INSTITUTO FEDERAL DE BRASÍLIA

CAMPUS BRASÍLIA

CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS PARA INTERNET

Gabriel Santos

Marcos Vasconcellos

Pedro L. B. G. de Araujo

RELATÓRIO DE PRÁTICA INTEGRADA DE CIÊNCIA DE DADOS E INTELIGÊNCIA ARTIFICIAL: TAREFA 5.7 – LIMPEZA DOS DADOS

Brasília

2020

SUMÁRIO

1. Objetivos

Partindo pelo ponto da grande importância dos campos da Inteligência Artificial e da Ciência de Dados na área da Tecnologia da Informação, possuir no mínimo um breve conhecimento e contato com tais tecnologias, mostra-se imprescindível para um verdadeiro profissional de TI.

Sobre o campo da Inteligência Artificial, de acordo com Stuart Russell; Peter Norvig (2013), a IA define-se por um comportamento que se relaciona a processos de pensamento e raciocínio. E para seu estudo e trabalho, devemos considerar quatro definições principais sobre a Inteligência Artificial, sendo estas: pensar como um humano; agir como humano; pensar racionalmente, e agir racionalmente.

Passando para a área da Ciência de Dados, Joel Grus, (2016), diz que a quantidade de dados espalhadas por todos os lados em nosso cotidiano, é extratosférica. E um cientista de dados é alguém que trabalha com dados dos mais diversos tipos e tamanhos, geralmente desorganizados, extraindo-os com o objetivo de transformá-los em conhecimento.

Apresentada uma abordagem geral sobre conceitos importantes relativos à ambas às áreas temas deste projeto, o objetivo deste documento é introduzir gradativamente o leitor, ao desafio a nós proposto pelo Instituto Federal de Brasília. Tal desafio, consiste na aplicação de diversas tarefas menores, gradativamente aplicadas ao longo do projeto, visando um objetivo final. Para alcançarmos este objetivo, utilizaremos conhecimentos e ferramentas ligados à IA e Ciência de Dados, e metodologias agéis como o Scrum.

2. Descrição do Problema

Na quarta etapa do projeto, após realização da coleta de dados relacionados à OVNIS e o desenvolvimento dos gráficos e mapas, voltaremos a trabalhar com ocorrêncais de OVNIS que aplicam apenas à estados dos Estados Unidos. Todavia, agora o objetivo é realizar uma nova limpeza no arquivo CSV gerado com estas informações.

Conforme já realizado anteriormente, a exclusão de registros com valores vazios será executada novamente, mantendo também, apenas os registros ligados aos Estados Unidos. Além disso, também faremos a exclusão de variáveis irrelevantes para a análise dos dados. Por fim, manteremos apenas os registros do tipo *Shapes* com mais de 1000 ocorrências registradas, e salvaremos o dataframe resultante em um arquivo CSV com o nome “df_OVNI_limpo”.

3. Desenvolvimento

A nossa atuação na resolução do problema proposto, inicia-se com a utilização da linguagem Python e algumas de suas bibliotecas. Segundo OLIVEIRA, Marcos (2019) “As bibliotecas e pacotes Python são um conjunto de módulos e funções úteis que minizam o uso de código em nossa vida cotidiana. Essas bibliotecas e pacotes destinam-se a uma variedade de soluções modernas.”.

Algumas bibliotecas específicas foram utilizadas como base para a construção e desenvolvimento do projeto. Estas, respectivamente denominam-se e possuem como objetivo:

- **Numpy**: É o mais popular pacote de processamento de Arrays do Python. Não apenas oferece arrays e matrizes, também as gerencia.
- **Requests**: Uma grande biblioteca HTTP que funcionca sob a licença Apache 2.0. Seu objetivo é tornar as solicitações HTTP mais responsíveis e fáceis de usar.
- **BeatifulSoup**: Consiste em uma biblioteca para extração e análise de documentos HTML e XML.
- **Pandas**: Uma das diversas bibliotecas Python voltadas para a área da Ciência de Dados. É um pacote de software Python voltado para a manipulação de estruturas de dados de forma intuitiva.
- **Matplotlib**: Biblioteca que utiliza Python Script para criar gráficos e plotagens bidimensionais, permitindo a criação de múltiplos eixos simultâneos.

3.1 Código implementado

Abaixo, serão apresentados as estruturas de códigos com a utilização de uma, ou algumas das bibliotecas acima citadas, que nos auxiliaram no desenvolvimento desta etapa.

A Figura 1, exibe à seguir a importação das bibliotecas nesta esta etapa do projeto.

```
import pandas as pd
import numpy as np
```

Figura 1 - Importação das Bibliotecas Python Utilizadas

A imagem a seguir, conta com o trecho do código responsável por carregar o arquivo CSV com o resultado da extração e organização dos dados de OVNIS, filtrando-os para estados dos EUA.

```
OVNIS_EUA = pd.read_csv('../5.4-Exploracao-dos-dados-com-SQL/OVNIS_EUA.csv')
OVNIS_EUA
```

Figura 2 – Importando o CSV de Relatos de OVNIS nos Estados Unidos

Abaixo, nas figuras à seguir, temos os código responsável identificar e remover os registros com valores nulos para *City* e *Shape*, pois *State* não possui valores nulos.

```
# identificando as colunas que possuem valores nulos
OVNIS_EUA.isnull().sum()
```

Figura 3 – Identificação das Colunas com Valores Nulos para *City* e *Shape*

```
#excluindo as linhas com valores nulos para City e Shape
# State não possui valores nulos
OVNIS_EUA_WN = OVNIS_EUA.dropna(subset=['city','shape']) # WN = WHITHOUT NULLS
OVNIS_EUA_WN
```

Figura 4 – Exclusão de Valores Nulos para *City* e *Shape*

Adiante, verificamos se os valores nulos foram realmente excluídos.

```
#testando se os valores nulos foram exluidos
OVNIS_EUA_WN.isnull().sum() # retorno AttributeError: 'tuple' object has no attribute 'isnull'
```

Figura 5 – Verificação da Exclusão de Valores Nulos para *City* e *Shape*.

Após a exclusão dos valores nulos para as colunas *City* e *Shape*, geramos um CSV chamado "OVNIS_EUA_WN", para que possamos trabalhar com um arquivo CSV mais específico para que o necessitamos.

```
#exportando o dataframe com os valores nulos tratados
OVNIS_EUA_WN.to_csv('OVNIS_EUA_WN.csv', index=False)
```

Figura 6 – CSV com o Resultado da Exclusão de Nulos das Colunas *City* e *Shape*

Para a próxima etapa, identificaremos e exluiremos os valores desconhecidos para as linhas *City*, *State* e *Shape*. As imagens à seguir representam a construção do código responsável por tal.

```
#indentificando valores unknown em City, State e Shape
unknown = ((OVNIS_EUA_WN['city'] == 'Unknown') | (OVNIS_EUA_WN['state'] == 'Unknown')
           | (OVNIS_EUA_WN['shape'] == 'Unknown'))
UNKNOWN = ((OVNIS_EUA_WN['city'] == 'UNKNOWN') | (OVNIS_EUA_WN['state'] == 'UNKNOWN')
           | (OVNIS_EUA_WN['shape'] == 'UNKNOWN'))
# OVNIS_EUA_WN[unknown]
# OVNIS_EUA_WN[UNKNOWN]
```

Figura 7 – Identificação das Linhas com Valores Desconhecidos para *City*, e *Shape*

```
#removendo unknown do dataframe
# WU = WITHOUT UNKNOWNS
OVNIS_EUA_WU = OVNIS_EUA_WN.drop(index=OVNIS_EUA_WN[unknown].index)
# OVNIS_EUA_WU
OVNIS_EUA_WU = OVNIS_EUA_WU.drop(index=OVNIS_EUA_WU[UNKNOWN].index)
OVNIS_EUA_WU
```

Figura 8 – Exclusão dos Valores Desconhecidos para as Linhas *City*, e *Shape*

Abaixo, na Figura 9, realizaremos a exportação do resultado obtido até o momento, para um arquivos CSV chamado "OVNIS_EUA_WU".

```
#exportando o dataframe com os valores unknown tratados
OVNIS_EUA_WU.to_csv('OVNIS_EUA_WU.csv', index=False)
```

Figura 9 – CSV com o Resultado da Exclusão de Desconhecidos para as Linhas *City*, e *Shape*

Após a exclusão dos valores nulos e desconhecidos para as linhas e colunas de *City*, *Shape* e *State*, filtraremos os registros para apenas estados dos Estados Unidos.

```
#Lista dos 51 estados dos Estados Unidos
states_eua = ['AK','AL','AR','AZ','CA','CO','CT','DE','DC','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','VA','VT','WA','WI','WV','WY']

lista_eua = []

for i in range(len(OVNIS_EUA_WU)):
    if(OVNIS_EUA_WU.iloc[i,2] in states_eua):
        lista_eua.append(OVNIS_EUA_WU.iloc[i])

OVNIS_EUA_51 = pd.DataFrame(lista_eua).reset_index(drop= True)
OVNIS_EUA_51
```

Figura 10 – Filtrando os Registros para Estados dos Estados Unidos

Logo após, fazemos uma nova exportação do resultado obtido. Para tanto, a Figura 11 mostra o código para a exportação do resultado, para um arquivo CSV chamado “OVNIS_EUA_51”.

```
#exportando o dataframe com os valores dos 51 estados
OVNIS_EUA_51.to_csv('OVNIS_EUA_51.csv', index=False)
```

Figura 11 – CSV Resultante da Filtragem para Estados dos Estados Unidos

À seguir, precisamos fazer a remoção de colunas não tão relevantes para o resultado final que buscamos. As colunas removidas, serão: *Duration*, *Summary* e *Posted*.

```
#removendo as colunas Duration, Summary e Posted do Dataframe
# DC = DROP COLUMNS
OVNIS_EUA_51_DC = OVNIS_EUA_51.drop(labels=['duration', 'summary', 'posted'], axis=1)
OVNIS_EUA_51_DC
```

Figura 12 – Exclusão das Colunas *Duration*, *Summary* e *Posted*

Abaixo, exportamos o resultado para um arquivo CSV chamado “OVNIS_EUA_51_DC”.

```
#exportando o csv sem as colunas duration, summary e posted
OVNIS_EUA_51_DC.to_csv('OVNIS_EUA_51_DC.csv', index=False)
```

Figura 13 – CSV Resultante da Exclusão das Colunas *Duration*, *Summary* e *Posted*

Por fim, manteremos apenas os registros do tipo *Shape* mais populares, ou seja, com mais de 1000 ocorrências, exportando-o para um arquivo CSV.

A imagem referente à Figura 13, exibe a filtragem dos registros menos populares, ou seja, com inferiores a 1000 ocorrências.

```
#Filtrando as formas menos populares para exclusão das linhas do dataframe
shapes_menos_populares = pd.value_counts(OVNIS_EUA_51_DC['shape']).to_frame().reset_index()
shapes_menos_populares.columns = ['forma', 'quantidade']
shapes_menos_populares_excluir =  shapes_menos_populares.query("quantidade <=1000")
shapes_menos_populares_excluir
```

Figura 14 – Filtragem de Registros do Tipo *Shape* Inferiores à 1000 Ocorrências

Realizada a filtragem, foram identificados que os registros *Shape* inferiores à 1000 ocorrências, são: *Cylinder*; *Diamond; Shape*; *Chevron*; *Teardrop*; *Egg*; *Cone*, e *Cross.* Portanto, é necessário que façamos a exclusão destes registros. Para tanto, temos a Figura 15 e Figura 16 à seguir.

```
#indentificando  as formas Cylinder, Diamond, Chevron, Teardrop, Egg, Cone e Cross
formas_excluir = ((OVNIS_EUA_51_DC['shape'] == 'Cylinder') | (OVNIS_EUA_51_DC['shape'] == 'Diamond')  | (OVNIS_EUA_51_DC['shape'] == 'Chevro
n')  |                         (OVNIS_EUA_51_DC['shape'] == 'Teardrop') | (OVNIS_EUA_51_DC['shape'] == 'Teardrop') | (OVNIS_EUA_51_DC['shap
e'] == 'Egg') |
                    (OVNIS_EUA_51_DC['shape'] == 'Cone') | (OVNIS_EUA_51_DC['shape'] == 'Cross'))

OVNIS_EUA_51_DC[formas_excluir]
```

Figura 15 – Identificação de Registros do Tipo *Shape* Inferiores à 1000 Ocorrências

```
#Excluindo as linhas do dataframe das formas menos populares
#removendo as formas Cylinder, Diamond, Chevron, Teardrop, Egg, Cone e Cross do dataframe
# WS = WITHOUT SHAPES MENOS POPULARES (<= 1000)
OVNIS_EUA_51_DC_WS = OVNIS_EUA_51_DC.drop(index=OVNIS_EUA_51_DC[formas_excluir].index)

OVNIS_EUA_51_DC_WS
```

Figura 16 Exclusão de Registros do Tipo *Shape* Inferiores à 1000 Ocorrências

Conforme informado acima, após a execução de todos os itens propostos nesta etapa do projeto, exportaremos o resultado final para um arquivo CSV chamado “df_OVNI_limpo”.

4. Considerações Finais

Com a conclusão de todas as etapas de trabalho, o resultado obtido nos gerou um arquivo do tipo CSV, que apresenta todos os registros de OVNIS dos Estados Unidos. Todavia, com os filtros aplicados, esta apresentação de registros limita-se à registros do tipo *Shape* com mais de 1000 ocorrências. Os códigos e o arquivos CSV desenvolvidos no decorrer desta etapa, encontram-se no no repositório Git Hub do grupo, no endereço https://github.com/Prof-Fabio-Henrique/pratica-integrada-icd-e-iia-2020-1-g5-gmp/tree/master/5.7-Limpeza-de-dados.

Para tanto, o uso da linguagem Python e suas bibliotecas foi essencial. Através da integração de ambos, foi possível gerar a tabela e organizar as informações coletadas com o uso de colunas, separando as diferentes informações por data e hora, tipo, descrição do ocorrido, entre outros. Com isso, é facilmente notada a importância do campo da Ciência de Dados, que cumpre exatamente com o que promete: coletar dados espalhados em grandes quantidades, e gerar informações úteis com estes dados. E isso, em uma visão generalizada, consegue mudar a forma com que o mundo e suas informações são vistas. Trazendo uma maior facilidade na visualização e entendimento dos mais diversos dados, assim como um maior valor para os mesmos.

Projetos completos e de real valor para o mercado como este que se inicia (que por enquanto se conclui), tendem apenas à agregar positivamente no conhecimento e experiência prática dos estudantes dos cursos de Tecnologia da Informação do Instituto Federal de Brasília. Tais conhecimentos práticos, servem tanto para dar uma base de como um profissional atua no mercado, com os mais diversos tipos de projetos e demandas, assim como para prepará-los para o tal mercado, quiçá para a montagem de um portfólio.

5. Referências

RUSSELL Stuart; NORVIG Peter, **Inteligência Artificial, 3ª edição**. Rio de Janeiro: Elsevier, 2013.

GRUS, Joel. ***Data Science* do Zero: Primeiras Regras com Python**. Rio de Janeiro: Altabooks, 2016.

OLIVEIRA, Marcos. TERMINAL ROOT. **As 30 Melhores bibliotecas e pacotes Python para Iniciantes**. 2019. Disponível em: https://terminalroot.com.br/2019/12/as-30-melhores-bibliotecas-e-pacotes-python-para-iniciantes.html. Acesso em: 18 de Setembro de 2020.