**‘Cricket Match Analysis and Prediction’ Report**

**GitHub Link:** <https://github.com/abhishekgoud23/Match-prediction-and-analysis>

**Project Description:**

The goal of this project was to build a binary classification model to predict the winner of cricket matches based on various features such as team performance, venue, player statistics, etc. The dataset used for this project consisted of historical cricket match data, including both domestic and international matches. Here we used a dataset from Kaggle. It consists of all the details and statistics of all the years in the Indian Premier League (IPL).

Dataset link: <https://www.kaggle.com/datasets/rajsengo/indian-premier-league-ipl-all-seasons>

**Data Cleaning:**

The dataset was thoroughly examined to identify any missing values, duplicates, or irrelevant columns. Missing values were either imputed or removed, depending on the nature of the data. Irrelevant columns that did not contribute to the prediction task were dropped from the dataset. Our potential features include home and away teams, venue, toss decision, innings wise scores. Furthermore, the remaining columns were encoded into numeric form to be used for model building.

**Feature Engineering:**

To enhance the predictive power of the model, several new features were engineered based on the existing data. For instance, metrics such as run rate, economy rate, score difference, and wicket difference between the two teams were calculated. Additionally, categorical features were one-hot encoded to transform them into a numerical format, making them suitable for model training.

**Feature Selection:**

Feature selection is a crucial step in machine learning projects. In this project, several feature engineering techniques were employed to select informative features, including encoding categorical variables (e.g., home team, away team, toss won, venue, home captain) and creating new features (e.g., home run rate, away run rate, home economy rate, away economy rate, score difference, wicket difference).

**Model Building:**

As this was a binary classification problem, multiple modeling techniques were utilized to predict the outcome of cricket matches. The models employed were:

1. Logistic Regression

2. Random Forest Classifier

3. Gradient Boosting Classifier

4. Support Vector Classifier

**Model Evaluation:**

To evaluate the performance of the predictive models, various evaluation metrics were employed, including accuracy, precision, recall, and F1-score. These metrics provide insights into how well the models are performing in terms of correctly predicting the winner of cricket matches.

**Results**:

After evaluating the models, the following results were obtained:

- Logistic Regression:

Accuracy: 0.9372

Precision: 0.9387

Recall: 0.9372

F1-score: 0.9369

- Support Vector Classifier:

Accuracy: 0.9631

Precision: 0.9633

Recall: 0.9641

F1-score: 0.9630

- Gradient Boosting Classifier:

Accuracy: 0.9741

Precision: 0.9743

Recall: 0.9741

F1-score: 0.9741

- Random Forest Classifier:

Accuracy: 0.9889

Precision: 0.9891

Recall: 0.9889

F1-score: 0.9889

Comparing the models based on these metrics, we can observe the following:

Random Forest Classifier has the highest accuracy, precision, recall, and F1-score among all the models. It achieves an accuracy of 0.9889, indicating that it predicts the winner of cricket matches with high accuracy.

Gradient Boosting Classifier also performs well with an accuracy of 0.9741 and high precision, recall, and F1-score. It is a reliable model for predicting the outcome of cricket matches.

Support Vector Classifier demonstrates good performance with an accuracy of 0.9631 and high precision, recall, and F1-score. It can be considered as a suitable model for this classification task.

Logistic Regression, while having slightly lower accuracy compared to the other models, still achieves respectable results with an accuracy of 0.9372 and consistent precision, recall, and F1-score.

The Random Forest Classifier stands out as the best-performing model based on the evaluation metrics, followed by the Gradient Boosting Classifier and Support Vector Classifier. However, the choice of the most suitable model depends on the specific requirements, constraints, and trade-offs of the project. Further analysis and comparison can be performed to select the optimal model for deployment.

**Conclusion**:

In conclusion, this project successfully utilized machine learning techniques to predict the outcome of cricket matches. The models achieved high accuracy and demonstrated the potential for using these methods in the analysis and prediction of cricket match results. The findings from this project can be valuable for cricket teams, fantasy sports industries, and enthusiasts looking to gain insights from historical cricket match data. While the models performed well, there is still room for further improvement and exploration of more advanced techniques in future iterations of the project.