**IMDb Score Prediction using Data Science…**Data Science

**TEAM MEMBER: Priyadharshini.R**

**NAAN MUTHALVAN ID au723721205306**

**Introduction:**

**Problem Definition:** IMDb scores are determined by user ratings and can change over time as more users rate the movie or show.

The problem is to develop a machine learning model to predict the IMDb scores of movies available on Films based on their genre, premiere date, runtime, and language.

This project involves data collection, data prepossessing, feature engineering, clustering algorithms, visualization, and interpretation of results.

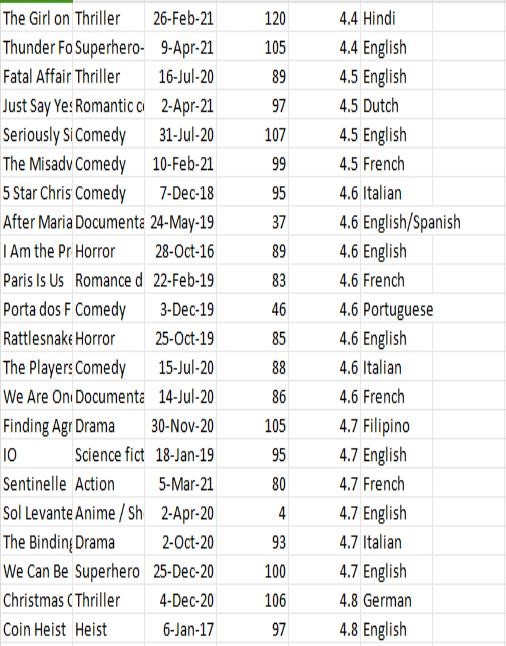
The model aims to accurately estimate the popularity of movies to assist users in discovering highly rated films that align with their preferences.

**Project Phase 2:** Develop a machine learning model to predict the IMDb scores of movies available on Films based on their genre, premiere date, runtime, and language.

**Dataset:**

Datasetlink :https://www.kaggle.com/datasets/luisc orter/netflix-original-films-imdb-scores/





**Program :**

import numpy as np import pandas as pd import os for dirname, \_, filenames in os.walk('/kaggle/input'):

for filename in filenames:

print(os.path.join(dirname, filename))

import matplotlib.pyplot as plt import seaborn as sns import plotly.express as px from datetime import datetime, timedelta ds = pd.read\_csv("/kaggle/input/netflix-original films-imdb-scores/NetflixOriginals.csv",encoding

= "ISO-8859-1") ds\_date = ds.copy() ds.head(5)



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() ds\_date["Premiere"]=ds\_date["Premiere"].apply(la mbda x: "".join(x for x in x.replace(".",","))) ds\_date["PremiereDate"]=ds\_date["Premiere"].appl y(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() plt.figure(figsize=(16, 5))

ds['Genre'].value\_counts().head(10).plot(kind='bar', color='red') plt.xlabel('Genre') plt.ylabel('Number of Genre') plt.xticks(rotation=90) plt.show(block=True)

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

ds\_lang = ds['Language'].value\_counts() ds\_lang.head(5).plot(kind='bar', color='red') plt.xlabel('Languge') plt.ylabel('Number of Language') plt.show(block=True) ds.groupby('Language').agg({'Runtime': 'sum'}).sort\_values('Runtime',

ascending=False).head(5).plot(kind='bar',color='red

')

plt.xlabel('Language') plt.ylabel('Runtime') plt.show(block=True) ds\_english =

ds[ds['Language']=='English'].sort\_values('IMDB

Score', ascending=False) ds\_english.head() ds\_date.groupby('Year').agg({'Runtime': 'sum'}).sort\_values('Runtime', ascending=False).plot(kind='bar', color='red')

plt.xlabel('Year') plt.ylabel('Sum of Runtime') plt.show(block=True) ds\_date.groupby('Year').agg({'Title': 'count'}).sort\_values('Title',

ascending=False).plot(kind='bar', color='red')

plt.xlabel('Year') plt.ylabel('Number of Film') plt.show(block=True)

Output:

English 401

Hindi 33

Spanish 31

|  |  |
| --- | --- |
| French | 20 |
| Italian | 14 |
| Portuguese | 12 |
| Indonesian | 9 |
| Japanese | 6 |
| Korean | 6 |
| German | 5 |
| Turkish | 5 |
| English/Spanish | 5 |
| Polish | 3 |
| Dutch | 3 |
| Marathi | 3 |
| English/Hindi | 2 |
| Thai | 2 |
| English/Mandarin | 2 |
| English/Japanese | 2 |
| Filipino | 2 |
| English/Russian | 1 |
| Bengali | 1 |
| English/Arabic | 1 |
| English/Korean | 1 |
| Spanish/English | 1 |
| Tamil | 1 |
| English/Akan | 1 |

Khmer/English/French 1

Swedish 1

Georgian 1

Thia/English 1

English/Taiwanese/Mandarin 1

English/Swedish 1 Spanish/Catalan 1

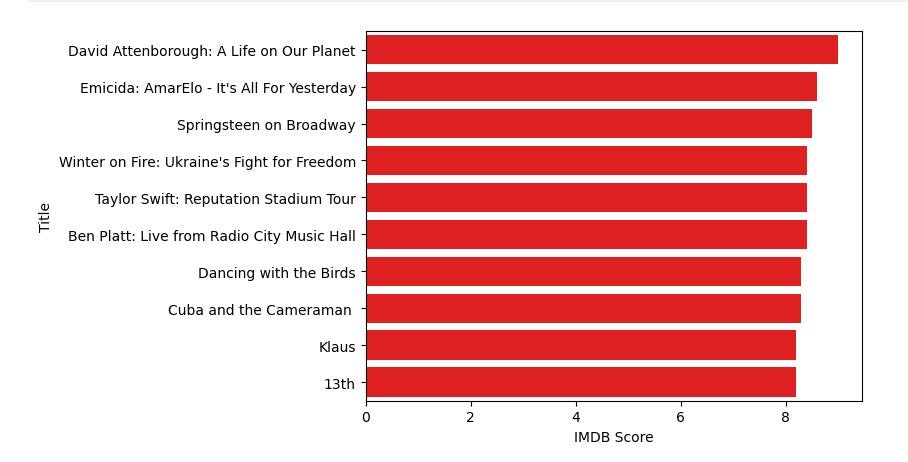
Spanish/Basque 1

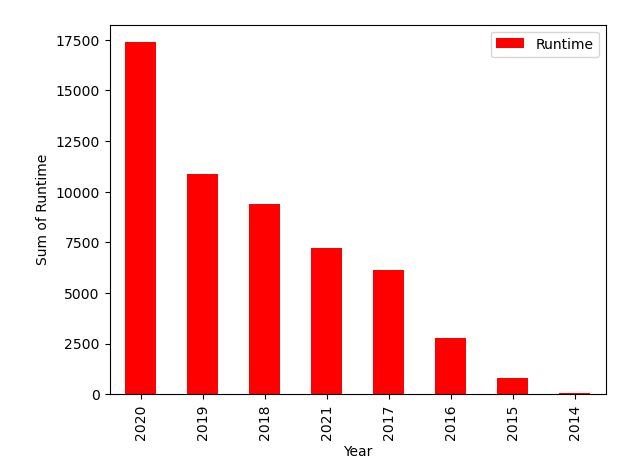
Norwegian 1

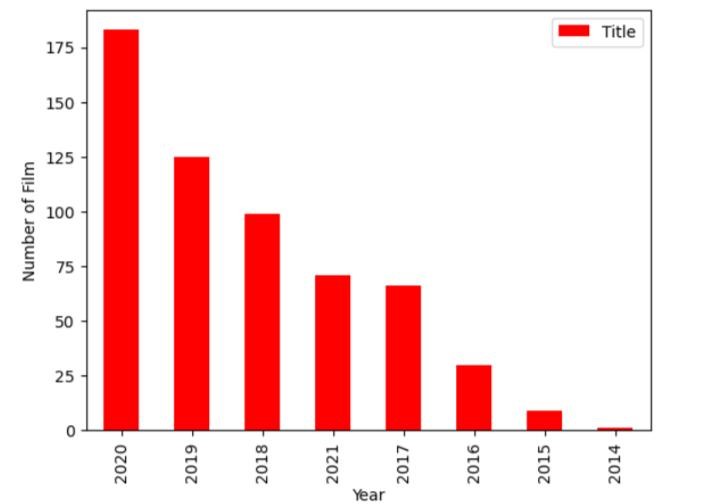
Malay 1

English/Ukranian/Russian 1

Name: Language, dtype: int64







**Conclusion :**

In conclusion, predicting IMDb scores

is a complex task that involves various factors and challenges.IMDb scores are influenced by a multitude of subjective and contextual factors, and no model can perfectly capture all of these nuances.

To improve IMDb score predictions,

it's crucial to consider factors such as user reviews, genre, director, actors, and release date, among others. However, it's essential to remember that IMDb scores are ultimately a reflection of audience opinions, and these opinions can change over time. Therefore, any prediction model should be periodically updated and validated against new data.