o dos dados que queremos acessar
20  newData.to_csv("OVNIs_Index.csv", index = False)
21  indexOVNIs = pd.read_csv('OVNIs_Index.csv')
```

Fonte: Própria

Aqui foi feito o tratamento para inserir o conteúdo das tabelas HTML dentro de cada link. Para isso foi realizada a concatenação do link do site com a variável date, que é modificada conforme o link que se quer acessar. Depois usamos o resquest.get para puxar o conteúdo da página, e então foi concatenado o conteúdo da página com as colunas do dataframe ‘dfHTML’.

Figura 4 - Leitura e concatenação do conteúdo das tabelas HTML

```
23  #A partir daqui fazemos o tratamento para inserir o conteúdo das tabelas HTML dentro de cada link
24  #Método para acessar um link com uma data específica
25  def tableHMTL(date):
26      url = 'http://www.nuforc.org/webreports/ndxe' + date + '.html'
27      page = requests.get(url)
28      dfHMTL = pd.read_html(page.text)
29      dfHMTL = pd.DataFrame(np.concatenate(dfHMTL), columns=columns)
30      return dfHMTL
```

Fonte: Própria

Nesse trecho foram criadas as colunas do df ‘tablesOVNIS’ e o próprio df que ao final é utilizado para gerar o arquivo OVNIS.csv.

Figura 5 - Colunas e DataFrame ‘tableOVNIS’

```
32  #Passar dataOVNIs['Reports'] como index para chamar links do dataHTML
33  columns = ['Data/Hora','Cidade','Estado','Forma','Duracao','Descricao','Postagem']
34  tablesOVNIS = []
35  tablesOVNIS = pd.DataFrame(tablesOVNIS, columns=columns)
```

Fonte: Própria

O laço 'for’ abaixo permite acessar cada link da página de relatos de OVNIS e imprimir esses relatos. Para isso, foi usado dataOVNIS[‘Reports’] como index para as impressões.

Figura 6 - Impressão dos relatos

```
#Esse bloco acessa cada link da página de relatos de OVNIS e imprime cada relato
#passando dataOVNIs['Reports'] como index para chamar links do dataHTML
for i in indexOVNIs['Reports']:
    data_relato = str(i)
    data_split = data_relato.split("/")
    data_split.reverse()
    date = ''.join(data_split)
    tablesOVNIS = pd.concat([tableHMTL(date),tablesOVNIS])
```

Fonte: Própria

A fase da coleta termina com a criação do arquivo final 'OVNIS.csv' para fazer o armazenamento dos dados coletados pela raspagem feita na página do site NUFORC.

Figura 7 - DataFrame tablesOVNIS ('./OVNIS.csv')

```
tablesOVNIS.to_csv('OVNIS.csv')
tablesOVNIS
```

	Data/Hora	Cidade	Estado	Forma	Duracao	Descricao	Postagem
0	11/30/97 22:15	Cardinal (Canada)	ON	Cone	7 min's	My wife and I were just on our way to bed when...	12/2/00
1	11/30/97 18:00	Sacramento	CA	Flare	30 seconds	I think we saw something simular as discribed ...	6/2/98
2	11/30/97 02:00	LeMars (IA)/Denver (CO) (uncertain; traveling ...	NE	Light	approx. 5 minutes	Multi-colored "pole," appearing to hang in the...	12/2/00
3	11/29/97 20:00	Gorman (60 miles north of Los Angeles)	CA	Light	10-min.	I was coyote calling on top of a mountain at n...	6/2/98
4	11/29/97 18:15	Orcas Island	WA	Light	1 hour	a bright light in the ssw sky slowly moves dow...	6/2/98
...	...	...	...	...	...	...	...
490	10/1/17 11:25	Denver	CO	Triangle	15 minutes	A cluster of whitish balls with blue chevron w...	10/5/17
491	10/1/17 10:00	Mount pleasant	SC	Light	Still going	White light in the day sky	10/5/17
492	10/1/17 08:10	Mechanicsburg	PA	Triangle	10 minutes	3 bright orange lights in a triangular shape j...	10/5/17
493	10/1/17 04:30	DeLand	FL	Light	20 minutes	I noticed whay appeared to be one very bright ...	10/5/17
494	10/1/17 01:00	Clay City	KY	Triangle	>1 hour	V-shape or triangle or boomerang shape with li...	10/5/17

100964 rows × 7 columns

Fonte: Própria

3.3 Exploração

Para calcular o número de linhas e colunas do nosso DataFrame, foi utilizado o '.shape'.

Figura 8 - Quantidade de linhas e colunas do 'OVNIS.csv'

LISTA DE INTENS OBRIGATÓRIOS PARA A EXPLORAÇÃO

```
[58]  # 1- Saber a quantidade de linhas, observações ou variáveis que foram coletadas. -> NÚMERO DE LINHAS
      import pandas as pd

      print('A quantidade de linhas e colunas coletadas são:', tablesOVNIS.shape)
```

```
A quantidade de linhas e colunas coletadas são: (100964, 7)
```

Em seguida, foi criada uma query para selecionar os Estados da tabela tablesOVNIS e contar o número de relatos de avistamento de ovnis por cada Estado, sendo que o resultado final foi colocado em ordem decrescente.

Figura 9 - Tratamento e Intervalo do DataFrame

```
# 2- Quantos relatos ocorreram por estado em ordem decrescente?

q='''
  SELECT Estado, COUNT(*)
  FROM tablesOVNIS
  GROUP BY Estado
  HAVING COUNT(*)
  ORDER BY COUNT(*) desc
'''

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

A figura 10 traz o resultado do código acima.

Figura 10 - Tratamento e Intervalo do DataFrame

	Estado	COUNT(*)
0	CA	11527
1	None	7247
2	FL	5655
3	WA	4956
4	TX	4172
...	...	...
63	NT	21
64	YT	21
65	PE	20
66	PR	18
67	YK	5

68 rows × 2 columns

No tópico 3, foi feita a anulação de todo o index da linha na qual houvesse valor vazio, nulo ou NaN na coluna ‘Estado’.

Figura 11 - Tratamento e Intervalo do DataFrame

```
[69]
# 3- Remover possíveis campos vazios (sem estado)
tablesOVNIS.drop(tablesOVNIS.index[tablesOVNIS['Estado'] == 'None'], inplace = True)
tablesOVNIS.drop(tablesOVNIS.index[tablesOVNIS['Estado'] == None], inplace = True)

# Remover possíveis campos NaN
tablesOVNIS['Estado'].dropna()

tablesOVNIS
```

No tópico 4, foi realizado um filtro nos Estados os quais a análise estava sendo aplicada, de modo a somente os Estados dos EUA fossem analisados na pesquisa de avistamento dos OVNIS. Com o resultado trazido pelo filtro, foi criado um arquivo.csv chamado 'OVNIs_limpo.csv'

Figura 12 - Tratamento e Intervalo do DataFrame

```
[65]
# 4- Limitar a análise aos estados dos Estados Unidos.

import pandas
!pip install -U pandasql
import pandasql

def filtro_EUA(filename):

  # Tabela de Estados dos EUA
  ovnis_data = pandas.read_csv(filename)

  q='''
      SELECT *
      FROM ovnis_data
      WHERE Estado in ('AK','AL','AR','AZ','CA','CO','CT','DE','FL','GA','HI','IA','ID','IL',
      'IN','KS','KY','LA','MA','MD','ME','MI','MN','MO','MS','MT','NC','ND',
      'NE','NH','NJ','NM','NV','NY','OH','OK','OR','PA','RI','SC','SD','TN',
      'TX','UT','VT','VA','WA','WI','WV','WY')
      ORDER BY Estado
  '''
  filtro_est_USA = pandasql.sqldf(q, locals())

  OVNIS_limpo = pd.DataFrame(filtro_est_USA)
  OVNIS_limpo.to_csv("OVNIs_limpo.csv", index = False)

  return OVNIS_limpo

filtro_EUA('./OVNIS.csv')
```

A figura 13 traz o resultado do código acima

Figura 13 - Tratamento e Intervalo do DataFrame

	Unnamed: 0	Data/Hora	Cidade	Estado	Forma	Duracao	Descricao	Postagem
0	37	11/17/97 03:00	Wasilla	AK	Triangle	3 minutes	A huge trianglar object moving in a true north...	9/2/05
1	37	1/15/98 13:00	Alaska (remote)	AK	Disk	20 seconds	saucer shaped object passed directly above me	3/16/00
2	58	1/8/98 22:38	Fairbanks	AK	Oval	3 Secs	A large phlorescent green oval shaped object m...	2/16/99
3	55	3/15/98 20:30	Anchorage	AK	Oval	15 minutes	Driving home after daughters birthday, noticed...	6/18/98
4	57	3/15/98 20:00	North Pole	AK	Sphere	90 sec	Red orb, blue orb...similar to the Marfa Lights	5/24/05
...	...	...	...	...	...	...	...	...
89271	224	7/17/17 21:00	Cody	WY	Cylinder	30 seconds	Craft heading Northeast at a high rate of spee...	7/23/17
89272	293	7/13/17 04:00	Pinedale	WY	Sphere	None	Star-like orb flashing colors in night sky.	7/23/17
89273	34	10/28/17 20:00	Cody and Wapiti (between)	WY	Fireball	10 seconds	There were fireballs coming down. ((anonymous ...	11/3/17
89274	282	10/15/17 20:15	Douglas	WY	Sphere	3 minutes	2 of my friends and I were standing on my balc...	10/19/17
89275	460	10/4/17 02:30	Casper	WY	Unknown	1	7 foot tall, 'Grey', Casper, Wyoming, paid no ...	10/19/17

89276 rows × 8 columns

No tópico 5, foi realizada uma consulta ao DataFrame ‘df2’ (‘./OVNIs.limpo.csv’), que deveria selecionar todas as cidades com ao menos 10 relatos de avistamento de ovnis.

Figura 14 - Tratamento e Intervalo do DataFrame

```
[70]  # 5- Consulta por cidades, com o objetivo de saber quais contêm o maior número
        # de relatos (cidades que apresentem ao menos 10 relatos).

      df2 = pd.read_csv('./OVNIs_limpo.csv')

      q='''
        SELECT Cidade, COUNT(*)
        FROM df2
        GROUP BY Cidade
        HAVING COUNT(*) > 9
        ORDER BY COUNT(*) desc
      '''

      consulta1 = pandasql.sqldf(q, locals())
      consulta1
```

A figura 15 traz o resultado do código acima.

Figura 15 -

	Cidade	COUNT(*)
0	Phoenix	562
1	Seattle	554
2	Portland	484
3	Las Vegas	477
4	San Diego	402
...	...	...
1862	Willow Springs	10
1863	Wolf Point	10
1864	Woodville	10
1865	Yarmouth	10
1866	Yucca Valley	10

1867 rows × 2 columns

No tópico 6, foi explicado o porquê da cidade número 1 do ranking da consulta (trazida na figura 15) ter sido a cidade de Phoenix.

Figura 16 -

```
[ ]   # 6- Com o dado anterior, responder a seguinte pergunta: por que será que essa é
        #a cidade que possui mais relatos?

      '''
        Phoenix é a cidade que possui mais relatos devido a um evento denominado
        "Luzes de Phoenix" que ocorreu no estado do Arizona e Nevada (EUA) e em Sonora (MEX)
        no ano de 1997. Os fenômenos ópticos ocorridos nessa região foram visto por milhares de pessoas,
        o que ocasionou uma série de registros de avistamento de OVNIS's no BD do site do NUFORC
      '''
```

Para o tópico 7, foi criada uma query que traz todas as cidades do Estado com maior número de relatos de avistamento de OVNIS. Além disso, só deveriam ser coletadas as cidades com ao menos 10 relatos.

Figura 17 -

```
[76]
# 7- Fazer uma query exclusiva para o estado com maior número de relatos, buscando
# cidades que possuam um número superior a 10 relatos.

# Enfatizar a cidade, a quantidade de relatos e formato do objeto não identificado.

df3 = pd.read_csv('./OVNIs_limpo.csv')

q='''
  SELECT Estado, Cidade, COUNT(*), Forma
  FROM df3
  WHERE Estado = 'CA'
  GROUP BY Cidade
  HAVING COUNT(*) > 9
  ORDER BY COUNT(*) desc
'''

consulta2 = pandasql.sqldf(q, locals())
consulta2
```

A figura 18 traz o resultado do código acima.

Figura 18 -

	Estado	Cidade	COUNT(*)	Forma
0	CA	San Diego	399	Other
1	CA	Los Angeles	385	Other
2	CA	Sacramento	243	Unknown
3	CA	San Jose	225	Oval
4	CA	San Francisco	199	Triangle
...	...	...	...	...
277	CA	Seal Beach	10	Flash
278	CA	South Pasadena	10	Circle
279	CA	Union City	10	Light
280	CA	Willits	10	Other
281	CA	Yucca Valley	10	Disk

282 rows × 4 columns

4. Considerações finais

Quanto às dificuldades encontradas, o grupo G2DJMN inicialmente apresentou falhas na distribuição das tarefas entre seus membros. Isso fez com que algumas pessoas ficassem com mais tarefas do que outras, o que ocasionou uma sobrecarga e atraso no desenvolvimento da primeira fase do projeto.

Felizmente o grupo percebeu o equívoco cometido em sua organização inicial e se uniu para finalizar as demais fases da atividade juntos, de modo que os membros tiveram que integrar seus conhecimentos na resolução das demais fases do projeto.

Outra dificuldade encontrada foi em fazer a extração dos dados do site de modo eficaz e sem sobrecarregar o código. Já na fase de exploração dos dados, a equipe teve que se esforçar para entender o uso da biblioteca **pandasql** na exploração dos dados, visto que 3 dos membros do grupo não conheciam a biblioteca até então.

Quanto aos pontos positivos e à possibilidade de melhoria, a equipe está mais unida e comunicativa, mas é necessário que os membros se unam diariamente para a atualização das atividades realizadas na Sprint. Com isso, as dificuldades em cada tarefa serão identificadas mais rapidamente, assim como seu solucionamento e a entrega das Sprints no prazo.

Referências

Wikipedia.**National UFO Reporting Center**. 2021. Disponível em: <https://en.wikipedia.org/wiki/National_UFO_Reporting_Center>. **Acesso em**: 03 de agosto de 2021.

Documentação, **Pandas**. 2021. Disponível em: <https://pypi.org/project/pandasql/>. **Acesso em**: 08 de agosto de 2021

Link Arquivos do Grupo

GitHub G2. **Projeto final**: Disponível em: <https://github.com/infocbra/pratica-integrada-cd-e-am-2021-1-g2-dmjn/tree/main/Sprint 1>