**IPL WIN PREDICTION**

**Introduction**

This dataset captures match-level information from IPL tournaments over multiple seasons. It includes team details, match results, toss outcomes, venues, and performance indicators. It is designed to be used for match outcome analysis and winner prediction.

**Aim**

The aim of this project is to predict the winner of an IPL match based on various features such as the teams playing, toss winner, toss decision, and other match-related factors using different machine learning models like Logistic Regression, SVM, KNN, Decision Trees, Random Forest, and XGBoost. The best model must be selected after evaluating the performance and hyperparameters must be tuned for further improvements.

**About the Dataset**

The IPL dataset contains detailed information about cricket matches played in the Indian Premier League (IPL). Key columns include:

* **Season:** The year of the IPL season.
* **City:** The city where the match was played.
* **Date:** The date of the match.
* **team1 and team2:** The two teams playing in the match.
* **toss\_winner and toss\_decision:** The team that won the toss and the decision they made (whether to bat or bowl).
* **result:** The outcome of the match (win or loss).
* **dl\_applied:** Indicates if the Duckworth-Lewis method was applied due to interruptions.
* **winner:** The team that won the match.
* **win\_by\_runs and win\_by\_wickets:** The margin of victory by runs or wickets.
* **player\_of\_match:** The player awarded for exceptional performance.
* **venue:** The stadium where the match was held.
* **umpires:** The officials overseeing the match.

**Data Understanding**

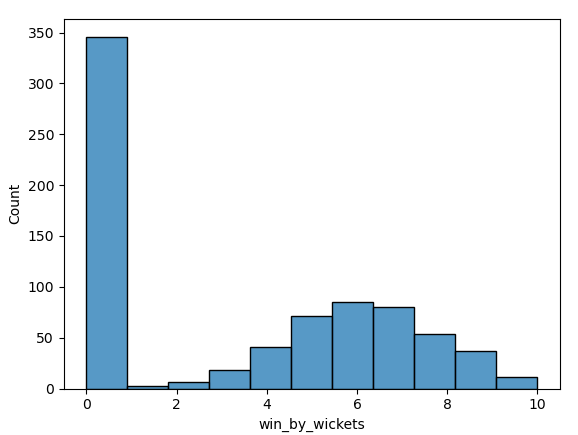
Initially, starting with loading the dataset using pandas after importing libraries. Finding the structure of the dataset. Also, viewing the starting and ending vales of the dataset. Then describing the dataset for finding the descriptive statistics.

**Data Cleaning**

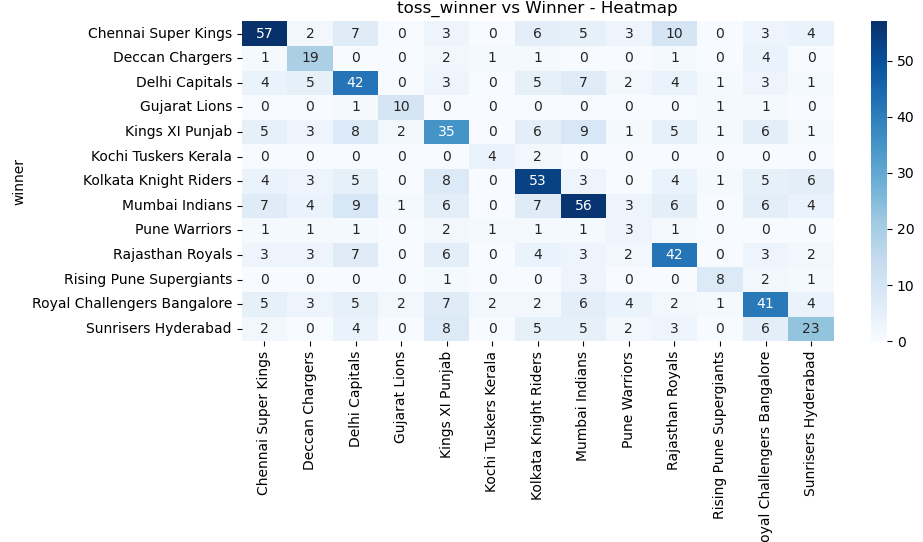
In cleaning, handling the null values by finding them with their percentage and according to the percentage, decided whether to remove or to replace them with the appropriate values. After that checking and retaining the consistency of the values. Finally, checking for any duplicates. Then save them in an excel file.

**Exploratory Data Analysis (EDA)**

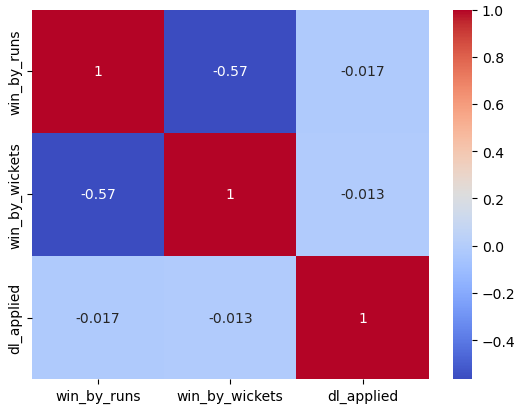
**Univariate Analysis:** Here I have shown the distribution of each column values. I have got the output as shown below.



**Bivariate Analysis:** Here, I have compared the specific variables with the target variable which is the “winner”. The output is shown below.



**Multivariate Analysis:** Here, I have compared each and every other numerical value column by visualizing using heatmap. The output is shown below.



**Data Preprocessing**

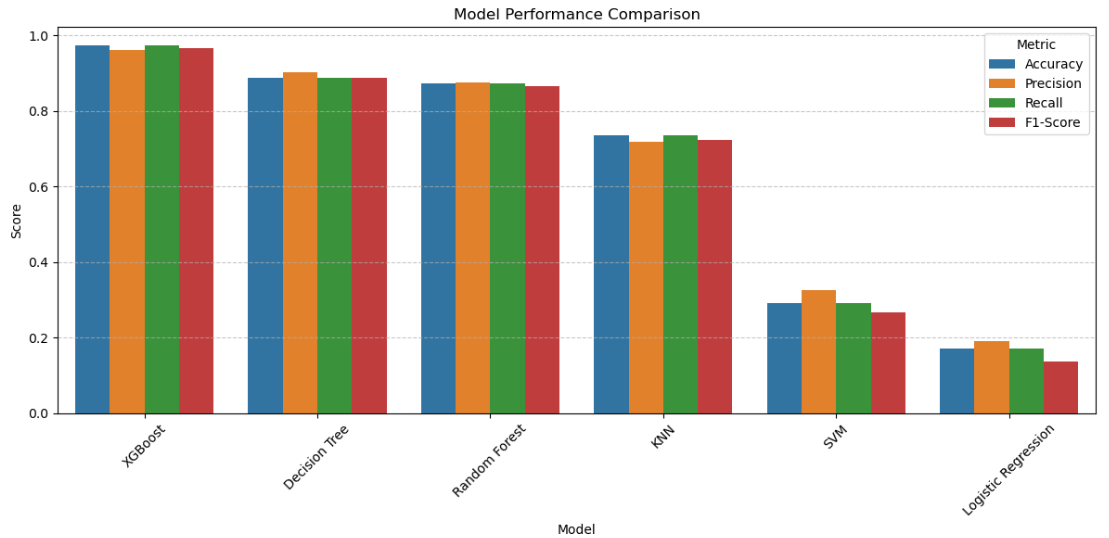
In preprocessing, initiated with finding the outliers by visualizing as well as using IQR method. Though it has heavy outlier values, so I just go with replacing the values using log-transform. Then, for encoding the categorical values to numerical values, I used both LabelEncoding and One-Hot Encoding. After that to standardize the values in win\_by\_run and win\_by\_wicket, I used standard scaling. Then fixing the features and target variables. Finally, splitting the dataset into training and testing with the ratio of 80:20.

**Modeling Process**

I have trained under 6 models, they are: Logistic Regression, SVM, KNN, Random Forest, Decision Tree and XGBoost. For each model I have calculated the evaluation metrices (like Accuracy, Precision, F1 Score, Recall) and model evaluation by Confusion Matrix and Classification Report. Then, hyperparameter tuning was done for which the model got higher accuracy.

**Model Comparison**

Under model comparison, I have compared all the 6 trained models based on their accuracy to predict the best performance model. I visualize them using the bar plot is shown below. Finally, saving them in the pickle (pkl) file. After that to maintain a report, I have saved the output in the txt file, which consisting of best model, best hyperparameters, their accuracy, precision rate, F1 score, recall value, confusion matrix and classification report.



**Conclusion**

The IPL Match Winner Prediction model was successfully developed using historical match data. The dataset underwent essential pre-processing steps including feature selection, encoding categorical variables, and data splitting. Machine learning models such as Logistic Regression, Random Forest, or other classifiers were trained and evaluated on this data.

The final model achieved reasonable accuracy on the test set, which is **XG Boosting** model has **97.35%** indicating its capability to generalize to unseen matches. Feature importance analysis also highlighted key variables (e.g., toss winner, team batting first, venue) that significantly influence match outcomes.

**Future Steps**

* **Feature Engineering**
  + Incorporate player-level statistics (form, fitness, strike rate, economy rate).
  + Include contextual factors like weather, pitch type, and match pressure (e.g., knockout stage).
* **Time Series Consideration**
  + Model performance trends over seasons to capture team evolution or form fluctuations.
* **Live Match Updates**
  + Enhance the model to use live match data for real-time win probability prediction.
* **Advanced Models**
  + Explore ensemble methods like XGBoost or neural networks for potentially better performance.
  + Hyperparameter tuning using grid search or Bayesian optimization.