**PREDICTING IMDB SCORES**

**INTRODUCTION:**

Creating a model to predict IMDb ratings involves steps. Initially we focus on loading and preparing the dataset. WE are particularly interested in broad measures such as mean squared error(MSE), and R- squared(R2). We tested the accuracy and durability of our models using these criteria.

The ultimate goal of building an IMDb rating prediction model is to accurately predict the IMDb ratings of movies based on various features and attributes. The initial steps involve loading and preprocessing the dataset to prepare it for machine learning, which is crucial for creating a reliable predictive model.

Here's an overview of the process;

**DATASET:**

**DATA SOURCE:** <https://www.kaggle.com/datasets/luiscorter/netflix-original-films-imdb-scores>



**Gathering Data:**

Collect IMDb movie data, including information, like genre, cast, reviews and IMDb ratings.

**Data Cleanup:**

Address any missing values, outliers or inconsistencies in the dataset.

**Feature Engineering:**

Generate features from the data, such as analyzing sentiment in reviews or encoding genres using one hot encoding.

**Data Splitting:**

Divide the dataset into training and testing sets to assess how well the model performs.

**Normalization/Scaling:**

Standardize or normalize numerical features to ensure consistency.

**Model Selection:**

Choose a machine learning or deep learning model, for predicting ratings.

**Model Training:**

Train the selected model using the training data.

**Model Evaluation:**

Evaluate how well the model performs using metrics (RMSE for regression tasks) and test dataset.

**Tuning:**

Make adjustments, to the models hyperparameters in order to enhance its performance.

**Prediction:**

Utilize the trained model to make IMDb rating predictions, for movies that're new or haven't been seen before.

**Deployment:**

If needed put the model into use.

**Maintenance:**

Regularly,maintain the model as new data becomes accessible.

**PROGRAM :**

**import pandas as pd**

**import numpy as np**

**import matplotlib.pyplot as plt**

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

**from sklearn.linear\_model import LinearRegression**

**from sklearn.metrics import mean\_squared\_error, r2\_score**

**from sklearn.preprocessing import LabelEncoder**

**from datetime import datetime**

**# Load the dataset**

**data = pd.read\_csv('/content/NetflixOriginals.csv', encoding='ISO-8859-1')**

**# Select relevant columns**

**selected\_columns = ["Title", "Genre", "Premiere", "Runtime", "IMDB Score", "Language"]**

**data = data[selected\_columns]**

**# Convert Premiere column to datetime**

**data["Premiere"] = pd.to\_datetime(data["Premiere"])**

**# Extract year from the Premiere date**

**data["Premiere\_Year"] = data["Premiere"].dt.year**

**# Encode categorical variables (Genre and Language) using Label Encoding**

**label\_encoder = LabelEncoder()**

**data["Genre\_Code"] = label\_encoder.fit\_transform(data["Genre"])**

**data["Language\_Code"] = label\_encoder.fit\_transform(data["Language"])**

**# Drop the original Genre and Language columns**

**data = data.drop(["Genre", "Language", "Premiere"], axis=1)**

**# Handle missing data (if any)**

**data.dropna(inplace=True)**

**# Define X (features) and y (target)**

**X = data[["Premiere\_Year", "Runtime", "Genre\_Code", "Language\_Code"]]**

**y = data["IMDB Score"]**

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

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

**# Create and train a Linear Regression model**

**model = LinearRegression()**

**model.fit(X\_train, y\_train)**

**# Make predictions on the test set**

**y\_pred = model.predict(X\_test)**

**# Evaluate the model**

**mse = mean\_squared\_error(y\_test, y\_pred)**

**r2 = r2\_score(y\_test, y\_pred)**

**print(f"Mean Squared Error: {mse}")**

**print(f"R-squared: {r2}")**

**# Data Visualization**

**# Scatter plot of actual vs. predicted IMDb scores**

**plt.figure(figsize=(8, 6))**

**plt.scatter(y\_test, y\_pred)**

**plt.title('Actual IMDb Score vs. Predicted IMDb Score')**

**plt.xlabel('Actual IMDb Score')**

**plt.ylabel('Predicted IMDb Score')**

**plt.show()**

**# Residual plot to check for homoscedasticity**

**residuals = y\_test - y\_pred**

**plt.figure(figsize=(8, 6))**

**plt.scatter(y\_pred, residuals)**

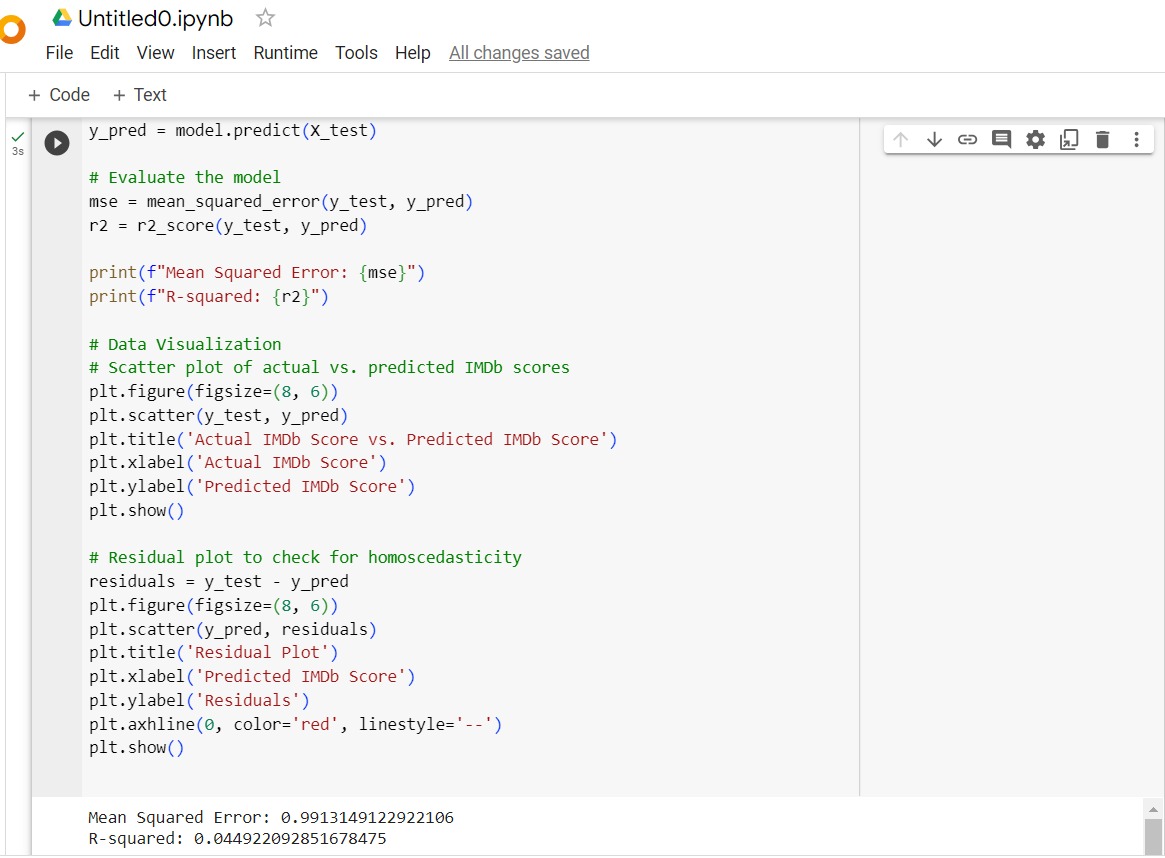
**plt.title('Residual Plot')**

**plt.xlabel('Predicted IMDb Score')**

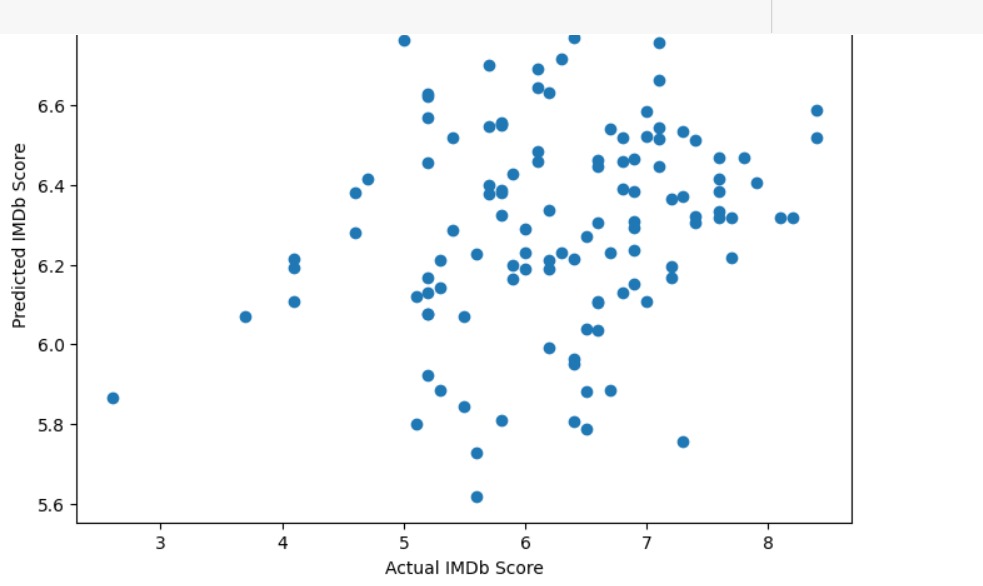
**plt.ylabel('Residuals')**

**plt.axhline(0, color='red', linestyle='--')**

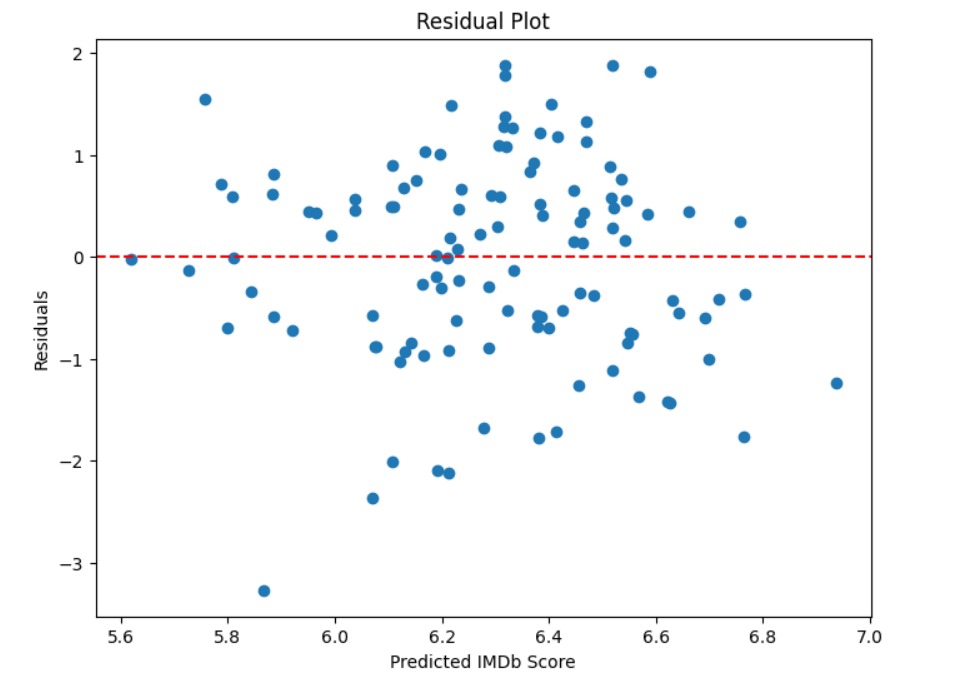
**plt.show()**

**OUTPUT :**

**DATA VISUALIZATION:**

**** Scatter plot: Actual vs predicted IMDb scores

Scatter plot: Residuals vs predicted IMDB scores

****

**CONCLUSION**:

In this IMDB scores study,we used regression models to forecast IMDB scores based on dataset features. R-squared,Mean Squared Error(MSE),were used to evaluate each model and the efficacy of our IMDb rating prediction model relies on the quality of data and the selection of modeling techniques. To improve predictions in real applications, more refining and feature engineering might be investigated.