IMDB SCORE PREDICTION

STATEMENT :

The goal of the IMDb Score Prediction project is to create a machine learning model that can accurately predict IMDb (Internet Movie Database) movie scores based on various features and attributes. IMDb scores are an important measure of a movie's quality and popularity, and accurately predicting them can be valuable to filmmakers, studios, and movie fans. This project focuses on the regression of IMDb scores as predictors of continuous variables.

DATASET LINK :

https://www.kaggle.com/datasets/luiscorter/netflix-original-films-imdb-scores

DESIGN THINKING PROCESS:

1. UNDERSTANDING THE PROBLEM : The project begins with a thorough understanding of the problem statement. We want to predict the IMDb score for a movie, and this requires a comprehensive analysis of the factors that affect a movie's score.

2. DATA COLLECTION: Film genre, budget, director, actor, year of release, etc. We have collected a dataset that includes many features such as This information is obtained from reliable sources and will be the basis for the development and evaluation of our model.

3. DATA PROCESSING: Raw data is processed to handle missing values, categorical variable codes, and increase specificity if necessary. This ensures that the data is in a suitable format for model training.

4. SELECTION MODEL : Since we are predicting a continuous objective variable (IMDB score), we selected a regression algorithm for this project. The selection of specific algorithms is based on the evaluation of experimental and model performance.

5. MODEL TRAINING : The selected regression model is trained on a subset of the database and hyperparameters are adjusted to optimize model performance.

6. EVALUATION CRITERIA: We have chosen the appropriate evaluation criteria to evaluate the performance of the model. These metrics include Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R ^ 2) to measure the accuracy and goodness of fit of the model.

7. VERIFICATION AND TESTING: Model performance is validated using a separate validation database and tested on unseen data to assess generalizability.

8. DEPLOYMENT MODEL: Once the model has shown satisfactory performance, it is deployed for actual prediction.

DATABASE AND PROCESSING:

The database used for this project contains information about movies, including genre, budget, director, actors, and year of release, along with their respective IMDb scores. Data processing involves handling missing values, coding categorical variables using techniques such as one-hot coding, and scaling numerical features for generalization.

MODEL SELECTION AND EVALUATION CRITERIA:

For this IMDb score prediction problem, we chose to use the *Linear Regression* algorithm. Linear regression is a simple but effective algorithm for regression problems, making it a suitable choice for predicting IMDb scores. It assumes a linear relationship between the characteristic and the target variable.

EVALUATION CRITERIA:

1. MEAN ABSOLUTE ERROR (MAE) measures the average absolute difference between the IMDb score prediction and the actual score. A lower MAE indicates a more accurate model.

2. MEAN SQUARED ERROR (MSE) measures the average squared difference between the predicted and actual scores. It penalizes larger errors more severely and is useful for identifying outliers.

3. R-SQUARED (R ^ 2) determines the proportion of variance in the IMDb score that the model can explain. An R^2 score close to 1 indicates that the model fits the data better.

IN SUMMARY, THE IMDB SCORE PREDICTION PROJECT AIMS TO DEVELOP A REGRESSION MODEL USING LINEAR REGRESSION TO PREDICT THE IMDB SCORE FOR MOVIES. CHOOSING EVALUATION METRICS SUCH AS MAE, MSE, AND R^2 WILL HELP YOU EVALUATE THE ACCURACY AND PERFORMANCE OF THE MODEL IN PREDICTING IMDB SCORES.

CODING :

```
# Import necessary libraries

import pandas as pd

import numpy as np

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LinearRegression

from sklearn.metrics import mean_absolute_error, mean_squared_error,
r2_score

from sklearn.preprocessing import LabelEncoder
```

```
import matplotlib.pyplot as plt

# Load the dataset (Assuming the dataset is stored in a CSV file)
data = pd.read_csv("/content/NetflixOriginals.csv", encoding="latin1")

# Data Preprocessing
# Handling missing values
data.fillna(0, inplace=True)

# Encoding categorical variables
label_encoders = {}
categorical_cols = ["Genre", "Language"]
for col in categorical_cols:
    le = LabelEncoder()
    data[col] = le.fit_transform(data[col])
    label_encoders[col] = le

# Feature Engineering
data['Premiere'] = pd.to_datetime(data['Premiere']).dt.year

# Convert 'Runtime' to string and then extract digits
data['Runtime'] = data['Runtime'].astype(str).str.extract('(\d+)').astype(float)

# Data Visualization
plt.scatter(data['Premiere'], data['IMDB Score'])
plt.xlabel('Year of Premiere')
```

```
plt.ylabel('IMDB Score')
plt.title('IMDB Score vs. Year of Premiere')

# Split the data into features and target
X = data[['Genre', 'Premiere', 'Runtime', 'Language']]

# Exclude 'Title' from the features
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)

# Model Selection
model = LinearRegression()

# Model Training
model.fit(X_train, y_train)

# Model Evaluation
y_pred = model.predict(X_test)
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Absolute Error: {mae}")
```

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

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


# Predict IMDb Scores for a specific movie

movie_to_predict = pd.DataFrame({'Genre': [1],  # Use label encoder values for Genre and Language

                                 'Premiere': [2020],

                                 'Runtime': [120],

                                 'Language': [2]})

predicted_score = model.predict(movie_to_predict)

print(f"Predicted IMDb Score: {predicted_score[0]}")
```

OUTPUT :