PROJETO: “BIG DATA ANALYTICS - ANÁLISE DE CLASSIFICAÇÃO DE FILMES”

OBJETIVO

O objetivo é usar técnicas de Data Analysis e Big Data para analisar o dataset MovieLens e propor melhorias no sistema de recomendação, além de entender as preferências dos usuários e o desempenho dos filmes.

EXECUÇÃO

Etapa preliminar:

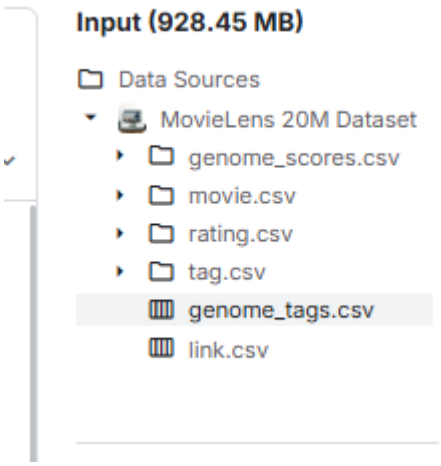
1. Crie sua conta na ferramenta GitHub;
2. Crie um repositório com o nome: MEUS PROJETOS;
3. Crie sua conta na ferramenta Google Colab;

*Use sua conta Google para ambos

Etapa Coleta de dados:

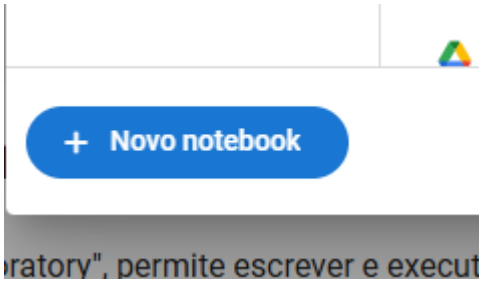
4. Baixe os arquivos no sistema Kaggle no endereço abaixo: https://www.kaggle.com/code/cesarcf1977/movielens-data-analysis-beginner-s-first/input

5. Faça download dos arquivos movie.csv e rating.csv apenas. Basta clicar em cima de cada um dele e escolar a opção de download:

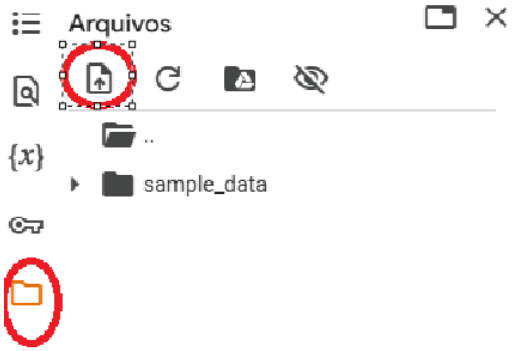


6. Após baixar para sua máquina, descompacte os arquivos para que você suba no Google Colab o arquivo em CSV e não em ZIP.

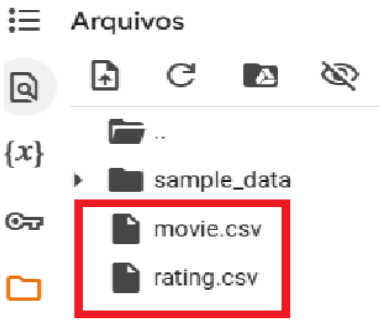
7. Abra o Google Colab e crie novo Notebook:



8. Suba os dois arquivos na pastinha através do botão Download:



9. Confira ser os arquivos estão na pasta:



Etapa processamento de dados:

10. Acesse novamente o GitHub e copie o código no aquivo nomeado como "CodigoAvaliaFilmes-fase1", por trecho explicado e realize as ações que estão sendo pedidas, conforme a evolução dos blocos (vide figura)

```
[1] from timeit import default_timer
    start = default_timer()
    import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    plt.style.use('ggplot')
```

```
[2] st = default_timer()

    # First time data load.
    movies = pd.read_csv('/content/movie.csv')
    ratings = pd.read_csv('/content/rating.csv')
```

```
movies.sort_values(by='movieId', inplace=True)
movies.reset_index(inplace=True, drop=True)
ratings.sort_values(by='movieId', inplace=True)
ratings.reset_index(inplace=True, drop=True)

print(ratings.dtypes)

# Split title and release year in separate columns in movies dataframe. Convert year to timesta
movies['year'] = movies.title.str.extract("\((\d{4})\)", expand=True)
movies.year = pd.to_datetime(movies.year, format='%Y')
movies.year = movies.year.dt.year # As there are some NaN years, resulting type will be float (
movies.title = movies.title.str[:-7]

# Categorize movies genres properly. Working later with +20MM rows of strings proved very resou
genres_unique = pd.DataFrame(movies.genres.str.split('|').tolist()).stack().unique()
genres_unique = pd.DataFrame(genres_unique, columns=['genre']) # Format into DataFrame to store
movies = movies.join(movies.genres.str.get_dummies().astype(bool))
movies.drop('genres', inplace=True, axis=1)

# Modify rating timestamp format (from seconds to datetime year)
#ratings.timestamp = pd.to_datetime(ratings.timestamp, unit='s')
ratings.timestamp = pd.to_datetime(ratings.timestamp, infer_datetime_format=True)
ratings.timestamp = ratings.timestamp.dt.year

# Check and clean NaN values
print ("Number of movies Null values: ", max(movies.isnull().sum()))
print ("Number of ratings Null values: ", max(ratings.isnull().sum()))
movies.dropna(inplace=True)
ratings.dropna(inplace=True)

# Organise a bit, then save into feather-formatand clear from memory
movies.sort_values(by='movieId', inplace=True)
ratings.sort_values(by='movieId', inplace=True)
movies.reset_index(inplace=True, drop=True)
ratings.reset_index(inplace=True, drop=True)

runtime = default_timer() - st
print ("Elapsed time(sec): ", round(runtime,2))
```

```
userId         int64
movieId      float64
rating       float64
timestamp     object
dtype: object
<ipython-input-3-f038f93b6e0a>:22: UserWarning: The argument 'infer_datetime_format' is deprecat
  ratings.timestamp = pd.to_datetime(ratings.timestamp, infer_datetime_format=True)
Number of movies Null values:  22
Number of ratings Null values:  1
Elapsed time(sec):  116.64
```

Etapa Análise Exploratória de dados:

11. Continue realizando os comandos indicados no GitHub usando os códigos contidos no arquivo: CodigoAvaliaFilmes-fase2

Etapa Visualização de Dados:

12. Agora, você irá montar seu relatório no Word e na área de Execução você irá colocar todos os gráficos criados por você na execução dos códigos de análise e estruturação dos dados.

13. Copie os gráficos e as informações que achar importante. Pode complementar com pontos de vistas das análises que você observou

Etapa Relatório Final e vídeo:

14. Construa o relatório com uma introdução (informações da execução do projeto), Execução (com todos os dados pedidos nos itens 12 e 13) e conclusão (com sua análise final de todo o projeto)

15. Grave um vídeo de até 3 minutos no formato MP4

16. Envie os dois arquivos para o Tutor avaliar.

17. Salve este documento, o relatório e o vídeo no seu GitHub para ficar como um portfólio seu.