**PROJECT TITLE: PREDICTING IMDB SCORES**

**(PHASE 3)**

**Team Members:**

* *GEETIKA.R-715521106014*
* *MIRUTHULA.P.V-715521106029*
* *SRI KARTHICK RAJAN.N-715521106050*
* *THANUJ.R-715521106052*
* *THARUN SRIPAL.A.D-715521106053*
* *HARSHAN.B-715521106304*

**PROBLEM STATEMENT:**

The problem at hand is to develop a machine learning model that can accurately predict the IMDb scores of movies based on several key attributes, including genre, premiere date, runtime, and language. IMDb scores represent the perceived quality and popularity of movies, making this prediction task valuable for assisting users in discovering high-rated films that align with their preferences.

**DATASET:**

Data set link: <https://www.kaggle.com/datasets/luiscorter/netflix-original-films-imdb-scores/>

The data set includes details about the film's title, genre, Premiere, runtime (the length of the film), IMDb ratings, and language.

**SOURCE CODE:**

**#Import Library**  
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import matplotlib.lines as lines

import matplotlib.gridspec as gridspec

import seaborn as sns

**#Data Loading**

ds = pd.read\_csvds = pd.read\_csv("NetflixOriginals.csv",encoding = "ISO-8859-1")

ds\_date = ds.copy()

ds.head(5)

**#Data Description**

ds.describe().T

ds.info(verbose=True,show\_counts=True)

ds.isna().sum()

ds['Title'].value\_counts()

ds['Genre'].value\_counts()

ds['Premiere'].value\_counts()

**#Data Preprocessing**

**# Converting object to date**

from datetime import datetime

ds\_date["Premiere"] = ds\_date["Premiere"].apply(lambda x: "".join(x for x in x.replace(".",",")))

ds\_date["PremiereDate"] = ds\_date["Premiere"].apply(lambda x: datetime.strptime(x, "%B %d, %Y").date())

ds\_date["Year"] = ds\_date["Premiere"].apply(lambda x: "".join(x for x in x.replace(",","").split()[-1]))

ds\_date["PremiereDate"] = pd.to\_datetime(ds\_date["PremiereDate"])

ds\_date

ds\_date.info()

ds['Language'].value\_counts()

ds['Genre'].value\_counts()

genre = ds['Genre'].value\_counts()

genre.head()

ds['Language'].value\_counts()

ds\_english = ds[ds['Language'] == 'English'].sort\_values('IMDB Score', ascending=False)

ds\_english.head()

**#Data Splitting**

from sklearn.model\_selection import train\_test\_split

X = ds\_date[['Title','Genre','Premiere','Runtime','Language']] # Replace with your features

y = ds\_date['IMDB Score'] # Replace with your target variable

**# Split the data into training and testing**

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

print('X\_train',X\_train)

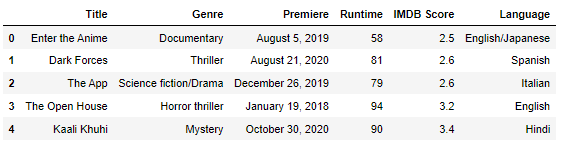
print('X\_test',X\_train)

print('y\_train',X\_train)

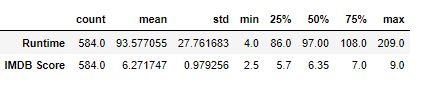
print('y\_train',X\_train)

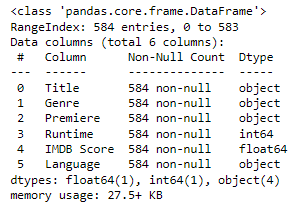
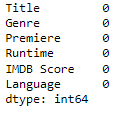
**OUTPUT:**

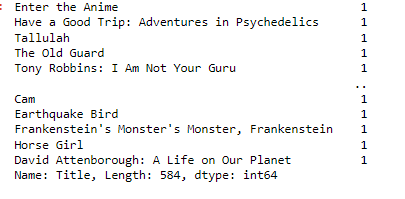
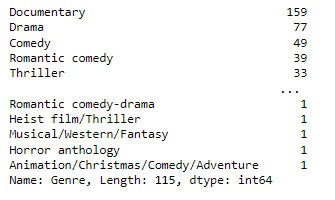
**Data Loading**

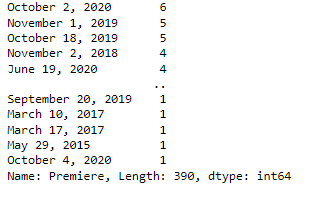
****

**Data Description**

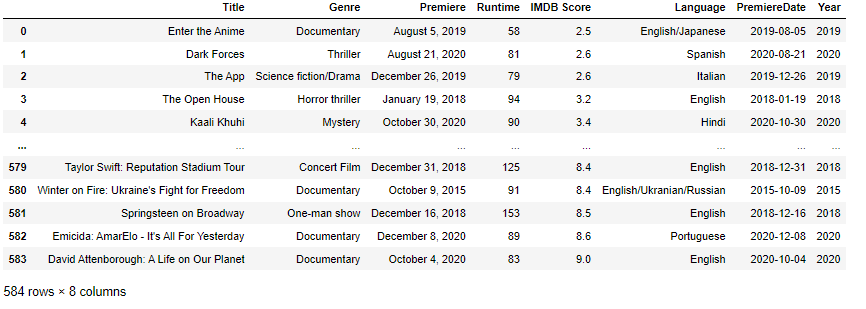
****

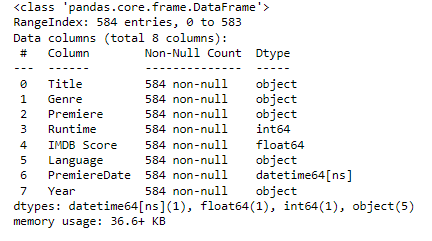
**** ****

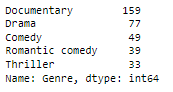


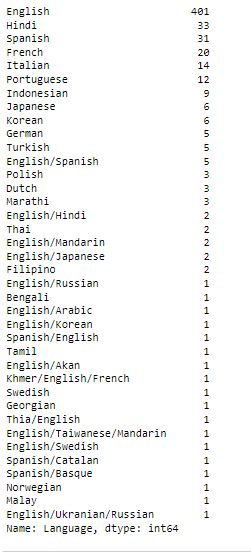
**Data Preprocessing**













**Data Splitting**

