VIT
BANGALORE

Name : Dharshani A

Register No: 23MSP3068

Section : Section -2

Course Code : CS6102

Course Name : PYTHON FOR DATA ANALYTICS

Project Title : Top IMDb Movies Analysis and Prediction

Top IMDb Movies Analysis and Prediction

PYTHON FOR DATA ANALYTICS [CS6102]

By :

- Gurucharan S (23MSP3074)
- Dharshani A (23MSP3068)

AIM: To analyze and visualize the data from the IMDB Top 250 movies, establish database connectivity, and implement linear regression for predicting movie ratings.

DESCRIPTION: The main goal of this project is to gain insights from the IMDb top 250 movies,preprocess the data for analysis, visualize the data patterns, establish a connection with an SQLite database to store and retrieve movie data, and implement a linear regression model to predict movie ratings.

Source of the dataset: IMDB (as inferred from the filename “imdbTop250.csv”).

Dataset description: The dataset contains various columns such as title, rating, score, and runtime of the top 250 movies on IMDB.

Source Code:

[1]:
```
# Import necessary libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import sqlite3
import shutil

from tabulate import tabulate
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics
from google.colab import drive
```

[ ]:
```
# Mount Google Drive
drive.mount('/content/drive')
src = '/content/drive/MyDrive/Colab Notebooks/imdbTop250.csv'
dst = '/content/drive/My Drive/Colab Notebooks/imdbTop250m.csv'

# Copy file from source location to destination location
shutil.copy(src, dst)
```

Building Predictive model and testing: Linear regression is a statistical method that models the relationship between a dependent variable and one or more independent variables. In predictive analysis, it is used to forecast values based on known data, allowing for the prediction of outcomes based on input variables.

In the below code linear regression is being applied to predict movie “Rating” based on its “Score”. By splitting the data into training and testing sets, the model is trained on a subset of the data and then evaluated on unseen data. This approach, commonly used in predictive analysis, helps in understanding how well the model will generalize to new, previously unseen data.

Data Pre-Processing: The data preprocessing involves reading the dataset, slicing it to get the first 50 movies, and handling missing values by dropping rows that have any.

```
[ ]: # Define a class for the use case
     class Usecase:
         @staticmethod
         def dataPreprocessing():
             # Read the CSV file into a DataFrame
             df = pd.read_csv('/content/drive/My Drive/Colab Notebooks/imdbTop250m. ↪csv')
             df = df.iloc[:51] # Filter the first 50 rows
             df = df.dropna() # Drop rows with missing values
             return df

         df1 = dataPreprocessing()
```

Exploratory Data Analysis(Visualization): The exploratory data analysis comprises several visual representations:

- Boxplot to identify outliers in votes.
- Scatter plot for comparing votes against runtime.
- Bar plot to show top 20 movie ratings.
- Histogram for comparing movie rankings with their runtime.

```
# Data Visualization code snippets

    @staticmethod
    def visualization_outliers(df):
        # Visualize outliers using a boxplot
        plt.figure(figsize=(10, 6))
        plt.xlabel("Total votes")
```

```
        plt.title("Identifying Outliers")
        sns.boxplot(y="Votes", data=df)
        plt.show()

    @staticmethod
    def visualization_scatterplot(df):
        # Visualize data using a scatterplot
        plt.figure(figsize=(10, 6))
        plt.xlabel("Total Votes")
        plt.ylabel("Runtime")
        plt.title("Comparison of Total votes and Runtime")
        sns.scatterplot(x="Votes", y="RunTime", data=df)
        plt.show()

    @staticmethod
    def visualization_barplot(df):
        # Visualize top 10 movies and their ratings using a barplot
        df_subset = df.iloc[0:10]
        plt.figure(figsize=(10, 6))
        plt.title("Movies and their Ratings")
        sns.barplot(x="Title", y="Rating", data=df_subset)
        plt.xticks(rotation=90) # Rotate x-axis labels for clarity
        plt.show()

    @staticmethod
    def visualization_histogram(df):
        # Visualize data using a histogram
        plt.figure(figsize=(10, 6))
        plt.xlabel("Ranking")
        plt.ylabel("Runtime")
        plt.title("Comparison of Total votes and Runtime")
        sns.histplot(data=df, x="Ranking", y="RunTime")
        plt.show()
```

Storing Data in Database: The processed data is stored in an SQLite database named “use case.db”. The movie details are saved under the table “TopIMDBMovies”. The database further allows for querying and extraction of movies based on different criteria related to their ratings. An SQLite database named “usecase.db” is created. The data from the dataset (columns 1 to 10) is stored in a table named “TopIMDBMovies” in the SQLite database. Various SQL queries are executed to retrieve and analyze data. For example:

- Movies with a rating of less than 8.
- The count of movies with a rating greater than 8.
- The movie with the highest rating.
- The movie with the lowest rating.

```
# Database Connectivity code snippets

    @staticmethod
    def database_connectivity(df1):
        # Extract columns 1 to 10 from the df1 DataFrame
        df2 = df1.iloc[:, 1:11]

        # Create a connection to the SQLite database
        conn = sqlite3.connect('usecase.db')
        cursor = conn.cursor()

        # Write the data to the database
        df2.to_sql(name="TopIMDBMovies", con=conn, if_exists='replace', index=False)

        # Query to retrieve records where RATING is less than 8
        select_query = "SELECT * FROM TopIMDBMovies WHERE RATING < 8"

        # Query to count the number of movies with a RATING greater than 8
count_query = "SELECT COUNT(TITLE) AS BEST_MOVIES FROM TopIMDBMovies WHERE RATING > 8"

        # Query to display the movie with the highest rating
        highest_query = "SELECT * FROM TopIMDBMovies WHERE Rating IN (SELECT MAX(Rating) FROM TopIMDBMovies )"

        # Query to display the movie with the minimum rating
        lowest_query = "SELECT * FROM TopIMDBMovies WHERE Rating IN (SELECT MIN(Rating) FROM TopIMDBMovies )"

        # Execute the queries and fetch results
        cursor.execute(select_query)
        results_low_rating = cursor.fetchall()

        cursor.execute(count_query)
        result_best_movies = cursor.fetchall()

        cursor.execute(highest_query)
        result_highest_rating = cursor.fetchall()

        cursor.execute(lowest_query)
        result_lowest_rating = cursor.fetchall()

        # Get column names (headers) for the first query
        headers_low_rating = [description[0] for description in conn.execute(select_query).description]
```

```
        # Get column names (headers) for the second query
        headers_best_movies = [description[0] for description in conn.
↪execute(count_query).description]

        # Get column names (headers) for the third query
        headers_highest_rating = [description[0] for description in conn.
↪execute(highest_query).description]

        # Get column names (headers) for the fourth query
        headers_lowest_rating = [description[0] for description in conn.
↪execute(lowest_query).description]

        # Print results in tabular form using tabulate
        print("Movies with Rating < 8:")
        print(tabulate(results_low_rating, headers_low_rating, tablefmt="grid")) print("\n")

        print("Number of Best Movies (Rating > 8):")
        print(tabulate(result_best_movies, headers_best_movies,␣
↪tablefmt="grid"))
        print("\n")

        print("Movie with the Highest Rating:")
        print(tabulate(result_highest_rating, headers_highest_rating,␣
↪tablefmt="grid"))
        print("\n")

        print("Movie with the Lowest Rating:")
        print(tabulate(result_lowest_rating, headers_lowest_rating,␣
↪tablefmt="grid"))

        # Commit and close the connection
        conn.commit()
        cursor.close()
        conn.close()
```

Building Predictive model and testing: Linear regression is a statistical method that models the relationship between a dependent variable and one or more independent variables. In predictive analysis, it is used to forecast values based on known data, allowing for the prediction of outcomes based on input variables.

In the below code linear regression is being applied to predict movie “Rating” based on its “Score”. By splitting the data into training and testing sets, the model is trained on a subset of the data and then evaluated on unseen data. This approach, commonly used in predictive analysis, helps in understanding how well the model will generalize to new, previously unseen data.

```
# Data preparation for linear regression and rating prediction code snippet

    @staticmethod
    def dataset_linearregression(df):
        # Prepare data for linear regression
            y = df["Rating"]
            x = df[["Score"]]
            x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3,
random_state=10)

# Create an instance of the Usecase class
usecase_instance = Usecase()

# Perform data preprocessing
df1 = usecase_instance.dataPreprocessing()
```

OUTPUT:

```
[ ]: from google.colab import drive
     drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

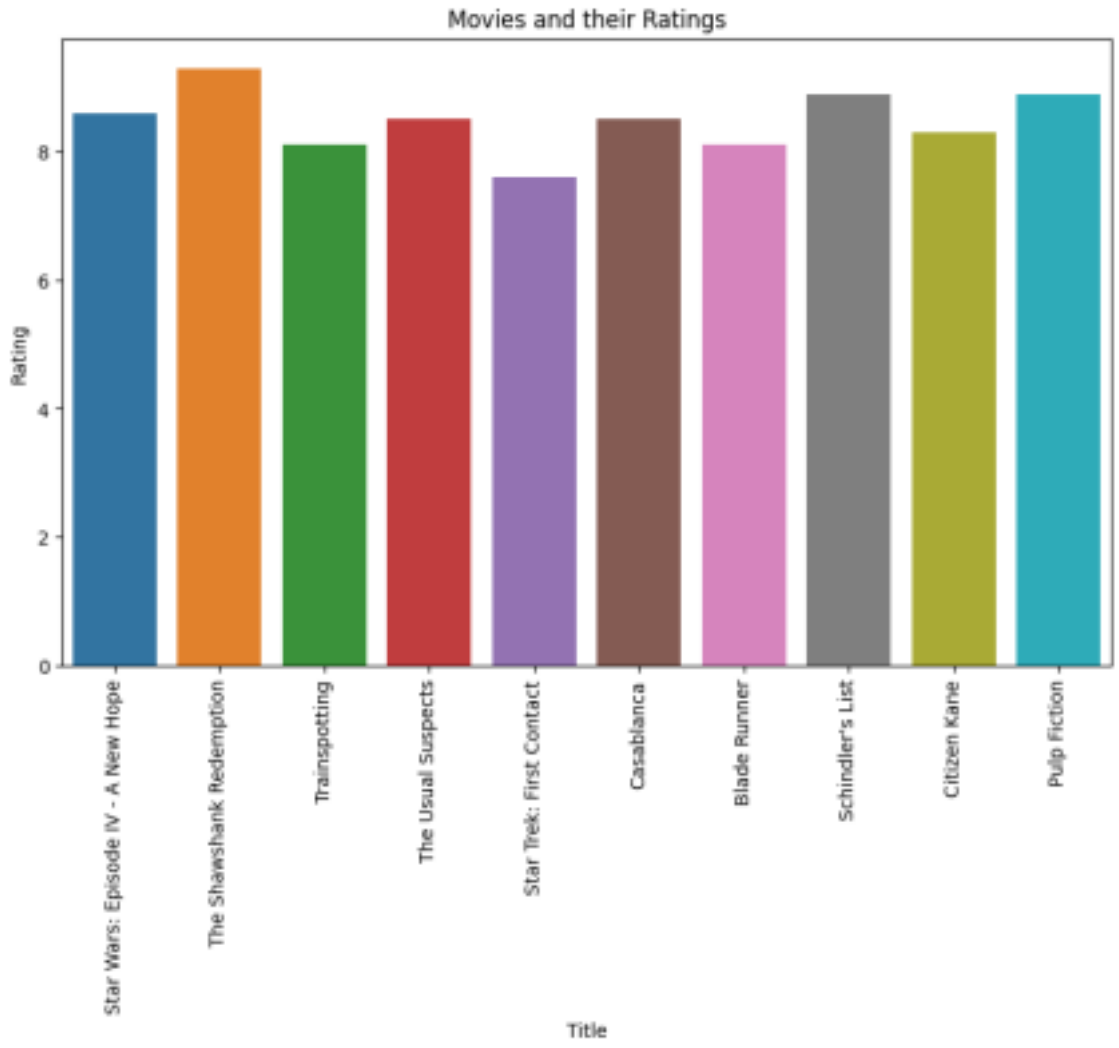
```
[ ]: usecase_instance.df1
```

```
[ ]:
[74] usecase_instance.df1
```

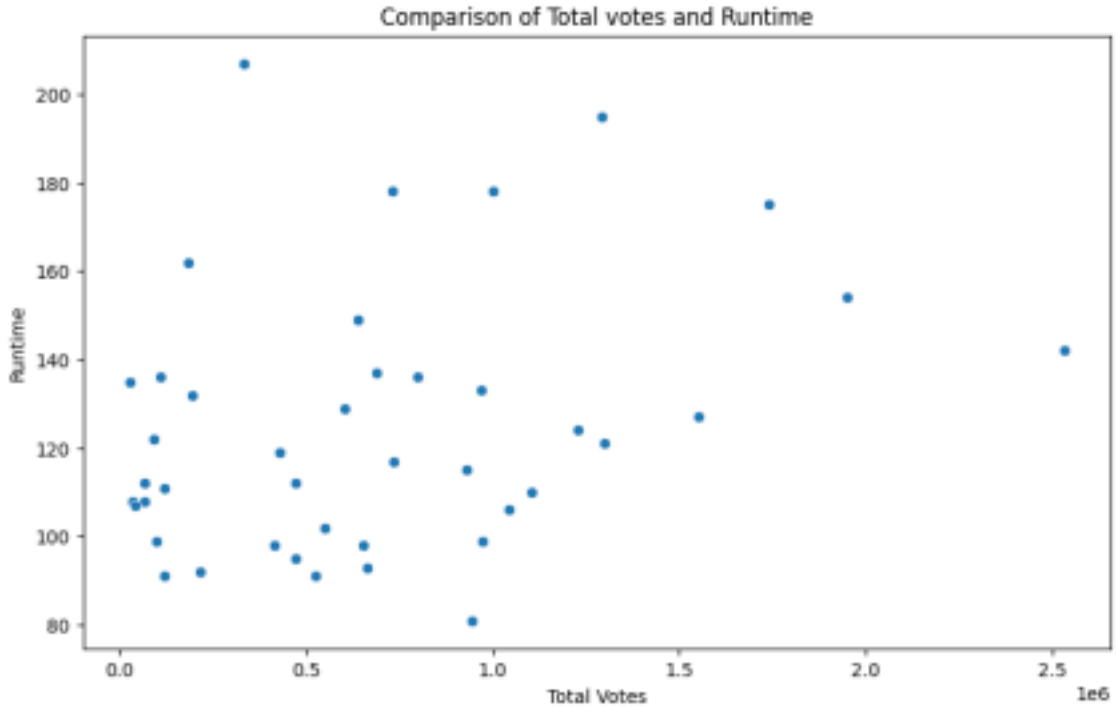
	Ranking	IMDByear	IMDBlink	Title	Date	RunTime	Genre	Rating	Score	Votes	Gross	Director	Cast1	Cast2	Cast3	Cast4
0	1	1996	/title/tt0076759/	Star Wars: Episode IV - A New Hope	1977	121	Action, Adventure, Fantasy	8.6	90.0	1299781	322.74	George Lucas	Mark Hamill	Harrison Ford	Carrie Fisher	Alec Guinness
1	2	1996	/title/tt0111161/	The Shawshank Redemption	1994	142	Drama	9.3	80.0	2529673	28.34	Frank Darabont	Tim Robbins	Morgan Freeman	Bob Gunton	William Sadler
2	3	1996	/title/tt0117951/	Trainspotting	1996	93	Drama	8.1	83.0	665213	16.50	Danny Boyle	Ewan McGregor	Ewen Bremner	Jonny Lee Miller	Kevin McKidd
3	4	1996	/title/tt0114814/	The Usual Suspects	1995	106	Crime, Drama, Mystery	8.5	77.0	1045626	23.34	Bryan Singer	Kevin Spacey	Gabriel Byrne	Chazz Palminteri	Stephen Baldwin
4	7	1996	/title/tt0117731/	Star Trek: First Contact	1996	111	Action, Adventure, Drama	7.6	71.0	122819	92.00	Jonathan Frakes	Patrick Stewart	Jonathan Frakes	Brent Spiner	LeVar Burton
5	8	1996	/title/tt0034583/	Casablanca	1942	102	Drama, Romance, War	8.5	100.0	551575	1.02	Michael Curtiz	Humphrey Bogart	Ingrid Bergman	Paul Henreid	Claude Rains
6	9	1996	/title/tt0083658/	Blade Runner	1982	117	Action, Drama, Sci-Fi	8.1	84.0	736925	32.87	Ridley Scott	Harrison Ford	Rutger Hauer	Sean Young	Edward James Olmos
7	10	1996	/title/tt0108052/	Schindler's List	1993	195	Biography, Drama, History	8.9	94.0	1292510	96.90	Steven Spielberg	Liam Neeson	Ralph Fiennes	Ben Kingsley	Caroline Goodall
8	11	1996	/title/tt0033467/	Citizen Kane	1941	119	Drama, Mystery	8.3	100.0	428750	1.59	Orson Welles	Orson Welles	Joseph Cotten	Dorothy Comingore	Agnes Moorehead
9	12	1996	/title/tt0110912/	Pulp Fiction	1994	154	Crime, Drama	8.9	94.0	1948662	107.93	Quentin Tarantino	John Travolta	Uma Thurman	Samuel L. Jackson	Bruce Willis
10	13	1996	/title/tt0057012/	Dr. Strangelove or: How I Learned to Stop Worr...	1964	95	Comedy, War	8.4	97.0	474011	0.28	Stanley Kubrick	Peter Sellers	George C. Scott	Sterling Hayden	Keenan Wynn
11	14	1996	/title/tt0068646/	The Godfather	1972	175	Crime, Drama	9.2	100.0	1741574	134.97	Francis Ford Coppola	Marlon Brando	Al Pacino	James Caan	Diane Keaton
12	15	1996	/title/tt0116209/	The English Patient	1996	162	Drama, Romance, War	7.4	87.0	186242	78.65	Anthony Minghella	Ralph Fiennes	Juliette Binoche	Willem Dafoe	Kristin Scott Thomas
13	16	1996	/title/tt0112573/	Braveheart	1995	178	Biography, Drama, History	8.3	68.0	1003006	75.60	Mel Gibson	Mel Gibson	Sophie Marceau	Patrick McGoohan	Angus Macfadyen
14	18	1996	/title/tt0116905/	Lone Star	1996	135	Drama, Mystery, Western	7.4	78.0	29329	13.27	John Sayles	Chris Cooper	Elizabeth Peña	Stephen Mendillo	Stephen J. Lang
15	19	1996	/title/tt0114709/	Toy Story	1995	81	Animation, Adventure, Comedy	8.3	95.0	945624	191.80	John Lasseter	Tom Hanks	Tim Allen	Don Rickles	Jim Varney
16	20	1996	/title/tt0073486/	One Flew Over the Cuckoo's Nest	1975	133	Drama	8.7	84.0	969223	112.00	Milos Forman	Jack Nicholson	Louise Fletcher	Michael Berryman	Peter Brocco
17	22	1996	/title/tt0116282/	Fargo	1996	98	Crime, Thriller	8.1	85.0	654107	24.61	Joel Coen, Ethan Coen	William H. Macy	Frances McDormand	Steve Buscemi	Peter Stormare
18	23	1996	/title/tt0110877/	The Postman	1994	108	Biography, Comedy, Drama	7.7	81.0	35664	21.85	Michael Radford, Massimo Troisi	Massimo Troisi	Philippe Noiret	Maria Grazia Cucinotta	Renato Scarpa
19	24	1996	/title/tt0111495/	Three Colors: Red	1994	99	Drama, Mystery, Romance	8.1	100.0	100082	4.04	Krzysztof Kieslowski	Irène Jacob	Jean-Louis Trintignant	Frédérique Feder	Jean-Pierre Lorit
20	25	1996	/title/tt0047478/	Seven Samurai	1954	207	Action, Drama	8.6	98.0	334350	0.27	Akira Kurosawa	Toshirô Mifune	Takashi Shimura	Keiko Tsushima	Yukiko Shimazaki
21	26	1996	/title/tt0080684/	Star Wars: Episode V - The Empire Strikes Back	1980	124	Action, Adventure, Fantasy	8.7	82.0	1228288	290.48	Irvin Kershner	Mark Hamill	Harrison Ford	Carrie Fisher	Billy Dee Williams
22	28	1996	/title/tt0110413/	Léon: The Professional	1994	110	Action, Crime, Drama	8.5	64.0	1105424	19.50	Luc Besson	Jean Reno	Gary Oldman	Natalie Portman	Danny Aiello
23	29	1996	/title/tt0093779/	The Princess Bride	1987	98	Adventure, Family, Fantasy	8.1	77.0	416207	30.86	Rob Reiner	Cary Elwes	Mandy Patinkin	Robin Wright	Chris Sarandon
24	30	1996	/title/tt0114388/	Sense and Sensibility	1995	136	Drama, Romance	7.7	84.0	111580	43.18	Ang Lee	Emma Thompson	Kate Winslet	James Fleet	Tom Wilkinson
25	31	1996	/title/tt0062622/	2001: A Space Odyssey	1968	149	Adventure, Sci-Fi	8.3	84.0	641401	56.95	Stanley Kubrick	Keir Dullea	Gary Lockwood	William Sylvester	Daniel Richter
26	32	1996	/title/tt0105236/	Reservoir Dogs	1992	99	Crime, Drama, Thriller	8.3	79.0	974876	2.83	Quentin Tarantino	Harvey Keitel	Tim Roth	Michael Madsen	Chris Penn
27	33	1996	/title/tt0112682/	The City of Lost Children	1995	112	Drama, Fantasy, Sci-Fi	7.5	73.0	67358	1.51	Marc Caro, Jean-Pierre Jeunet	Ron Perlman	Daniel Emilfork	Judith Vittet	Dominique Pinon
28	34	1996	/title/tt0082971/	Indiana Jones and the Raiders of the Lost Ark	1981	115	Action, Adventure	8.4	85.0	931142	248.16	Steven Spielberg	Harrison Ford	Karen Allen	Paul Freeman	John Rhys-Davies
29	35	1996	/title/tt0117887/	That Thing You Do!	1996	108	Comedy, Drama, Music	6.9	71.0	67061	25.81	Tom Hanks	Tom Hanks	Liv Tyler	Charlize Theron	Tom Everett Scott
30	36	1996	/title/tt0109445/	Clerks	1994	92	Comedy	7.7	70.0	218279	3.15	Kevin Smith	Brian O'Halloran	Jeff Anderson	Marilyn Ghigliotti	Lisa Spoonauer

36	37	1996	/title/tt0088846/	Brazil	1985	132	Drama, Sci-Fi	7.9	84.0	196892	9.93	Terry Gilliam	Jonathan Pryce	Kim Greist	Robert De Niro	Katherine Helmond
37	38	1996	/title/tt0071853/	Monty Python and the Holy Grail	1975	91	Adventure, Comedy, Fantasy	8.2	91.0	525003	1.23	Terry Gilliam, Terry Jones	Graham Chapman	John Cleese	Eric Idle	Terry Gilliam
38	39	1996	/title/tt0047396/	Rear Window	1954	112	Mystery, Thriller	8.5	100.0	473590	36.76	Alfred Hitchcock	James Stewart	Grace Kelly	Wendell Corey	Thelma Ritter
39	40	1996	/title/tt0114369/	Se7en	1995	127	Crime, Drama, Mystery	8.6	65.0	1552226	100.13	David Fincher	Morgan Freeman	Brad Pitt	Kevin Spacey	Andrew Kevin Walker
40	41	1996	/title/tt0090605/	Aliens	1986	137	Action, Adventure, Sci-Fi	8.3	84.0	690005	85.16	James Cameron	Sigourney Weaver	Michael Biehn	Carrie Henn	Paul Reiser
41	42	1996	/title/tt0066921/	A Clockwork Orange	1971	136	Crime, Sci-Fi	8.3	77.0	798504	6.21	Stanley Kubrick	Malcolm McDowell	Patrick Magee	Michael Bates	Warren Clarke
43	44	1996	/title/tt0094336/	Withnail & I	1987	107	Comedy, Drama	7.6	84.0	42901	1.54	Bruce Robinson	Richard E. Grant	Paul McGann	Richard Griffiths	Ralph Brown
44	45	1996	/title/tt0060196/	The Good, the Bad and the Ugly	1966	178	Adventure, Western	8.8	90.0	731123	6.10	Sergio Leone	Clint Eastwood	Eli Wallach	Lee Van Cleef	Aldo Giuffrè
45	46	1996	/title/tt0114746/	12 Monkeys	1995	129	Mystery, Sci-Fi, Thriller	8.0	74.0	602534	57.14	Terry Gilliam	Bruce Willis	Madeleine Stowe	Brad Pitt	Joseph Melito
47	48	1996	/title/tt0112431/	Babe	1995	91	Comedy, Drama, Family	6.8	83.0	122545	66.60	Chris Noonan	James Cromwell	Magda Szubanski	Christine Cavanaugh	Miriam Margolyes
49	50	1996	/title/tt0112818/	Dead Man Walking	1995	122	Crime, Drama	7.5	80.0	92996	39.39	Tim Robbins	Susan Sarandon	Sean Penn	Robert Prosky	Raymond J. Barry

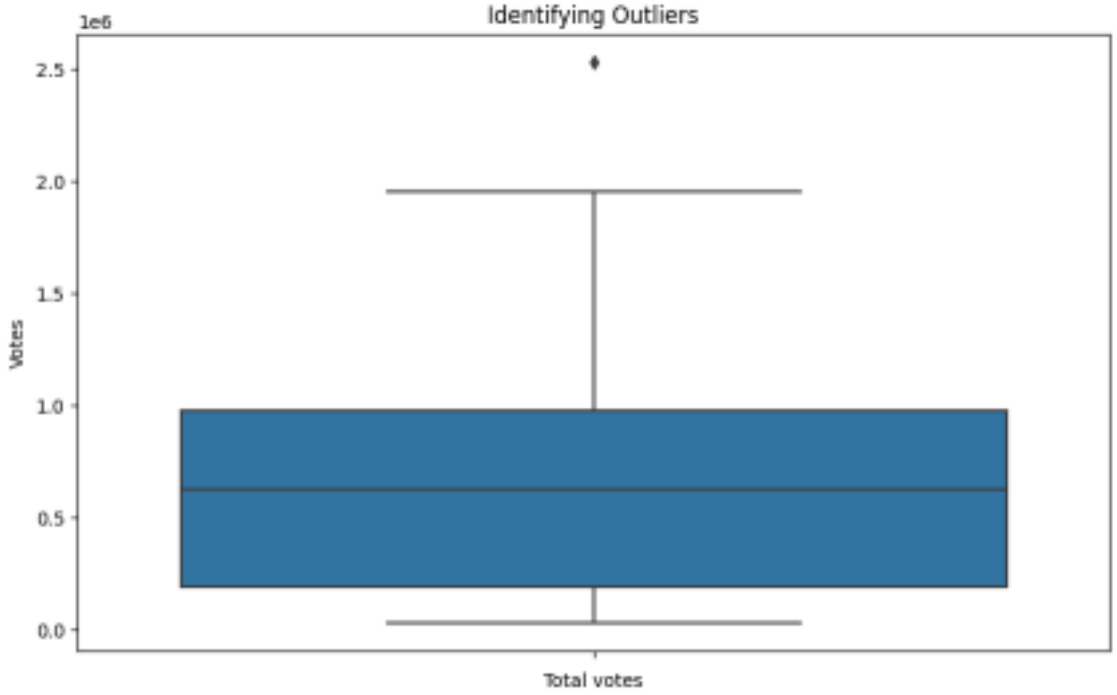
[ ]: usecase_instance.visualization_barplot(df1)



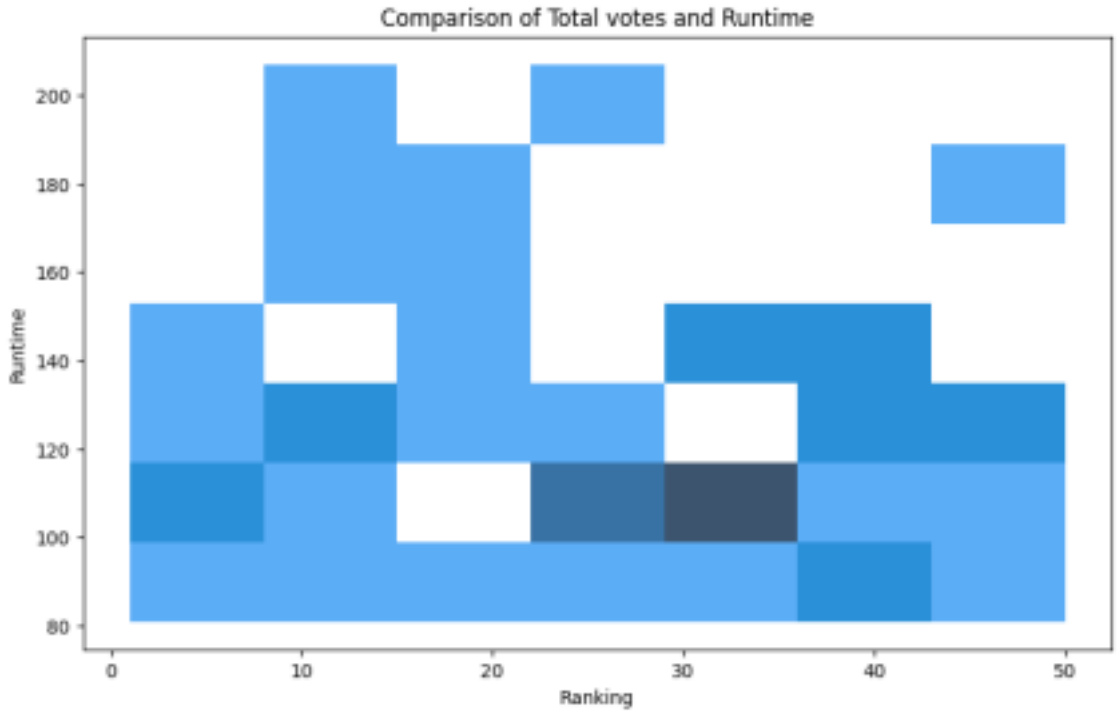
[ ]: usecase_instance.visualization_scatterplot(df1)



[ ]: usecase_instance.visualization_outliers(df1)



[ ]: usecase_instance.visualization_histogram(df1)



[ ]: *# Call database connectivity function*

```
usecase_instance.database_connectivity(df1)
```

```
# Call database connectivity function
usecase_instance.database_connectivity(df1)
```

```
Movies with Rating < 8:
```

IMDByear	IMDBlink	Title	Date	RunTime	Genre	Rating	Score	Votes	Gross
1996	/title/tt0117731/	Star Trek: First Contact	1996	111	Action, Adventure, Drama	7.6	71	122819	92
1996	/title/tt0116209/	The English Patient	1996	162	Drama, Romance, War	7.4	87	186242	78.65
1996	/title/tt0116905/	Lone Star	1996	135	Drama, Mystery, Western	7.4	78	29329	13.27
1996	/title/tt0110877/	The Postman	1994	108	Biography, Comedy, Drama	7.7	81	35664	21.85
1996	/title/tt0114388/	Sense and Sensibility	1995	136	Drama, Romance	7.7	84	111580	43.18
1996	/title/tt0112682/	The City of Lost Children	1995	112	Drama, Fantasy, Sci-Fi	7.5	73	67358	1.51
1996	/title/tt0117887/	That Thing You Do!	1996	108	Comedy, Drama, Music	6.9	71	67061	25.81
1996	/title/tt0109445/	Clerks	1994	92	Comedy	7.7	70	218279	3.15
1996	/title/tt0088846/	Brazil	1985	132	Drama, Sci-Fi	7.9	84	196892	9.93
1996	/title/tt0094336/	Withnail & I	1987	107	Comedy, Drama	7.6	84	42901	1.54
1996	/title/tt0112431/	Babe	1995	91	Comedy, Drama, Family	6.8	83	122545	66.6
1996	/title/tt0112818/	Dead Man Walking	1995	122	Crime, Drama	7.5	80	92996	39.39

```
Number of Best Movies (Rating > 8):
```

BEST_MOVIES
29

```
Movie with the Highest Rating:
```

IMDByear	IMDBlink	Title	Date	RunTime	Genre	Rating	Score	Votes	Gross
1996	/title/tt0111161/	The Shawshank Redemption	1994	142	Drama	9.3	80	2529673	28.34

```
Movie with the Lowest Rating:
```

IMDByear	IMDBlink	Title	Date	RunTime	Genre	Rating	Score	Votes	Gross
1996	/title/tt0112431/	Babe	1995	91	Comedy, Drama, Family	6.8	83	122545	66.6

[ ]:
```
#Setting value for x and y
df2 = usecase_instance.df1
y= df2["Rating"]
x= df2[["Score"]]
x_train,x_test, y_train,y_test = train_test_split(x,y, test_size=0.3, ↪random_state
  = 10)
x_train
```

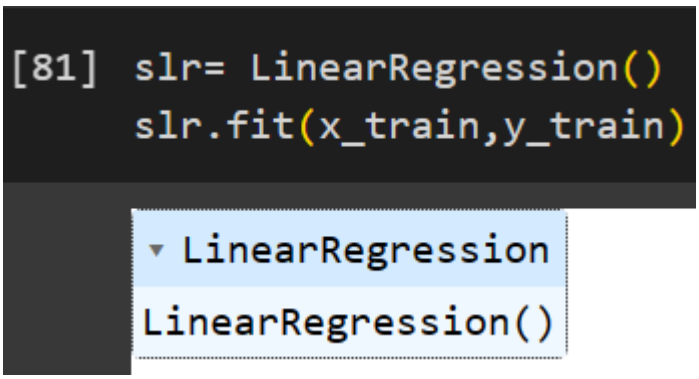
[ ]: Score

```
#Setting value for x and y
df2 = usecase_instance.df1
y= df2["Rating"]
x= df2[["Score"]]
x_train,x_test, y_train,y_test = train_test_split(x,y, test_size=0.3, random_state = 10)
x_train
```

	Score
3	77.0
47	83.0
7	100.0
15	68.0
17	78.0
29	84.0
39	65.0
1	80.0
43	84.0
13	100.0
45	74.0
32	73.0
19	84.0
23	100.0
6	71.0
27	64.0
38	100.0
12	97.0
44	90.0
49	80.0
10	100.0
40	84.0
34	71.0
30	84.0
33	85.0
0	90.0
18	95.0
41	77.0
11	94.0

[ ]:
```
slr= LinearRegression()
slr.fit(x_train,y_train)
```

[ ]: LinearRegression()



[ ]:
```
#Print model coefficients
print('Intercept:',slr.intercept_)
print('Coefficient:', slr.coef_)
```

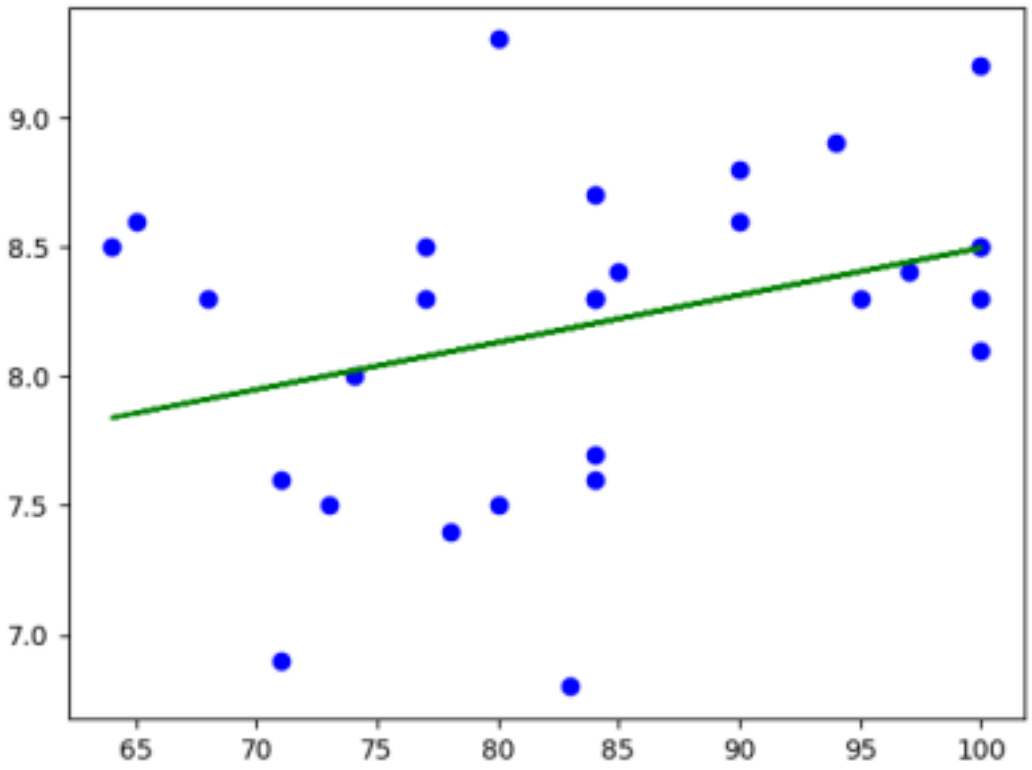
```
Intercept: 6.672150220621699
Coefficient: [0.0182186]
```

[ ]:
```
'''A linear regression model is implemented to predict movie ratings based on their scores.
 The dataset is split into training and testing sets using a 70:30 split ratio.'''
```

[ ]: 'A linear regression model is implemented to predict movie ratings based on their scores.\nThe dataset is split into training and testing sets using a 70:30 split ratio.'

[ ]:
```
#LINE OF Best Fit

    # Assuming x_train and y_train are your data

    x_train = np.array(x_train).reshape(-1, 1) # Reshaping if x_train is a 1D array

    # Create a linear regression model
    model = LinearRegression()

    # Fit the model to the data
    model.fit(x_train, y_train)

    # Predict y values for the x_train data
    y_pred = model.predict(x_train)

    # Plot the original data points
    plt.scatter(x_train, y_train, color='blue')

    # Plot the line of best fit
    plt.plot(x_train, y_pred, color='green')
```

```
plt.show()
```



[ ]:
```
y_pred_slr = slr.predict(x_test)
print('Prediction result:{}'.format(y_pred_slr))
```

[ ]: Prediction result:[7.94745246 8.1660757 8.11141989 8.38469895 8.20251291
8.07498269 8.18429431 8.22073151 8.1478571 8.45757336 8.25716872 8.20251291
8.33004314]

[ ]:
```
#actual value and predicted value
slr_diff = pd.DataFrame({'Actual value': y_test,'Predicted Value': y_pred_slr}) slr_diff
```

[ ]: Actual value Predicted Value

```
#actual value and predicted value
slr_diff = pd.DataFrame({'Actual value': y_test,'Predicted Value': y_pred_slr})
slr_diff
```

	Actual value	Predicted Value
35	7.7	7.947452
25	8.7	8.166076
31	8.3	8.111420
9	8.9	8.384699
36	7.9	8.202513
28	8.1	8.074983
2	8.1	8.184294
21	8.1	8.220732
22	7.7	8.147857
24	8.6	8.457573
14	7.4	8.257169
8	8.1	8.202513
37	8.2	8.330043

[ ]:
```
#predict for any value
slr.predict([[13]])
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have
valid feature names, but LinearRegression was fitted with feature names warnings.warn(
```

[ ]: array([6.90899207])

[ ]:
```
#R squared value
from sklearn.metrics import accuracy_score
print('R squared value: {:2f}'.format(slr.score(x,y)*100))
```

```
R squared value: 12.534178
```

[ ]:
```
# Calculate the Mean Absolute Error (MAE) between the actual and predicted values
MeanAbsErr = metrics.mean_absolute_error(y_test, y_pred_slr)

# Calculate the Mean Squared Error (MSE) between the actual and predicted values
MeanSquErr = metrics.mean_squared_error(y_test, y_pred_slr)
```

```
# Calculate the Root Mean Squared Error (RMSE) between the actual and predicted values
RootMeanSqErr = np.sqrt(metrics.mean_squared_error(y_test, y_pred_slr))

# Display the calculated error metrics with precision up to three decimal places print('Absolute
Mean error:', round(MeanAbsErr, 3))
print('Mean Square error:', round(MeanSquErr, 3))
print('Root Mean Square error:', round(RootMeanSqErr, 3))
```

```
Absolute Mean error: 0.284
Mean Square error: 0.134
Root Mean Square error: 0.366
```

CONCLUSION: The data from the IMDB Top 250 movies was successfully analyzed and visualized. Database operations were performed to store and query the data. A linear regression model was initiated for predicting movie ratings based on their scores.

Overall, the project gives a comprehensive overview of how to handle, process, and analyze data from a real-world dataset. The combination of data visualization, database operations, and the beginning of predictive modeling offers a solid foundation for further enhancements and deeper analyses.