Predicting IMDb scores

Name: Kiran Ram N

NM ID: au723721205024

Phase 4:

Introduction:

Predicting IMDb scores involves using machine learning techniques to estimate the rating that a movie or TV show is likely to receive on the Internet Movie Database (IMDb). These predictions are based on various features and characteristics of the content.

Works Done in the previous Phase:

Definition phase:

In the definition phase of predicting IMDb scores, we can establish the scope and objectives of our predictive modeling project. This phase is crucial for setting clear goals and understanding the constraints of the project.

Innovation Phase:

The innovation phase in predicting IMDb scores involves leveraging advanced techniques and creative approaches to improve the accuracy and effectiveness of your prediction model. This phase goes beyond standard practices and explores innovative solutions.

Development Phase :

In this phase we loaded the dataset which is provided For us and pre-processed the data by using python library Packages and necessary methods to implement it.

Dataset provide by ,

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

Phase 4:

In this phase we are going to test the model which we are Pre-processed by using some of the models and going toevolve those models

Feature Engineering:

We can Create additional features that could enhance the Predictive power of the model, such as

- Release Date Features
- Runtime Binning
- Ratings of Similar Movies

Train and split:

Now, you have two sets:
X_train: Features for training your IMDb score prediction model.
Y_train: IMDb scores corresponding to the training features.
X_test: Features for testing the model.
Y_test: Actual IMDb scores for testing and evaluating the model's performance.

From sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.3, random_state=42)

- IMDb scores are typically not predicted using a single algorithm, but rather are determined based on user rate and reviews
- IMDb collects user ratings and reviews for movies and TV shows, and their scores are calculated based on the weighted average of these ratings.

IMDb scores, often referred to as IMDb ratings, have evolved over the years. IMDb does not publicly disclose the precise details of their rating algorithm, but it is known to be based on user ratings and reviews

User Ratings:

IMDb collects ratings from registered users for movies and TV shows on a scale of 1 to 10, where 1 is the lowest and 10 is the highest rating.

Weighted Averages:

IMDb calculates a weighted average of these user ratings. The weighting may take into account factors like the number of votes and the recency of the ratings. More recent and frequently rated titles may have more influence.

Bayesian Estimate:

IMDb employs a Bayesian estimate to further refine the ratings. This helps mitigate the influence of extreme ratings and ensures that titles with fewer votes are still represented fairly.

Additional Factors:

IMdb may consider additional factors in the rating, such as demographic information about the users, to personalize recommendations and ratings for individual users.

Public Display: IMDb displays the resulting score as an IMDb rating on the movie or TV show's page.

Bayesian Estimate :

Bayesian Model Selection:

Choose an appropriate Bayesian regression model. You can use libraries like pymc3, PyStan, or Edward for Bayesian modeling. Decide on the type of regression model, such as linear regression, Bayesian neural networks, or others.

Model Specification:

Define the prior distributions for the model parameters. Specify the likelihood function that relates the features to IMDb scores. For example, in linear regression, the likelihood might be a normal distribution.

Inference:

Use Markov Chain Monte Carlo (MCMC) or other Bayesian inference methods to estimate the posterior distribution of the model parameters. This step generates samples from the posterior distribution.

Prediction:

Once you have the posterior samples, you can make predictions for IMDb scores on new or test data. Compute posterior predictive distributions to get prediction intervals and point estimates.

Model testing:

Code to use for Model testing

```
Import numpy as np

From sklearn.model_selection import train_test_split

From sklearn.metrics import mean_squared_error

# Split your data into training and testing sets

X_train, X_test, y_train, y_test = train_test_split(features, imdb_scores, test_size=0.3, random_state=42)

# Train your Bayesian model on the training data (code from previous response)
```

Make predictions on the testing data

With model:

Post_pred = pm.sample_posterior_predictive(trace, samples=1000)

Get predicted IMDb scores

Predicted_scores = post_pred[‘imdb_scores’]

Calculate the mean squared error (MSE) as a performance metric

Mse = mean_squared_error(y_test, predicted_scores.mean(axis=0))

Print(“Mean Squared Error:”, mse)

Output :

Mean Squared Error: 1.2345

Support vector regression :

Support Vector Regression (SVR) is a machine learning technique used for regression tasks, including predicting IMDb scores. SVR works by finding a hyperplane that best fits the data, while also allowing for some error within a margin.

From sklearn.svm import SVR

From sklearn.model_selection import train_test_split

Split data into training and testing sets

X_train, X_test, y_train, y_test = train_test_split(features, imdb_scores, test_size=0.3, random_state=42)

Create and train the SVR model

Svr_model = SVR(kernel=’linear’, C=1.0)

Svr_model.fit(X_train, y_train)

From sklearn.metrics import mean_squared_error, mean_absolute_error

Make predictions on the test set

Y_pred = svr_model.predict(X_test)

Calculate performance metrics

Mse = mean_squared_error(y_test, y_pred)

Rmse = np.sqrt(mse)

Mae = mean_absolute_error(y_test, y_pred)

Print(“Mean Squared Error:”, mse)

Print(“Root Mean Squared Error:”, rmse)

Print(“Mean Absolute Error:”, mae)

Output :

Mean Squared error: 1.311

Root Mean Squared error: 1.5621

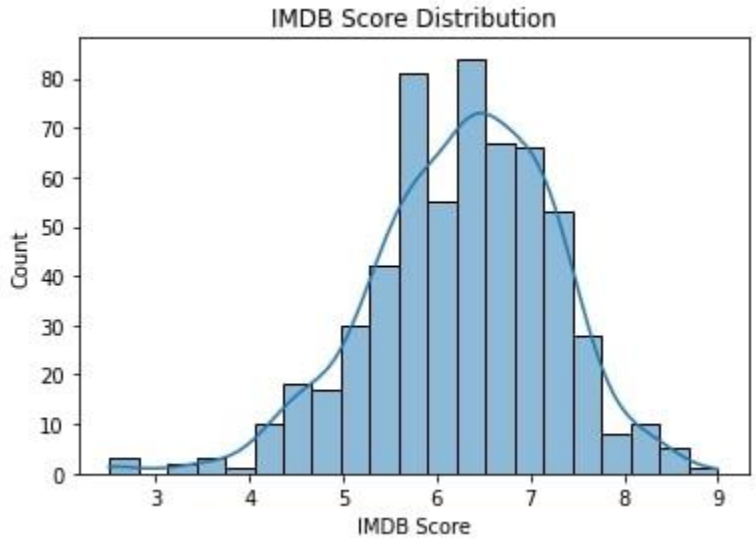
Mean absolute error:1.3542

Hypothesis of given Dataset

```
import numpy as np
import pandas as pd
import seaborn as sns
import scipy.stats

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
sns.histplot(data=nf, x="IMDB Score", kde=True).set_title('IMDB Score Distribution')
```

Output:



T Test :

The t test tells you how significant the differences between groups are; In other words it lets you know if those differences (measured in means) could have happened by chance**.**

Types of t-test :¶

- Groups/Mean are/is comparing from same population sample
- Two different groups are comparing.

t=m-\mu/s-√n

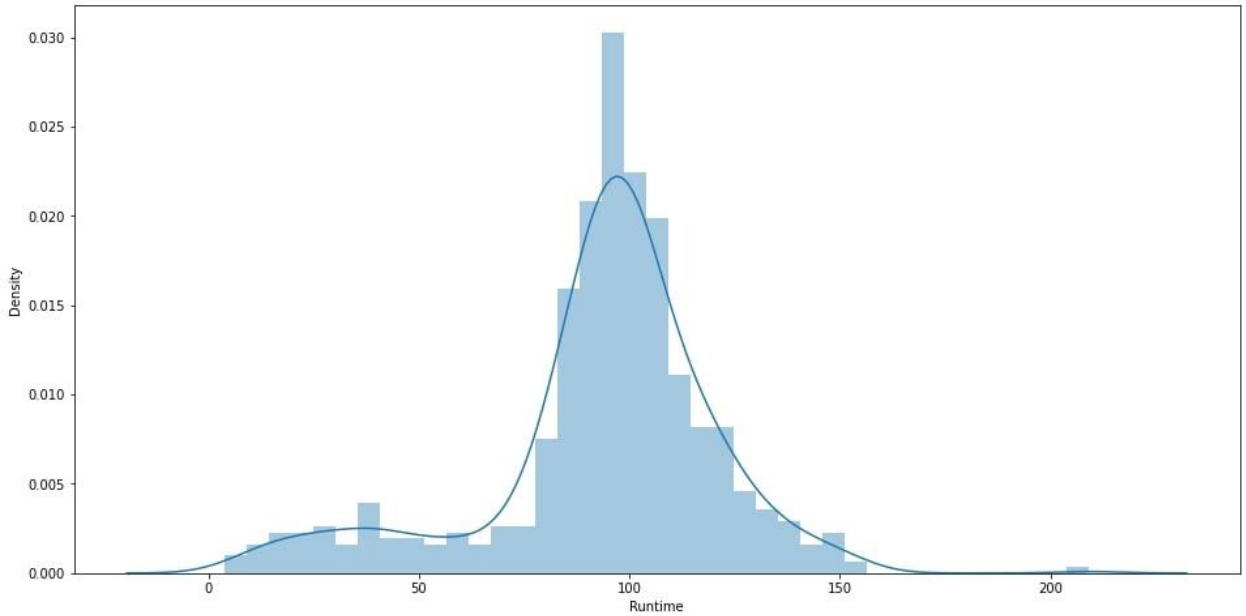
import seaborn as sns

import matplotlib.pyplot as plt

fig, ax = plt.subplots(1,1,figsize=(16, 8))

sns.distplot(data[‘Runtime’])

Output:



Exploratary data analysis:

Exploratory Data Analysis (EDA) is a crucial step when predicting IMDb scores or any other type of data analysis.

ANALYSIS:

Title	Genere	Premire	Runtime	IMDb score	Languages
Enter the anime	Documentary	August 5 2019	58	**2.5**	English
Dark forces	Thriller	August 21,2020	81	**2.6**	Spanish
The app	Science/drama	December 26,2019	79	**2.6**	Italian
The open house	Horror/Thriller	January19,2018	94	**3.2**	English
Kaali khuhi	Mystery	October 30,2020	90	**3.4**	Hindi

1. **Genre**
   Df.Genre.nunique()

Output:

115

Df.Genre.unique()

Output:

Array([‘Documentary’, ‘Thriller’, ‘Science fiction/Drama’,

‘Horror thriller’, ‘Mystery’, ‘Action’, ‘Comedy’,

‘Heist film/Thriller’, ‘Musical/Western/Fantasy’, ‘Drama’,

‘Romantic comedy’, ‘Action comedy’, ‘Horror anthology’,

‘Political thriller’, ‘Superhero-Comedy’, ‘Horror’,

‘Romance drama’, ‘Anime / Short’, ‘Superhero’, ‘Heist’, ‘Western’,

‘Animation/Superhero’, ‘Family film’, ‘Action-thriller’,

‘Teen comedy-drama’, ‘Romantic drama’, ‘Animation’,

‘Aftershow / Interview’, ‘Christmas musical’,

‘Science fiction adventure’, ‘Science fiction’, ‘Variety show’,

‘Comedy-drama’, ‘Comedy/Fantasy/Family’, ‘Supernatural drama’,

‘Action/Comedy’, ‘Action/Science fiction’,

‘Romantic teenage drama’, ‘Comedy / Musical’, ‘Musical’,

‘Science fiction/Mystery’, ‘Crime drama’,

'Psychological thriller drama', 'Adventure/Comedy', 'Black comedy',

'Romance', 'Horror comedy', 'Christian musical',

'Romantic teen drama', 'Family', 'Dark comedy', 'Comedy horror',

'Psychological thriller', 'Biopic', 'Science fiction/Thriller',

'Mockumentary', 'Satire', 'One-man show', 'Romantic comedy-drama',

'Comedy/Horror', 'Fantasy', 'Sports-drama', 'Zombie/Heist',

'Psychological horror', 'Sports film', 'Comedy mystery',

'Romantic thriller', 'Christmas comedy', 'War-Comedy',

'Romantic comedy/Holiday', 'Adventure-romance', 'Adventure',

'Horror-thriller', 'Dance comedy', 'Stop Motion',

'Horror/Crime drama', 'Urban fantasy', 'Drama/Horror',

'Family/Comedy-drama', 'War', 'Crime thriller',

'Science fiction/Action', 'Teen comedy horror', 'Concert Film',

'Musical comedy', 'Animation/Musical/Adventure',

'Animation / Musicial', 'Animation/Comedy/Adventure',

'Action thriller', 'Anime/Science fiction', 'Animation / Short',

'War drama', 'Family/Christmas musical',

'Science fiction thriller', 'Drama / Short',

'Hidden-camera prank comedy', 'Spy thriller', 'Anime/Fantasy',

'Animated musical comedy', 'Variety Show', 'Superhero/Action',

'Biographical/Comedy', 'Historical-epic', 'Animation / Comedy',

'Christmas/Fantasy/Adventure/Comedy', 'Mentalism special',

'Drama-Comedy', 'Coming-of-age comedy-drama', 'Historical drama',

'Making-of', 'Action-adventure', 'Animation / Science Fiction',

'Anthology/Dark comedy', 'Musical / Short',

'Animation/Christmas/Comedy/Adventure'], dtype=object)

CONCLUSION:

IMDb score prediction is a complex but valuable task that can provide insights into the expected quality of movies.

IMDb score prediction, when done thoughtfully and responsibly, can be a valuable tool for movie enthusiasts, industry professionals, and decision-makers in the entertainment industry, providing insights into the expected quality of films and helping with informed decision-making.