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

1 to 10 of 584 entries Filter

Title	Genre	Premiere	Runtime	IMDB Score	Language
Enter the Anime	Documentary	August 5, 2019	58	2.5	English/Japanese
Dark Forces	Thriller	August 21, 2020	81	2.6	Spanish
The App	Science fiction/Drama	December 26, 2019	79	2.6	Italian
The Open House	Horror thriller	January 19, 2018	94	3.2	English
Kaali Khuhi	Mystery	October 30, 2020	90	3.4	Hindi
Drive	Action	November 1, 2019	147	3.5	Hindi
Leyla Everlasting	Comedy	December 4, 2020	112	3.7	Turkish
The Last Days of American Crime	Heist film/Thriller	June 5, 2020	149	3.7	English
Paradox	Musical/Western/Fantasy	March 23, 2018	73	3.9	English
Sardar Ka Grandson	Comedy	May 18, 2021	139	4.1	Hindi

Show 10 per page

1 2 10 50 59

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 :

Untitled0.ipynb

File Edit View Insert Runtime Tools Help All changes saved

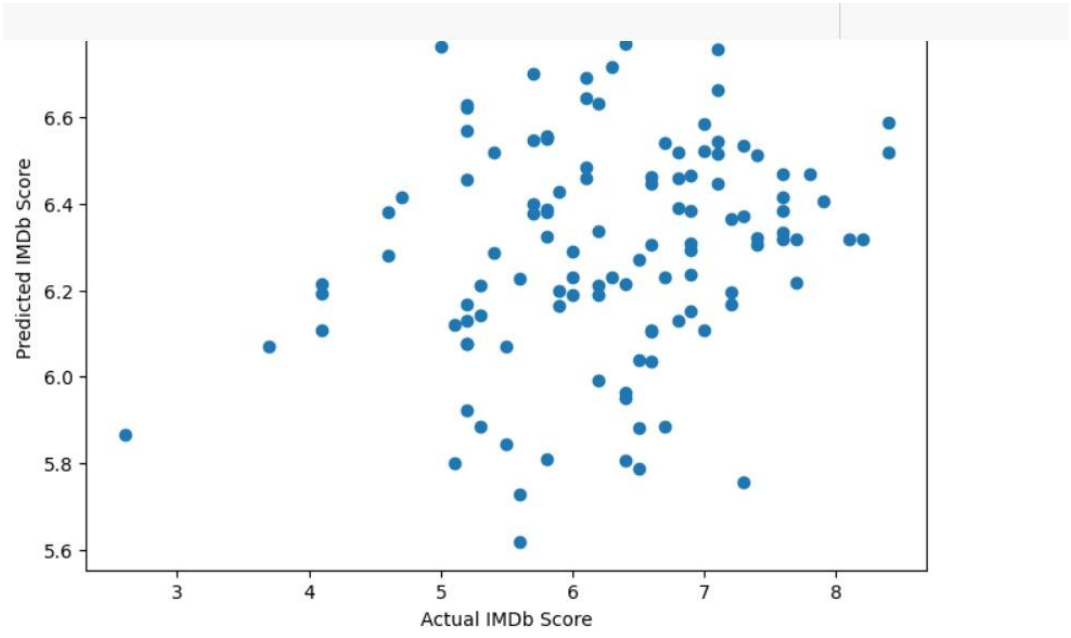
+ Code + Text

```
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()
```

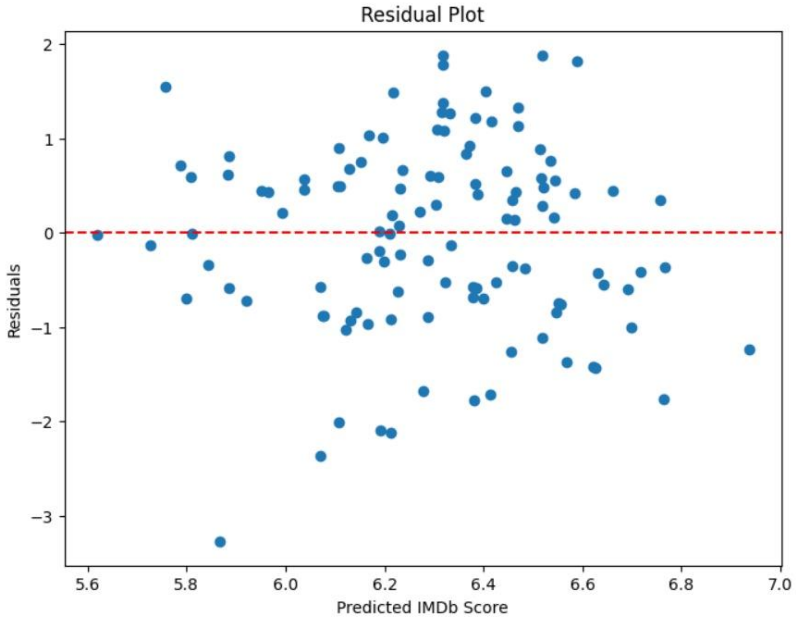
```
Mean Squared Error: 0.9913149122922106
R-squared: 0.044922092851678475
```

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.