Predicting IMDb scores

Name : Kiran Ram N

NM Id : au723721205024

Phase 5:

Works Done in previous phase:

Phase 1:

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.

Problem Definition:

IMDb scores are determined By user ratings and can change over time as more Users rate the movie or show.

Phase 2:

Data source:

The data for analysis is taken from,

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

Design Thinking:

Design Thinking is a user-centered, iterative approach to problem-solving and innovation.

INNOVATION:

- Deep Learning and Sentiment Analysis
- Collaborative Filtering
- Contextual Recommendations
- Interactive Visualization
- Crowdsourced Predictions
- Incorporate Social Media Data

Phase 3:

Data=pd.read_csv("/kaggle/input/netflix-original-films-imdb-Scores/NetflixOriginals.csv",encoding = "ISO-8859-1")

dataDate= data.copy()

data.head()

Data Import :

Title	Genere	Premire	Runtime	IMDb score	Language
Enter the animie	Documentary	August 5,2019	58	**2.5**	English/japanese
Dark Forces	Thriller	August 21,2020	81	**2.6**	Spanish
The app	Science/Dramatic	December 26,2019	79	**2.6**	Italian
The open house	Horror/Thriller	January19,2018	94	**3.8**	English
Kaali khuuhi	Mystery	October 30, 2020	90	**3.4**	Hindi

Data Preprocessing:

- Data cleaning
- Handling Missing Data

- Train and split data
- Normalization and scaling
- Model specific preprocessoing

Different analysis:

Regression Analysis:

Datas can treat the IMDB scores as continuous values and perform Regression analysis to predict scores. Linear regression, decision trees, Random forests, or gradient boosting algorithms are commonly used.

Classification Analysis:

Convert IMDB scores into categories (e.g., low, medium, high) and useClassification algorithms like logistic regression, SVM, or deep learning Models to predict the class.

Deep Learning Models:

Utilize deep neural networks, such as recurrent neural networks (RNNs) For text data or convolutional neural networks (CNNs) for image data, To predict IMDB scores.

Data Modeling:

Create a data model if necessary, which may involve defining Relationships between different data tables.

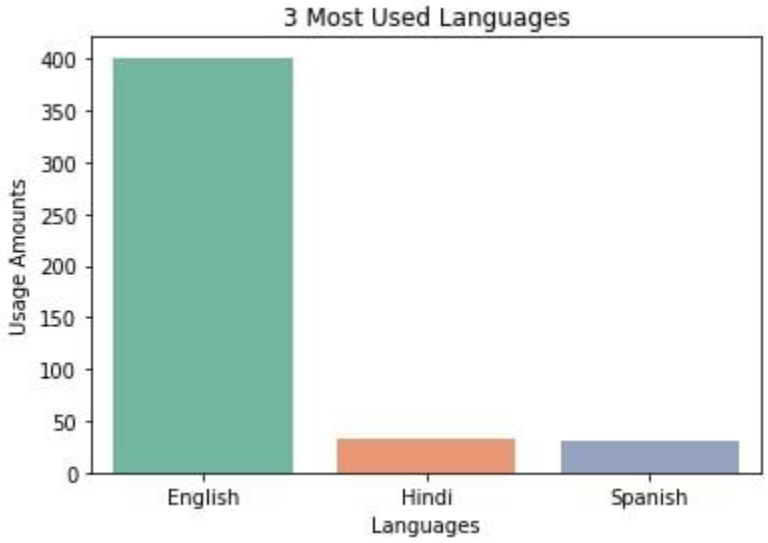
Visualization Creation:

In Cognas we can create various types of visualizations, like bar charts,Line charts, or scatter plots. Choose the type of visualization that best Represents IMDb score prediction.

Find the 3 most used languages in the movies in the data set.

```
Df_lang = df[‘language’].value_counts()
Df_lang.head(3).plot(kind=’bar’)
Plt.show(block=True)
```

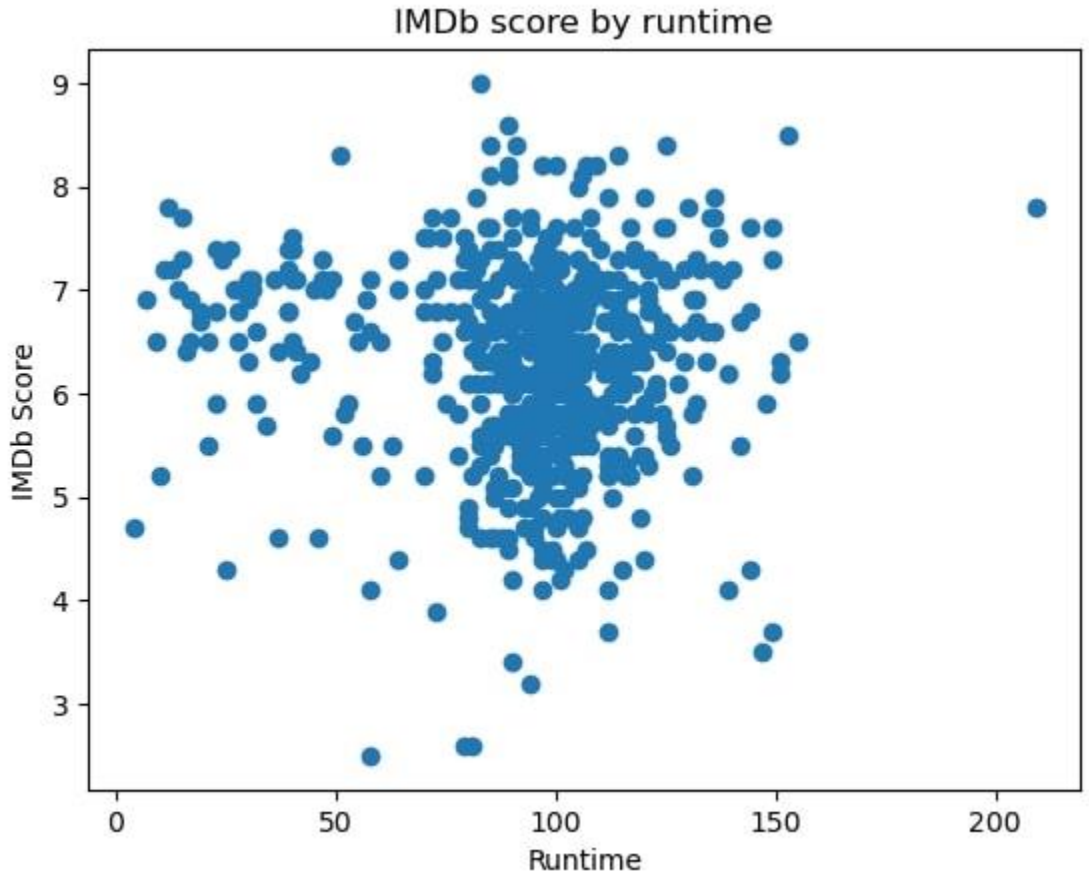
Output:



Scattered plot

```
Df[[‘title’, ‘runtime’]].sort_values(‘runtime’, ascending=False).head(10).
Plot(x=’title’, y=’runtime’, kind=’bar’)
Plt.xlabel(‘Movie Title’)
Plt.ylabel(‘Runtime’)
Plt.show(block=True) df[[‘title’, ‘runtime’]].sort_values(‘runtime’, ascen
Ding=False).head(10).plot(x=’title’, y=’runtime’, kind=’bar’)
Plt.xlabel(‘Movie Title’)
Plt.ylabel(‘Runtime’)
Plt.show(block=True)
```

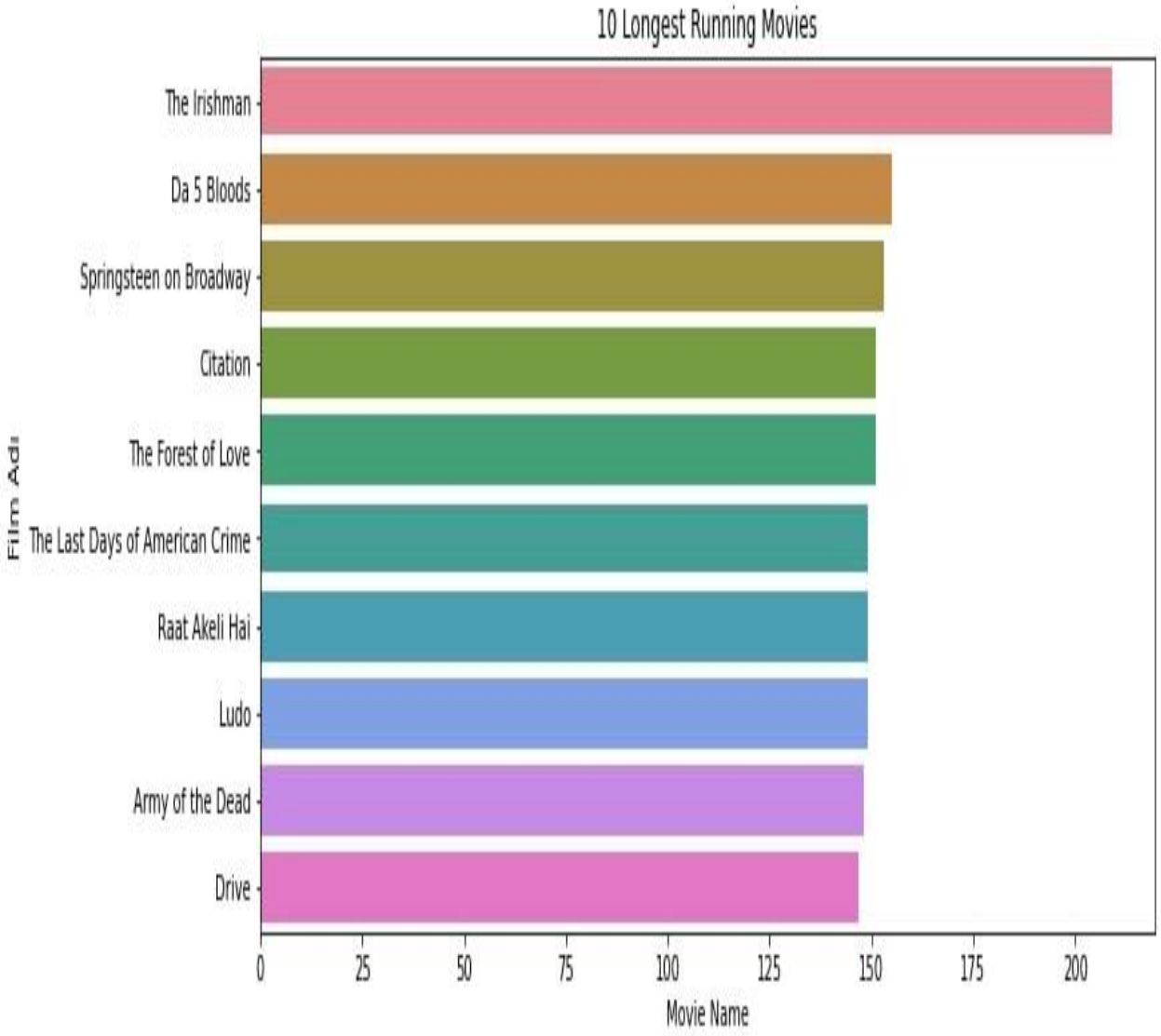
Output:



What are the top 10 movies with the highest ‘runtime’? Visualize it.

```
Df[[‘title’, ‘runtime’]].sort_values(‘runtime’, ascending=False).head(10).
Plot(x=’title’, y=’runtime’, kind=’bar’)
Plt.xlabel(‘Movie Title’)
Plt.ylabel(‘Runtime’)
Plt.show(block=True:
```

Output:


10 Longest Running Movies
The Irishman
Da 5 Bloods
Springsteen on Broadway
Citation
The Forest of Love
The Last Days of American Crime
Raat Akeli Hai
Ludo
Army of the Dead
Drive
0
25
50
75
100
125
150
175
200
Movie Name

Phase 4:

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

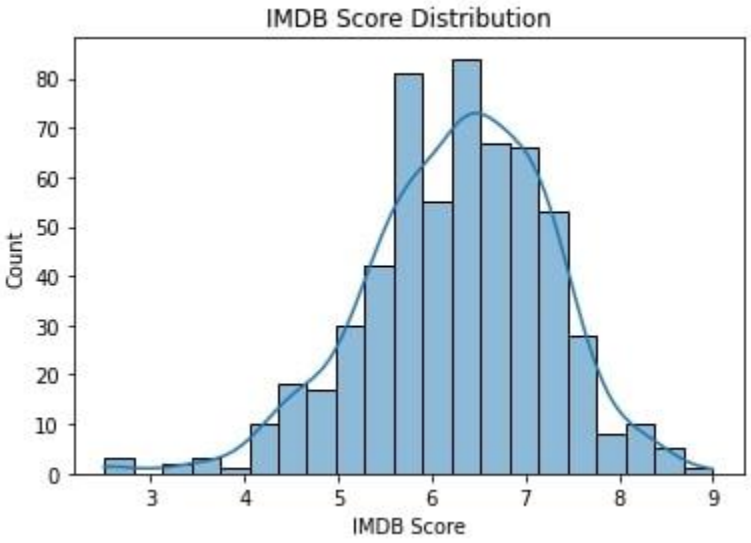
Root Mean Squared error: 1.5621

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:

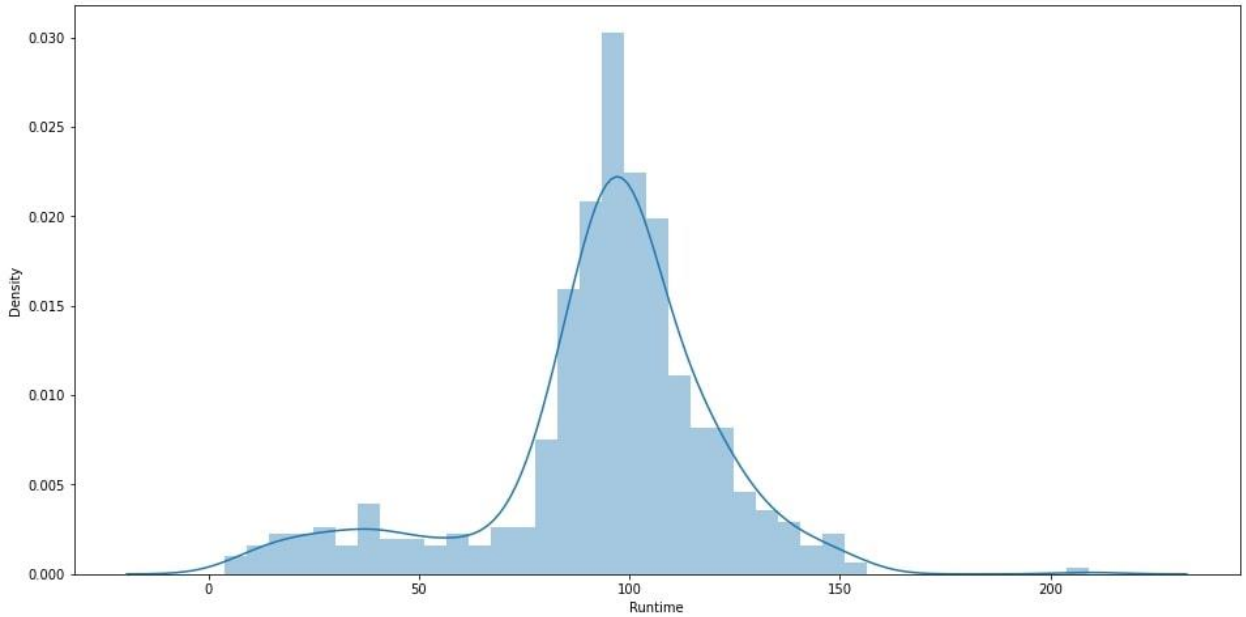


T Test :

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.

```
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