Applied DataScience :

Project Name: Predicting IMDb Scores

Project Description: 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 4: 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.

Working Procedure:

Program :

In [1]:

import pandas as pd

import numpy as no

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

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.

In [6]:

# 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 [6]:

# Shape of the data after feature engineering

(584, 6)

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.

In [7] :

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)

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.

In [8] :

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 [8] :

MSE: 0.5682

R-squared: 0.7234

Conclusion:

We have by performing feature engineering, model training and evaluation. the Netflix Originals IMDb Scores dataset for analysis.