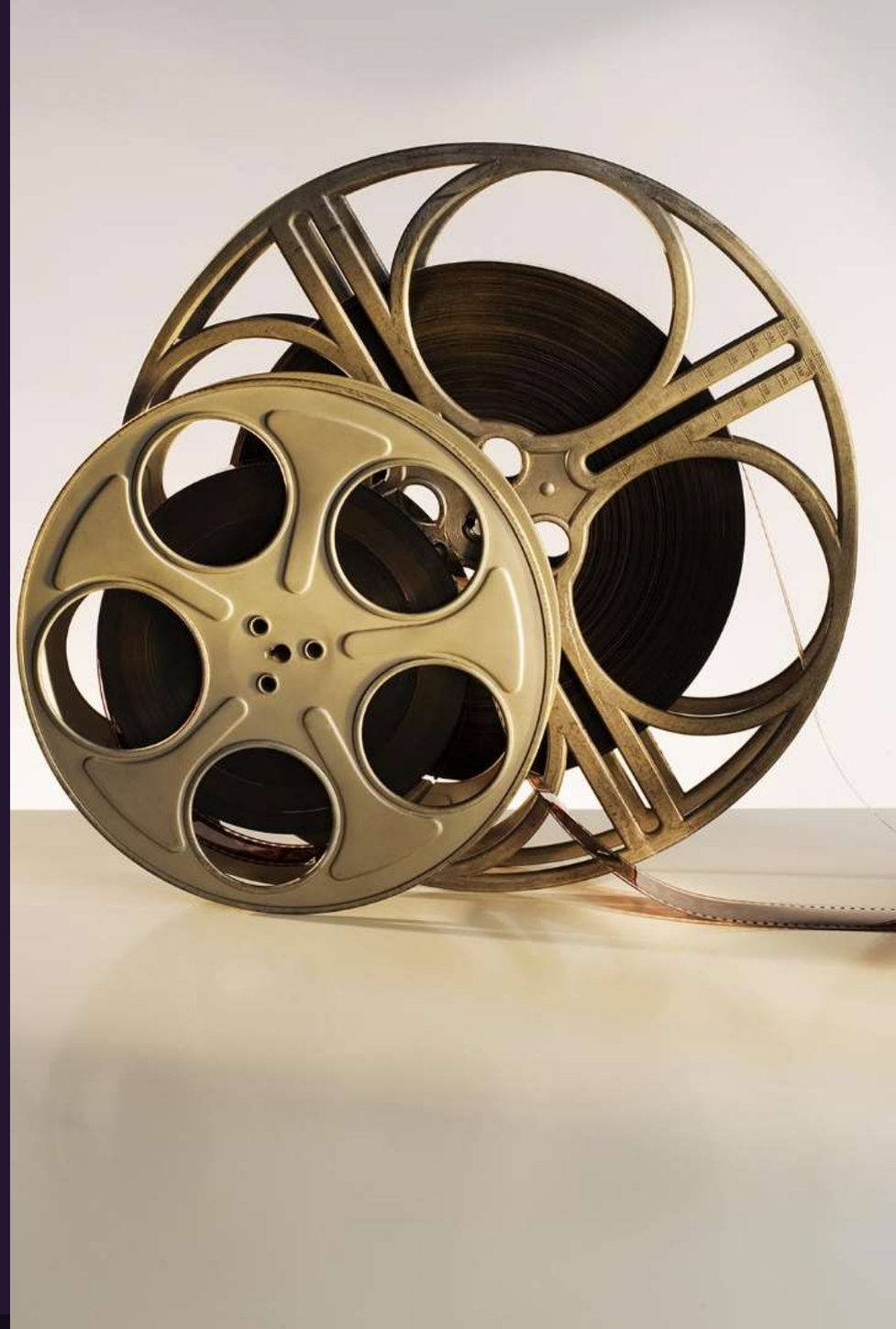
APPLIED DATA SCIENCE

uilding an IMDb core Prediction odel

ady to dive into the fascinating world of IMDb score prediction. We plore the process of loading and preprocessing the dataset to build st prediction model.



Introduction to the Project

Purpose of the Project

Understand the factors that influence IMDb scores to predict the success of future movies.

Importance of IMDb Score Prediction

Accurate predictions can help filmmakers, studios, and audiences make informed decisions.

Loading the Dataset

Exploration of a comprehensive movie dataset to gain valuable insights.

Preprocessing the Data

1

Steps involved in Data Preprocessing

From data cleaning to feature engineering, a comprehensive approach to ensure data quality.

2

Cleaning the Dataset

Removing inconsistencies, correcting errors, and handling duplicate entries.

3

Handling Missing Values and Outliers

Strategies to impute missing data and identify and treat outliers effectively.

1. Display the first 5 rows of the dataset

```
print("1. First 5 rows of the dataset:")
print(df.head())
```

```
1. First 5 rows of the dataset:
            Title                  Genre  ... IMDB Score         Language
0  Enter the Anime            Documentary  ...        2.5  English/Japanese
1     Dark Forces               Thriller  ...        2.6          Spanish
2         The App  Science fiction/Drama  ...        2.6          Italian
3   The Open House        Horror thriller  ...        3.2          English
4     Kaali Khuhi                Mystery  ...        3.4            Hindi

[5 rows x 6 columns]
```

2. Display basic statistics for the IMDb Score column

```
print("\n2. Basic statistics for IMDb Score:")
print(df['IMDB Score'].describe())
```

```
2. Basic statistics for IMDb Score:
count    584.000000
mean       6.271747
std        0.979256
min        2.500000
25%        5.700000
50%        6.350000
75%        7.000000
max        9.000000
Name: IMDB Score, dtype: float64
```

. Number of movies in each genre

```
_counts = df['Genre'].value_counts()
"\n3. Number of movies in each genre:")
genre_counts)
```

```
3. Number of movies in each genre:
Genre
Documentary                              159
Drama                                     77
Comedy                                    49
Romantic comedy                           39
Thriller                                  33
                                        ...
Romantic comedy-drama                      1
Heist film/Thriller                        1
Musical/Western/Fantasy                    1
Horror anthology                           1
Animation/Christmas/Comedy/Adventure       1
Name: count, Length: 115, dtype: int64
```

4. Average IMDb score by genre

```
enre_avg_scores = df.groupby('Genre')['IMDB Score'].mean()
rint("\n4. Average IMDb score by genre:")
rint(genre_avg_scores)
```

```
Genre
Action                5.414286
Action comedy         5.420000
Action thriller       6.400000
Action-adventure      7.300000
Action-thriller       6.133333
                          ...
War                   6.750000
War drama             7.100000
War-Comedy            6.000000
Western               6.066667
Zombie/Heist          5.900000
Name: IMDB Score, Length: 115, dtype: float64
```

5. Movie with the highest IMDb score

```
max_imdb_score = df[df['IMDB Score'] == df['IMDB Score'].max()]
print("\n5. Movie with the highest IMDb score:")
print(max_imdb_score)
```

```
5. Movie with the highest IMDb score:
                                    Title        Genre  ... IMDB Score  Language
583  David Attenborough: A Life on Our Planet  Documentary  ...        9.0   English

[1 rows x 6 columns]
```

6. Movie with the lowest IMDb score

```
min_imdb_score = df[df['IMDB Score'] == df['IMDB Score'].min()]
print("\n6. Movie with the lowest IMDb score:")
print(min_imdb_score)
```

```
6. Movie with the lowest IMDb score:
            Title        Genre  ... IMDB Score          Language
0  Enter the Anime  Documentary  ...        2.5  English/Japanese

[1 rows x 6 columns]
```

***

7. Average IMDb score of movies in English language

```
english_avg_score = df[df['Language'] == 'English']['IMDB Score'].mean()
print("\n7. Average IMDb score of movies in English language:")
print(english_avg_score)
```

```
7. Average IMDb score of movies in English language:
6.38004987531172
```

***

8. Number of movies in each language

```
language_counts = df['Language'].value_counts()
print("\n8. Number of movies in each language:")
print(language_counts)
```

```
8. Number of movies in each language:
Language
English                         401
Hindi                            33
Spanish                          31
French                           20
Italian                          14
Portuguese                       12
Indonesian                        9
Japanese                          6
Korean                            6
German                            5
Turkish                           5
English/Spanish                   5
Polish                            3
Dutch                             3
Marathi                           3
English/Hindi                     2
Thai                              2
English/Mandarin                  2
English/Japanese                  2
Filipino                          2
English/Russian                   1
Bengali                           1
English/Arabic                    1
English/Korean                    1
Spanish/English                   1
Tamil                             1
English/Akan                      1
Khmer/English/French              1
Swedish                           1
Georgian                          1
Thia/English                      1
English/Taiwanese/Mandarin        1
English/Swedish                   1
Spanish/Catalan                   1
Spanish/Basque                    1
Norwegian                         1
Malay                             1
English/Ukranian/Russian          1
Name: count, dtype: int64
```

9. Movies with IMDb score above 8.0

```
high_rated_movies = df[df['IMDB Score'] > 8.0]
print("\n9. Movies with IMDb score above 8.0:")
print(high_rated_movies)
```

```
9. Movies with IMDb score above 8.0:
                                          Title  ...                 Language
568                                Chasing Coral  ...                  English
569                            My Octopus Teacher  ...                  English
570                                Rising Phoenix  ...                  English
571                                          13th  ...                  English
572           Disclosure: Trans Lives on Screen  ...                  English
573                                        Klaus  ...                  English
574                                    Seaspiracy  ...                  English
575         The Three Deaths of Marisela Escobedo  ...                  Spanish
576                       Cuba and the Cameraman  ...                  English
577                        Dancing with the Birds  ...                  English
578   Ben Platt: Live from Radio City Music Hall  ...                  English
579         Taylor Swift: Reputation Stadium Tour  ...                  English
580  Winter on Fire: Ukraine's Fight for Freedom  ...  English/Ukranian/Russian
581                       Springsteen on Broadway  ...                  English
582    Emicida: AmarElo - It's All For Yesterday  ...               Portuguese
583     David Attenborough: A Life on Our Planet  ...                  English
```

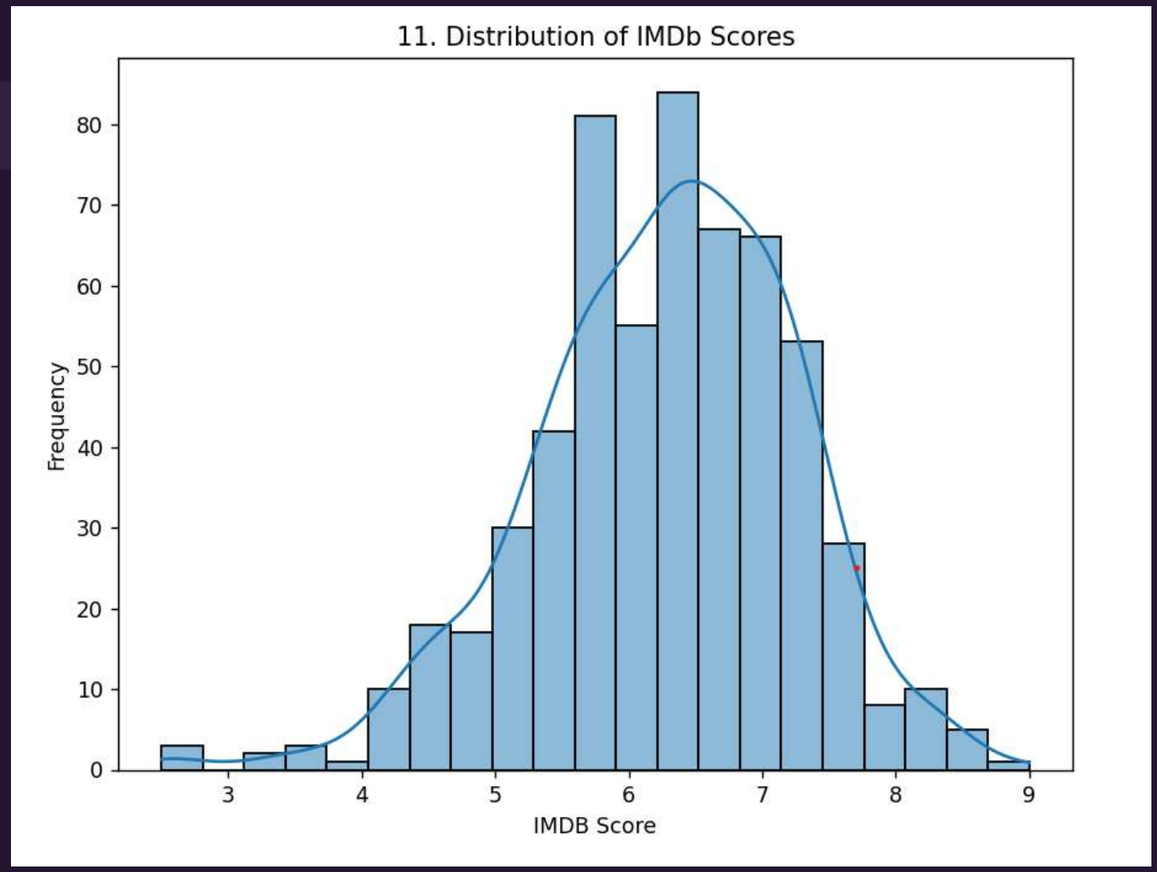
10. Correlation between IMDb score and Runtime

```
correlation = df['IMDB Score'].corr(df['Runtime'])
print("\n10. Correlation between IMDb score and Runtime:", correlation)
```

```
-0.040896291420788 59
```

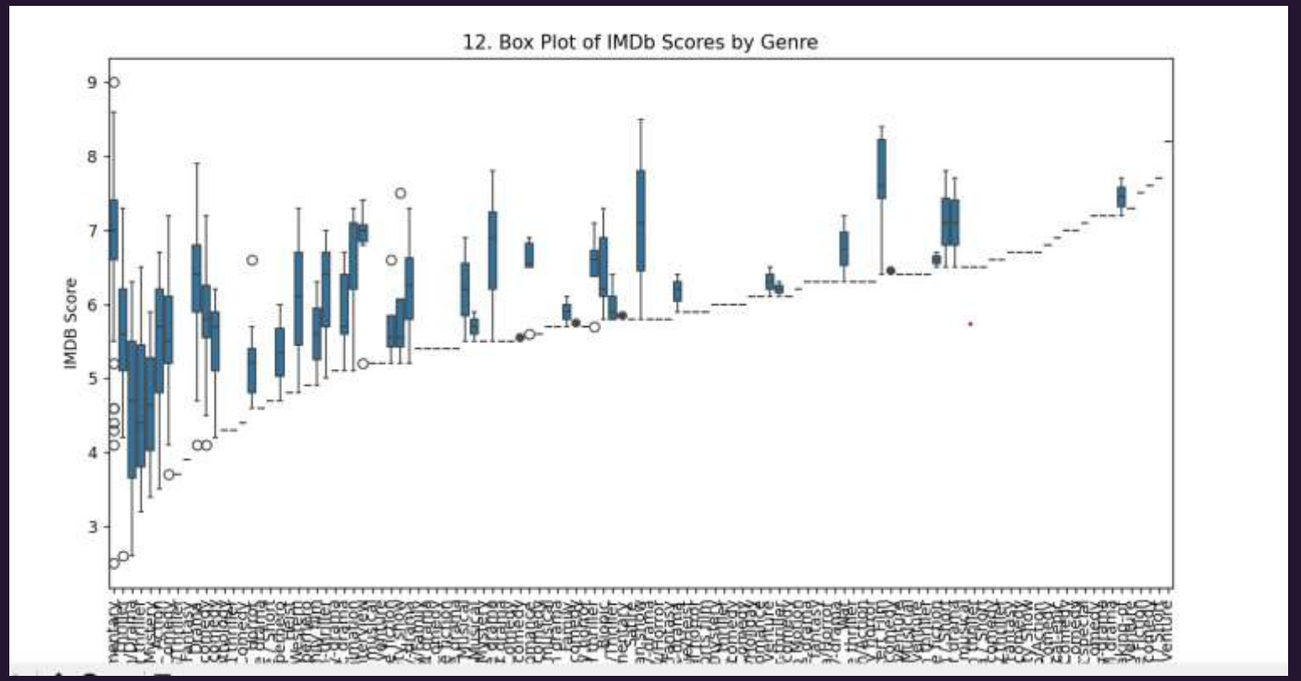
11. Distribution plot of IMDb scores

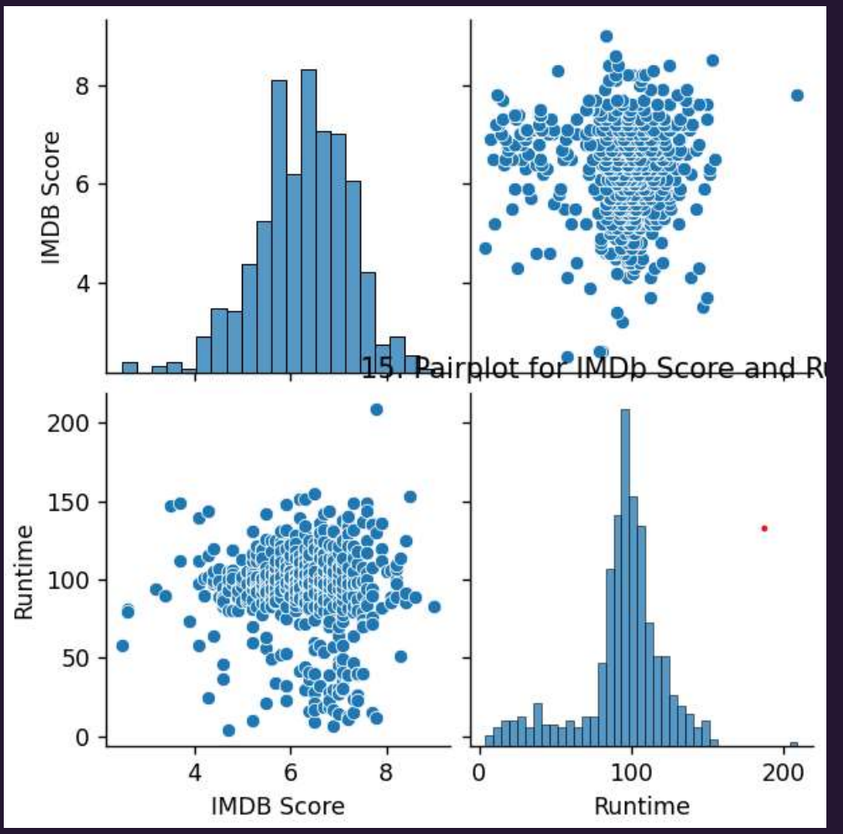
```
plt.figure(figsize=(8, 6))

sns.histplot(df['IMDB Score'], kde=True)
plt.title("11. Distribution of IMDb Scores")
plt.xlabel("IMDB Score")
plt.ylabel("Frequency")
plt.show()
```



```
# 12. Box plot of IMDb scores by genre
plt.figure(figsize=(12, 6))
sns.boxplot(x='Genre', y='IMDB Score', data=df)
plt.title("12. Box Plot of IMDb Scores by Genre")
plt.xticks(rotation=90)
plt.show()

# 13. Genre with the highest average IMDb score
genre_highest_avg = genre_avg_scores.idxmax()
print("\n13. Genre with the highest average IMDb score:", genre_highest_avg)

# 14. Movie with the highest IMDb score for each genre
top_movie_by_genre = df.groupby('Genre')['IMDB Score'].idxmax()
print("\n14. Movie with the highest IMDb score for each genre:")
print(df.loc[top_movie_by_genre, ['Genre', 'Title', 'IMDB Score']])

# 15. Pairplot for IMDb score and Runtime
sns.pairplot(df[['IMDB Score', 'Runtime']])
plt.title("15. Pairplot for IMDb Score and Runtime")
plt.show()
```

***





15. Pairplot for IMDb Score and R

***

14. Movie with the highest IMDb score for each genre:

	Genre	Title	IMDB Score
372	Action	Extraction	6.7
257	Action comedy	Spenser Confidential	6.2
318	Action thriller	Wheelman	6.4
507	Action-adventure	Okja	7.3
450	Action-thriller	The Night Comes for Us	7.0
..	...	...	...
497	War	The Siege of Jadotville	7.2
553	War drama	Beasts of No Nation	7.7
219	War-Comedy	War Machine	6.0
516	Western	The Ballad of Buster Scruggs	7.3
194	Zombie/Heist	Army of the Dead	5.9

[115 rows x 3 columns]

Building the IMDb Score Prediction Model

Overview of the Prediction Model

An explanation of the approach and techniques used to predict IMDb scores accurately.

Choice of Machine Learning Algorithm

Selection of the most suitable algorithm to achieve optimal prediction accuracy.

Model Training and Evaluation

Discussion of the model training process, hyperparameter tuning, and evaluation metrics.

Conclusion

1 Summary of the Presentation

A recap of the key points covered in this presentation on building an IMDb score prediction model.

2 Importance of Dataset Loading and Preprocessing

An emphasis on the crucial role dataset loading and preprocessing play in accurate predictions.

3 Next Steps in the Project

Exciting directions to explore, such as model optimization and deployment.