Projeto de prática Integrada de ciência de dados, inteligência artificial e machine learning

Relatório de avistamento de Objetos Voadores Não Identificados.

Sprint 3 → Análise temporal

Curso: Tecnologia em sistemas para internet

Estudantes:

Brenda Lopes Miranda Teixeira
Mateus Gomes da Silva Fonteles
Rickson Queiroz Marques de Souza
Samuel Araújo Lopes

Professores

Fábio Henrique
Diego Queiroz
Ana Régia

Brasília, agosto de 2021

INSTITUTO FEDERAL
Brasília

Sumário

1. Objetivos

A análise é a etapa do trabalho em que extraímos valor dos dados, criando hipóteses com base nos *insights* gerados pelos dados. Vamos agora criar um modelo preditivo a fim de realizar uma estimativa futura baseada nas experiências prévias armazenadas no nosso dataset.
Para alcançar este resultado, aplicamos aos nossos dados um modelo de machine learning para estimar, a partir dos dados contidos em nosso dataset, a quantidade de avistamentos que haverá no futuro.

2. Descrição do problema

Para realizar o trabalho proposto, vamos levantar os dados sobre a cidade de Fênix, EUA, para logo visualizá-los em forma de série temporal, com gráficos de barras e de linha, vamos construir nossos conjuntos de treinamento e de teste e investigar os parâmetros para encontrar o melhor modelo. Por último, vamos realizar uma previsão de avistamentos futuros utilizando a função SARIMAX.

3. Desenvolvimento

Este trabalho está sendo desenvolvido usando um Script Python por ser uma linguagem orientada a objetos é bastante maleável, o grupo está utilizando a plataforma Google Colaboratory, assim todos podem modificar e acrescentar o código quando necessário.

3. Código implementado

- Importar a biblioteca Pandas
- Criar um dataframe com o arquivo .CSV

```
# Importação do panda
import pandas as pd
# Carrega seu arquivo csv
ovnis_preparado = pd.read_csv('df_OVNI_preparado.csv')
ovnis_preparado
```

	City	State	Shape	Sight_day	Sight_month	Sight_time	Sight_date	Sight_weekday
0	Solomons Island	MD	Disk	22	9	20:00:00	1997-09-22	Segunda-feira
1	Annapolis	MD	Triangle	15	8	23:00:00	1998-08-15	Sábado
2	Chesapeak Bay	MD	Disk	26	10	20:00:00	1999-10-26	Terça-feira
3	Frederick	MD	Sphere	7	7	01:45:00	2000-07-07	Sexta-feira
4	Wheaton	MD	Triangle	1	10	20:00:00	2000-10-01	Domingo
...	...	...	...	...	...	...	...	...
54938	Washington, D.C. (above I-295 bridge)	DC	Flash	13	3	05:40:00	2016-03-13	Domingo
54939	Washington, D.C.	DC	Circle	12	4	01:30:00	2016-04-12	Terça-feira
54940	Washington	DC	Triangle	8	8	22:00:00	2016-08-08	Segunda-feira
54941	Washington, D.C.	DC	Other	2	10	11:00:00	2016-10-02	Domingo
54942	Washington, DC	DC	Circle	12	5	14:30:00	2017-05-12	Sexta-feira

- Criar dataframe somente com os dados da cidade de Fênix e ordenar por data de avistamento:

```
#filtra a cidade dentro do csv
cidade_phoenix = ovnis_preparado[ovnis_preparado['City']=='Phoenix']
cidade_phoenix.sort_values(by='Sight_date')
```

	City	State	Shape	Sight_day	Sight_month	Sight_time	Sight_date	Sight_weekday
24043	Phoenix	AZ	Disk	12	6	02:30:00	1999-06-12	Sábado
24061	Phoenix	AZ	Changing	12	11	23:33:00	2001-11-12	Segunda-feira
24073	Phoenix	AZ	Cigar	31	5	13:00:00	2003-05-31	Sábado
24090	Phoenix	AZ	Light	21	2	18:00:00	2005-02-21	Segunda-feira
24091	Phoenix	AZ	Light	20	3	03:30:00	2005-03-20	Domingo
...	...	...	...	...	...	...	...	...
26019	Phoenix	AZ	Other	15	6	15:35:00	2017-06-15	Quinta-feira
26035	Phoenix	AZ	Oval	6	7	21:25:00	2017-07-06	Quinta-feira
26031	Phoenix	AZ	Fireball	26	7	04:20:00	2017-07-26	Quarta-feira
26050	Phoenix	AZ	Flash	4	8	21:15:00	2017-08-04	Sexta-feira
26048	Phoenix	AZ	Light	14	8	00:20:00	2017-08-14	Segunda-feira

305 rows × 8 columns

- Importar a biblioteca PandaSQL
- Construir uma query para contar os avistamentos por dia, ordenando pela data:

```
import pandasql
# Roda o seu comando SQL e retorna um dataframe
query = '''
  SELECT  Sight_date  ,Count(*)  as  Views  FROM  cidade_phoenix   group  by
Sight_day,  Sight_month order by Sight_date
'''
views_phoenix= pandasql.sqldf(query.lower(), locals())
views_phoenix
```

	Sight_date	views
0	2001-11-12	1
1	2003-05-31	1
2	2005-06-08	1
3	2005-10-15	2
4	2006-04-30	1
...	...	...
198	2017-06-15	2
199	2017-07-06	1
200	2017-07-26	2
201	2017-08-04	1

- Converter a coluna sight_date para o tipo data
- Consultar o tipo da coluna para confirmar o sucesso da operação
- Separar o ano da coluna data e criar uma nova coluna chamada sight_year

```
#filtra a data e o ano
views_phoenix['Sight_date']                                    =
pd.to_datetime(views_phoenix['Sight_date'])
views_phoenix.dtypes
```

```
Sight_date    datetime64[ns]
views                  int64
dtype: object
```

```
views_phoenix['Sight_year'] = views_phoenix['Sight_date'].dt.year
views_phoenix
```

	Sight_date	views	Sight_year
0	2001-11-12	1	2001
1	2003-05-31	1	2003
2	2005-06-08	1	2005
3	2005-10-15	2	2005
4	2006-04-30	1	2006
...	...	...	...
198	2017-06-15	2	2017
199	2017-07-06	1	2017
200	2017-07-26	2	2017

- Criar uma nova consulta SQL para ver os avistamentos por ano (na cidade de Fênix) e guardar o resultado em um novo dataframe:

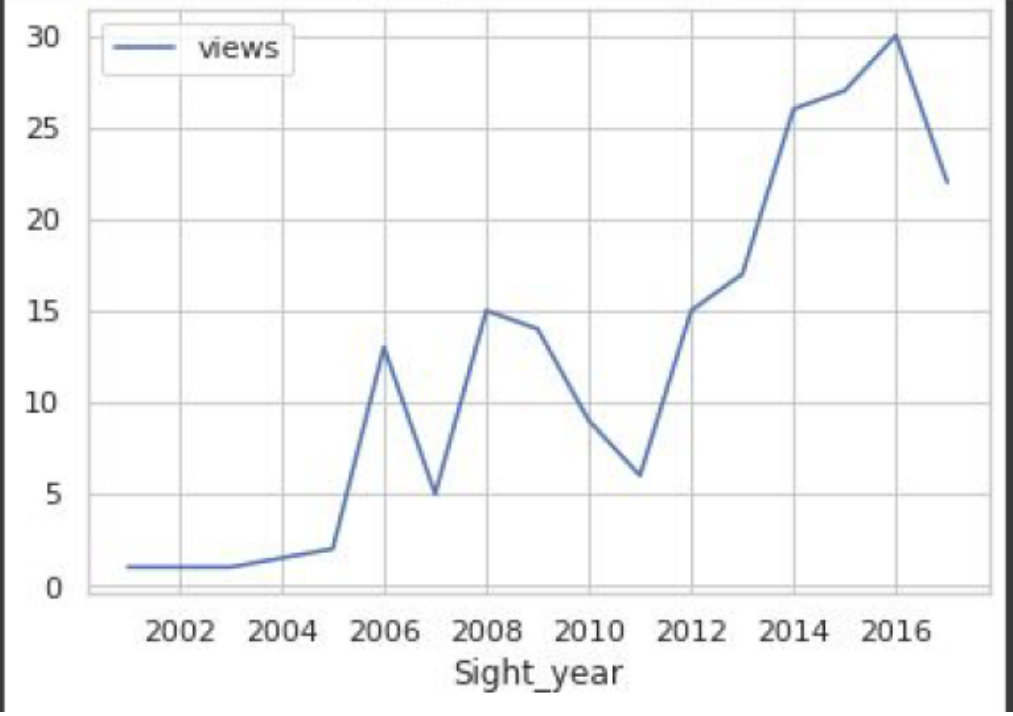
```
query = '''
  SELECT  Count(*)  as  views,  Sight_year  FROM  views_phoenix   group  by
Sight_year
'''
views_phoenix_per_year= pandasql.sqldf(query.lower(), locals())
views_phoenix_per_year
```

	views	Sight_year
0	1	2001
1	1	2003
2	2	2005
3	13	2006
4	5	2007
5	15	2008
6	14	2009
7	9	2010
8	6	2011
9	15	2012

- Gerar um gráfico para observar a quantidade de relatos por ano:

```
views_phoenix_per_year.plot.line(x='Sight_year',y='views')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f407d66c978>
```



4. Considerações Finais

Nesta etapa do trabalho visualizamos as ocorrências relatadas na cidade de Fênix, consultando por data, ano e número de ocorrências, a fim de conhecer o fluxo de ocorrências nesta localidade.
Além disso, expressamos este dado também em um gráfico de linhas.