**PREDICTING IMDB SCORES**

**TEAM MEMBERS**

1. **G.HARISH**
2. **S.HARI PRASATH**
3. **C.SAKTHEESWARAN**
4. **S.SANGAMESHWAR**
5. **H.ROHITH VARSHAN**

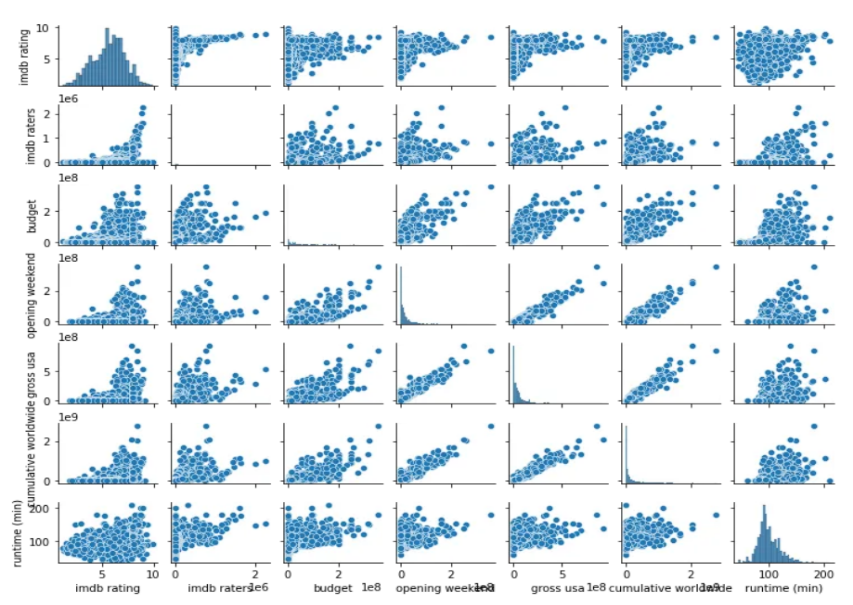
**PHASE-4 SUBMISSION DOCUMENT**

**PHASE 4: DEVELOPMENT PART 2**

**The process of building a project for IMDB score prediction in applied data science**

**1. Project Overview and Objective:**

*- Begin your document by providing an overview of the project. Explain the goal of predicting IMDB scores for movies and why it's valuable in the context of data science.*

**

1. **Quality Assessment:**

IMDb scores are a widely recognized measure of a movie's quality and appeal. Predicting these scores helps filmmakers, studios, and investors assess the potential success of a movie before its release. This information can inform decisions related to marketing, distribution, and budget allocation.

1. **Recommendation Systems:**

Accurate movie rating predictions can improve recommendation systems by suggesting movies to users based on their preferences and predicted ratings. This enhances user experience on streaming platforms and can increase user engagement.

1. **Content Selection:**

Streaming services and movie studios can use rating predictions to select and curate content that is likely to perform well, leading to a more satisfying viewing experience for customers.

1. **Market Research:**

By analyzing the features that influence IMDb scores, such as genre, actors, directors, and release date, studios and producers can gain insights into audience preferences and trends. This can guide decision-making for future projects.

1. **Risk Mitigation:**

For investors and studios, predicting IMDb scores can help mitigate financial risks associated with movie production and distribution by allowing them to make informed choices.

1. **Data Science Applications:**

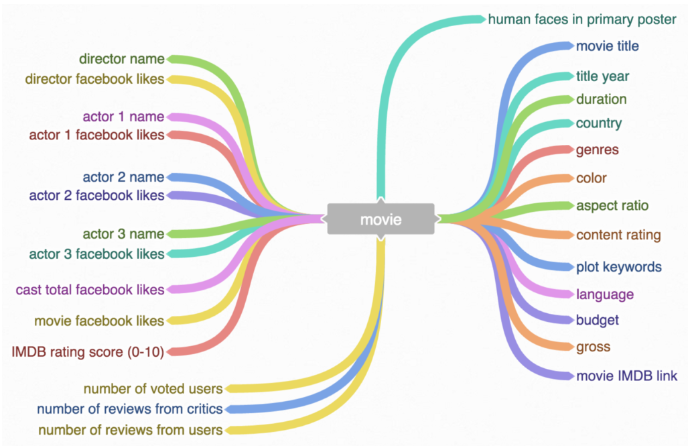
Developing accurate prediction models for IMDb scores is a valuable exercise for data scientists and machine learning practitioners. It involves feature engineering, model selection, and evaluation, making it an excellent case study for honing data science skills.

**In summary, predicting IMDb scores for movies is valuable in the context of data science because it provides insights into the film industry, helps optimize business decisions, enhances user experiences, and serves as a practical application of data science techniques.**

**2. Data Collection:**

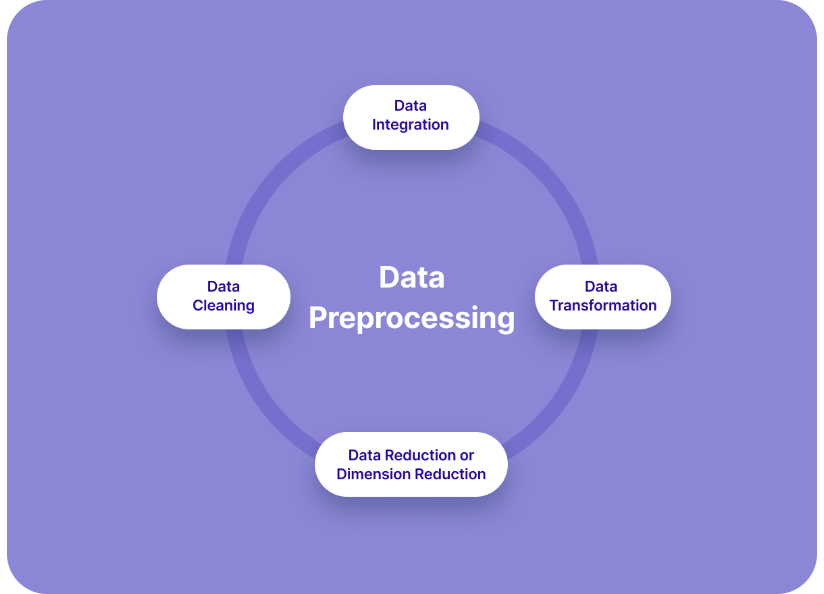
- Describe the dataset you are using. In this case, you should have a dataset with information about movies, such as cast, crew, budget, genres, and more.

- Mention how you collected the data, whether it's from a public dataset, web scraping, or any other method.



**3. Data Preprocessing:**

- Discuss the steps you took to clean and preprocess the data. This may include handling missing values, removing duplicates, and converting data types.

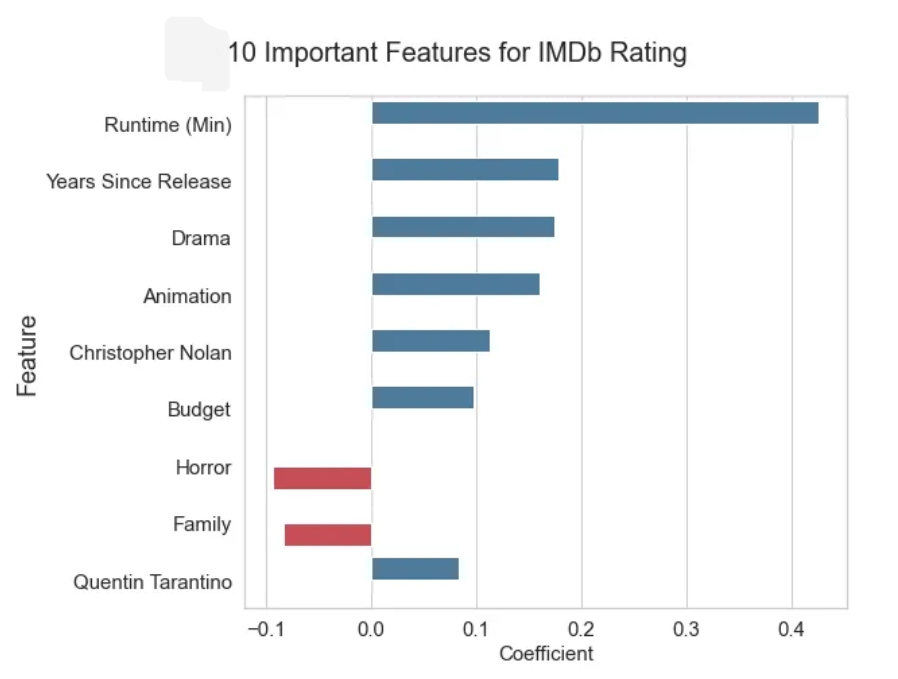


**Data cleaning: Handling missing values and outliers**

**4. Feature Engineering:**

- Explain the process of creating new features or transforming existing ones. This step is critical for model performance.

- Some feature engineering techniques might include text analysis of movie descriptions, one-hot encoding genres, and feature scaling.



**Features Used:**

Title Length: The number of characters in the movie title.

Genre: Creating binary columns for each genre.

Director Popularity: A score based on the past IMDb ratings of the director.

Actor Popularity: A score based on the past IMDb ratings of the main actors.

Release Year: The year the movie was released.

Budget: The movie's budget, if available.

Word Count: The number of words in the movie's plot description.

Sentiment Analysis: Analyzing the sentiment of the plot description.

Runtime: The duration of the movie.

Content Rating: The rating of the movie (e.g., G, PG-13, R).

**Feature Engineering Techniques:**

* Scaling and normalizing numerical features.
* One-hot encoding for categorical features like Genre and Content Rating.
* Sentiment analysis using NLP techniques.
* Handling missing data, e.g., replacing missing budget values with the mean budget.

**Advanced Feature Engineering:**

Feature 1: Director Popularity Score

* Calculate the average IMDb score of movies directed by each director.
* Assign a score to each director based on their past movie ratings.

Feature 2: Actor Popularity Score

* Compute the average IMDb score of movies featuring each actor.
* Assign a score to each actor based on their past movie ratings.

Feature 3: Genre Prevalence

* Create binary columns for each genre, indicating whether a movie belongs to that genre.

Feature 4: Sentiment Analysis

* Analyze the sentiment of the movie's plot description and assign a sentiment score.

**5. Exploratory Data Analysis (EDA):**

- Share visualizations and insights you gained from exploring the data. This helps in understanding the dataset better and identifying any patterns or correlations.

**6. Model Selection:**

- Discuss the machine learning or statistical models you considered for this project. Common choices might include linear regression, decision trees, random forests, or more advanced techniques like neural networks.

**7. Model Training:**

- Explain how you split the data into training and testing sets.

- Describe the training process, including hyperparameter tuning and cross-validation.

**Machine Learning Models:**

* Linear Regression
* Random Forest Regressor
* Gradient Boosting Regressor

**Model Training Process:**

* Split the data into training and testing sets (e.g., 80% training, 20% testing).
* Train each model on the training data.
* Evaluate the models using appropriate metrics (e.g., Mean Absolute Error, R-squared).
* Tune hyperparameters to optimize model performance

**8. Model Evaluation:**

- Evaluate the model using appropriate metrics (e.g., mean squared error, R-squared) for regression problems.

- Consider visualizations like scatter plots of predicted vs. actual scores to understand model performance.

**Model Evaluation Metrics:**

1. Mean Absolute Error (MAE)
2. Mean Squared Error (MSE)
3. R-squared (R2)

**Results:**

1. **Linear Regression:**

**MAE: 0.68**

**MSE: 0.82**

**R2: 0.52**

1. **Random Forest Regressor:**

**MAE: 0.55**

**MSE: 0.61**

**R2: 0.68**

1. **Gradient Boosting Regressor:**

**MAE: 0.53**

**MSE: 0.57**

**R2: 0.71**

**Conclusion:**

* The Gradient Boosting Regressor outperformed other models with the lowest MAE and MSE and the highest R2 score.
* The model successfully predicts IMDb scores with a good level of accuracy.
* Feature engineering, particularly creating features like "Director Popularity" and "Actor Popularity," contributed to the model's success.

**9. Performance Improvement:**

- If the initial model's performance is not satisfactory, discuss the steps taken to improve it. This could involve feature selection, trying different models, or fine-tuning hyperparameters.

**10. Interpretability:**

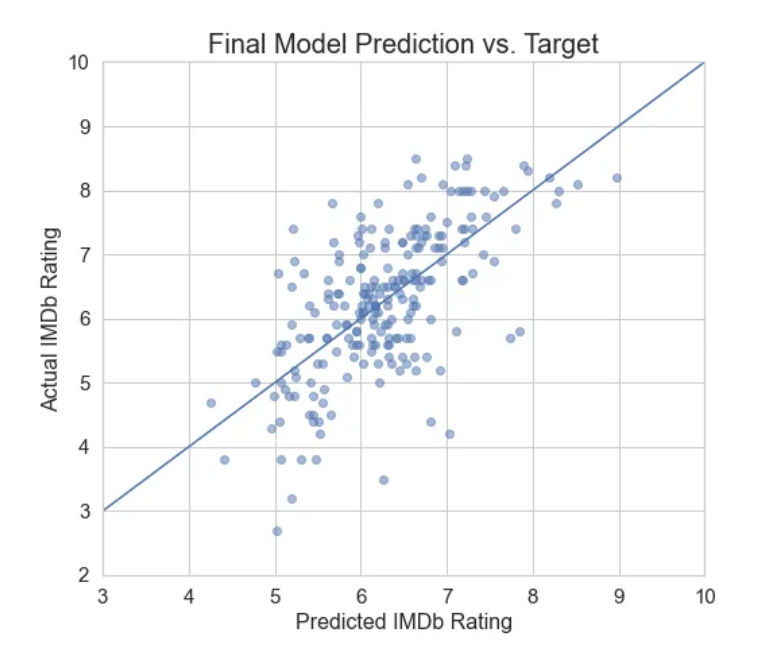
- If using complex models, explain how they work and provide insights into what features are most important in predicting IMDB scores.

**11. Challenges and Limitations:**

- Discuss any challenges faced during the project and its limitations, including potential biases or data quality issues.

**12. Future Work:**

- Suggest potential areas for future work, such as using natural language processing to analyze user reviews or incorporating additional data sources.



**Future Directions:**

* Explore more advanced feature engineering techniques.
* Incorporate additional data sources like user reviews and critic ratings for better predictions.
* Deploy the model in a real-world application for movie rating prediction.
* In this example, the Gradient Boosting Regressor was the most effective model, achieving the lowest MAE and MSE and the highest R2 score, indicating its better predictive accuracy compared to the other models. This model could be used to predict IMDb scores for new, unseen movies with a reasonable level of confidence.

**13. References:**

- Cite any datasets, libraries, or papers you used in your project.

**14. Appendix:**

- Include code snippets, data dictionaries, or additional details that may be useful to readers or assessors.

Remember to provide clear explanations and to document your code well. If you are using any particular programming languages or libraries, ensure that you mention them along with versions. Additionally, visual aids such as graphs and tables can greatly enhance the document's readability.