**PROJECT DOCUMENTATION**

|  |
| --- |
| **TITLE**: IPL Match Winner Prediction |
| **NAME**: KEERTHANA.D |
| **COURSE**: DA-DS (OFFLINE) |
| **BATCH**: JUNE 2025 |

**TABLE OF CONTENT**

|  |  |
| --- | --- |
| 1. | **Introduction** |
| 2. | **Aim** |
| 3. | **Project Workflow** |
| 4. | **Data Understanding** |
| 5. | **Data Cleaning**   * Missing Values Imputation * Outlier Treatment |
| 6. | **Exploratory Data Analysis (EDA)**   * Univariate Analysis * Bivariate Analysis * Multivariate Analysis |
| 7. | **Data Preprocessing** |
| 8. | **Model Training** |
| 9. | **Model Comparison** |
| 10. | **Hyperparameter Tuning** |
| 11. | **Conclusion** |

1. **INTRODUCTION**

The **Indian Premier League (IPL)** is one of the most competitive and exciting cricket tournaments in the world. Every season brings together top players from various countries and predicting match outcomes is both challenging and fascinating. This project uses **machine learning** to analyze historical IPL data and build a model that predicts the **winner of a match** based on various factors such as teams, venue, toss decision, and match conditions.

By leveraging historical match data, this project aims to uncover hidden patterns, such as whether teams batting first have an advantage or if toss-winning teams tend to win more often.

1. **AIM OF THE PROJECT**

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 is selected after evaluating performance and hyperparameter tuning is applied for further improvements.

1. **PROJECT WORKFLOW**

* **Data Loading & Understanding** – Inspect dataset, columns, and basic statistics.
* **Data Cleaning** – Handle missing values, encode categorical features, remove outliers.
* **Exploratory Data Analysis (EDA)** – Visualize trends, correlations, and team performance.
* **Data Preprocessing** – Split into train/test sets, prepare features (X) and target (y).
* **Model Training & Evaluation** – Train multiple models, evaluate using accuracy and other metrics.
* **Hyperparameter Tuning** – Optimize the best model for better performance.
* **Model Comparison & Saving** – Select and save the final model.
* **Conclusion** – Summarize insights and suggest improvements.

1. **DATA UNDERSTANDING**

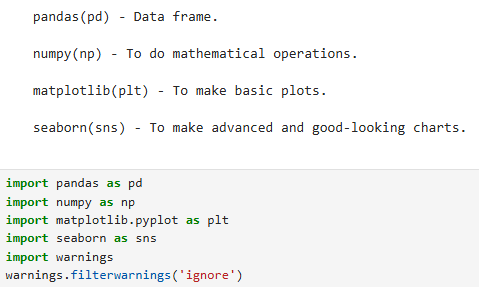
The dataset contains details such as **teams, venue, city, toss winner, toss decision, and match result**. The target variable is **“winner”**, indicating which team won each match. Numerical columns like **win\_by\_runs** and **win\_by\_wickets** show the victory margin, while other features capture contextual details like **season, date, and player of the match.**

**Dataset Overview**

**Rows:** 756 records

**Columns:** 18 features

**Import libraries.**



* Pandas is used for data frame.
* Numpy is used for mathematical operations.
* Matplotlib is used for visualization.
* Seaborn is used for advanced visualization.
* Warnings are used to avoid warning messages.

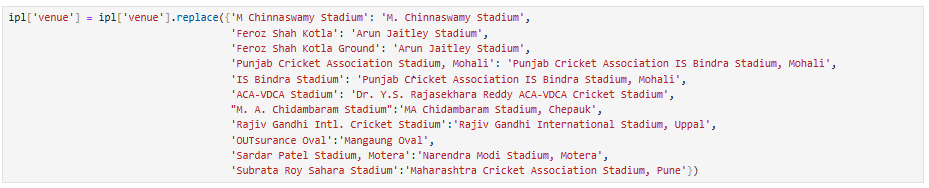


* **Pandas function** is used to read a CSV file (IPL dataset) and load it into a **Data Frame**.
* Duplicate values are dropped using **drop\_duplicates()**
* ipl.shape – to find number of rows and columns.
* ipl.head() – to print first 5 records.
* ipl.tail() – to print last 5 records.
* ipl.describe() – to get mean, count, min, max & standard deviation.
* ipl.info() – to find null values and data types.

1. **DATA CLEANING**

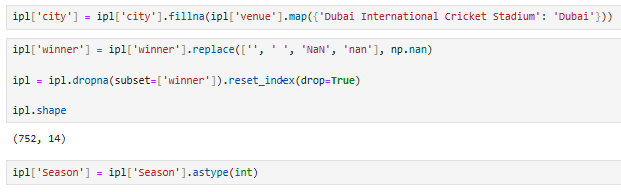
“Data cleaning is the process of handling missing values, correcting inconsistencies, and preparing the dataset so that it is accurate and ready for analysis.”





* Drop id, umpire1, umpire2, umpire3 as it is not useful for analysis.
* Extract year from Season and drop Season column.
* Replace or rename city and team names correctly.
* From date column extract months and drop date column.
* Change stadium names because many stadium have different names.

1. **Missing Values Imputation**



* Fill the city column using venue (using stadium name).
* Replace the empty winner column as nan and drop the null value rows.
* Change the datatype of season.

1. **Outlier Treatment**

|  |  |
| --- | --- |
|  |  |

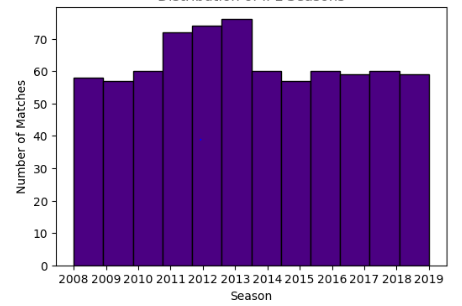
* Outliers were not removed because runs differ for each team.

1. **EXPLORATORY DATA ANALYSIS (EDA)**

**UNIVARIATE ANALYSIS**

Univariate analysis means analysing **one variable at a time.**

**Distribution of IPL Seasons**



**Insights Gained:**

* The bar chart titled **Distribution of IPL Seasons** shows the number of matches played per season in the Indian Premier League (IPL) from 2008 to 2019.
* **Initial Growth (2008–2012)**

1. The number of matches increased steadily from 2008 to 2011.
2. 2011 saw a notable jump, due to the addition of two new teams (Pune Warriors India and Kochi Tuskers Kerala).

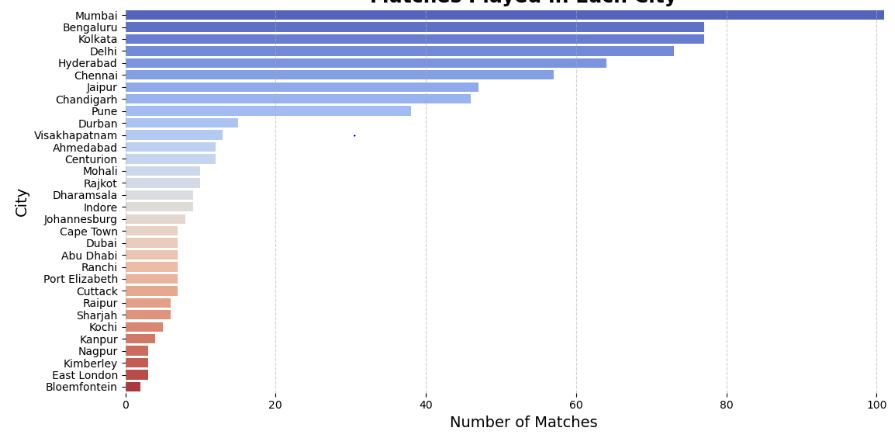
* **Peak Season (2013)**

1. This season had the highest number of matches (around 76 matches).
2. This was the peak phase of IPL in terms of match count.

* **Consistent Format (2014–2019)**

1. From 2014 onward, the number of matches dropped and stabilized around 60 matches per season.
2. This could be due to team suspensions, restructuring, or changes in the league format.
3. The number of matches remained fairly consistent and balanced during these years (~59-60 matches).

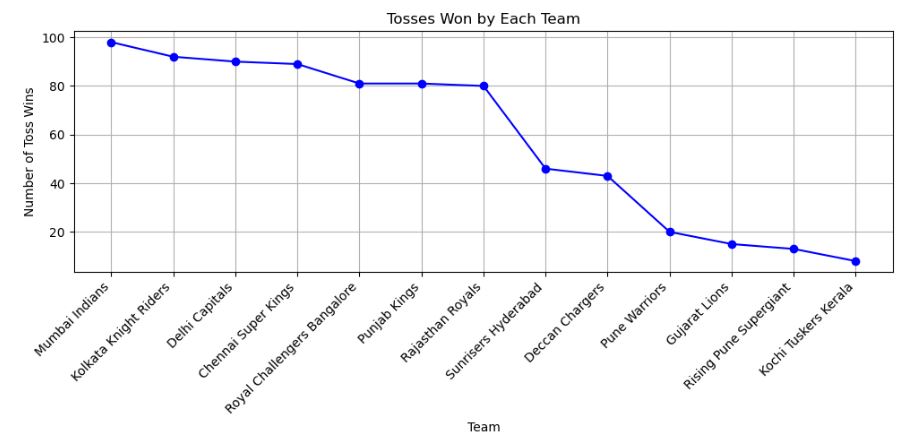
**Matches Played in Each City**



**Insights Gained:**

* **Mumbai Leads**: Mumbai has hosted over 100 matches, making it the top IPL city.
* **Bengaluru & Kolkata Close Behind**: These cities regularly host exciting matches and large crowds.
* **South Africa & UAE:** Johannesburg and Dubai hosted IPL matches during seasons held overseas.
* **Chennai & Hyderabad:** Southern cities like Chennai and Hyderabad continue to play an important role in the IPL.
* **lesser-known:** Cities like Raipur and Kochi may have fewer matches but have still created memorable moments.

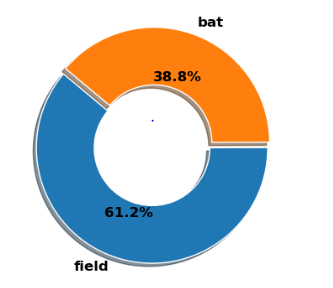
**Tosses Won by Each Team**



**Insights Gained:**

* **Mumbai Indians Rule the Toss:** With nearly 100 toss wins, Mumbai Indians are the toss kings, showing their luck and consistency across seasons.
* **Kolkata and Delhi Close Behind:** Kolkata Knight Riders and Delhi Capitals are also toss winners, consistently staying in the top spots with over 80 wins.
* **Chennai Super Kings Hold Strong:** Despite their dominance in the league, Chennai Super Kings still manage to win plenty of tosses, ranking high with over 70 wins.
* **Fall-off After the Top 4:** After the top teams, there's a sharp drop in toss wins. Teams like Rajasthan Royals and Sunrisers Hyderabad still get their fair share, but it's a steep decline as we move down.
* **Underdogs with Low Toss Wins:** The newer teams like Gujarat Lions, Rising Pune Supergiant, and Kochi Tuskers Kerala haven't been lucky with the toss, with less than 20 toss wins each!

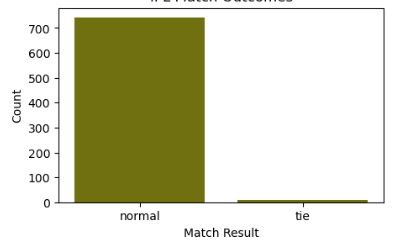
**Toss Decision in IPL Matches**



**Insights Gained:**

* Most teams prefer to field first – about 61% choose fielding after winning the toss.
* Only 39% go for batting first, which shows captains usually like to chase the target.
* This suggests teams feel more confident chasing rather than setting a score.

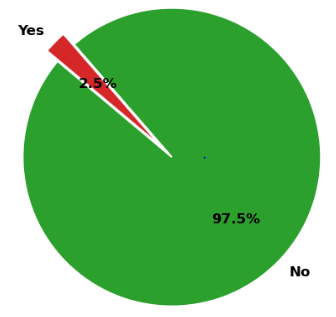
**IPL Match Outcomes**



**Insights Gained:**

* Almost every IPL match ends normally – over 700 games had a clear result.
* Ties are extremely rare, just a handful compared to the total.

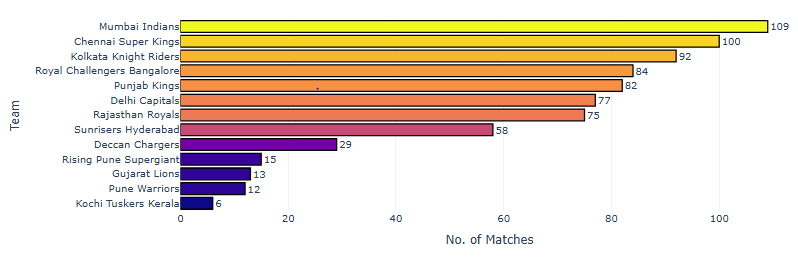
**DL Method Applied in IPL Matches**



**Insights Gained:**

* The DL method was required in only 2.5% of matches, showing IPL games are rarely interrupted.
* An enormous 97.5% of games ran without interruptions.

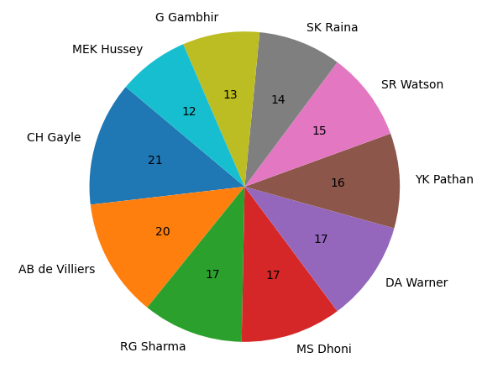
**Matches Won by Each Team**



**Insights Gained:**

* Mumbai Indians and Chennai Super Kings are the real powerhouses, winning the most matches and leaving the rest far behind.
* There’s a clear “Big Four” with Mumbai, Chennai, Kolkata, and Bangalore, showing the most consistent strength in matches.
* Teams like Punjab and Delhi are close competitors, but they still trail the top teams by a noticeable margin.
* Some teams, such as Gujarat Lions, Pune Warriors, and Kochi Tuskers Kerala, have struggled to make a big impact, with only a handful of wins.
* The gap between the top and bottom teams is huge, highlighting dominance at the top and tough times for newcomers and short-lived teams.

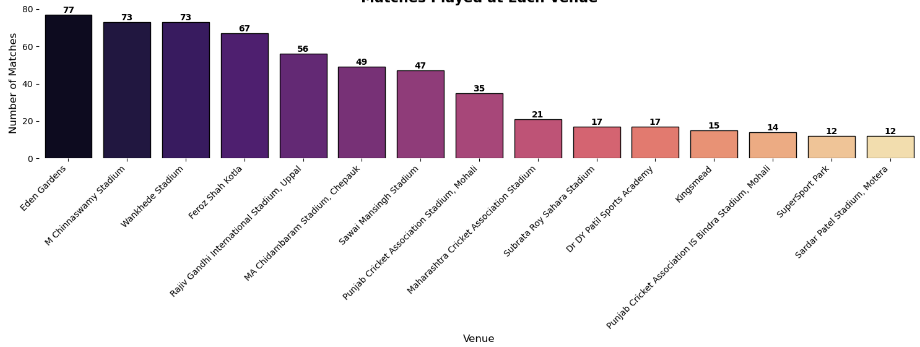
**Top 10 Players of the Match**



**Insights Gained:**

* Gayle wins the most! He's the best at getting "Player of the Match."
* Four other guys are right behind him – Dhoni, Warner, de Villiers, and RG Sharma all win a lot too.
* Everyone on this list is a top player who often helps their team win.

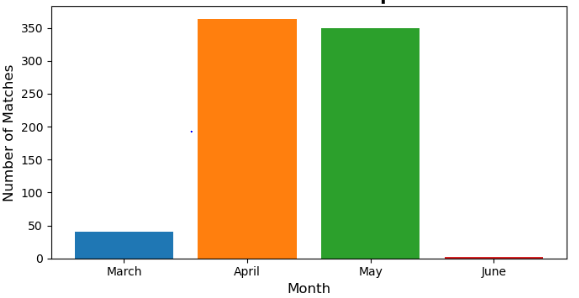
**Matches Played at Each Venue**



**Insights Gained:**

* M. Chinnaswamy Stadium and Eden Gardens are the busiest stadiums. They've hosted more games than anyone else.
* A few other stadiums like Arun Jaitley and Wankhede are also very popular.
* Some stadiums like Narendra Modi Stadium, SuperSport Park etc. have hosted far fewer matches.

**Number of IPL Matches per Month**



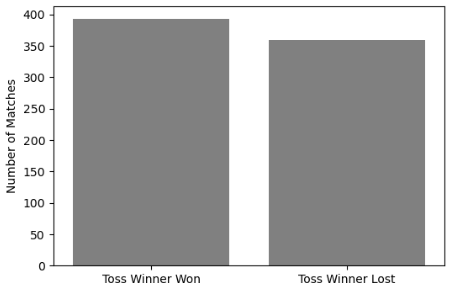
**Insights Gained:**

* April & May are Peak Months: Most matches happen then because it's the climax of sports seasons (playoffs/finals) and the weather is good.
* June is Slow: Activity drops as many leagues enter their off-season or summer break.
* March is Moderate: It's an earlier stage of the season, building up to the peak.
* Clear Seasonal Pattern: Match schedules follow seasons due to weather, league structure and holidays.

**BIVARIATE ANALYSIS**

Bivariateanalysis is the statistical study of the relationship between twovariables.

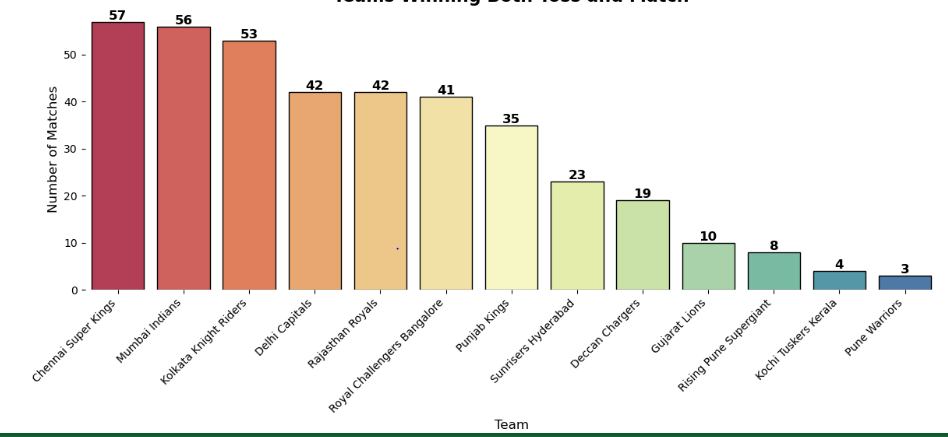
**Did Toss Winner Win the Match?**



**Insights Gained:**

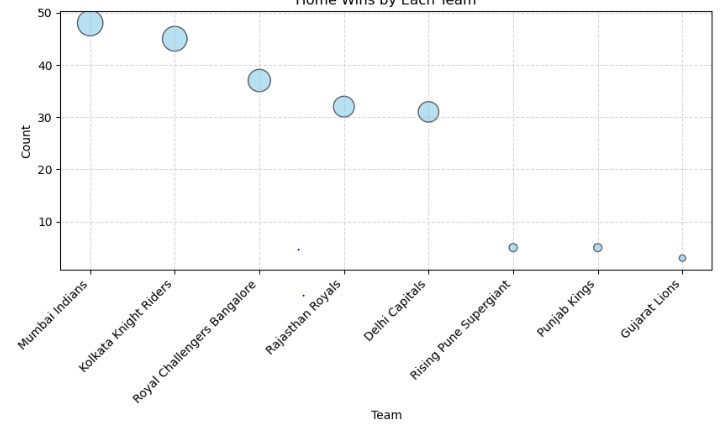
* Teams that win the coin toss usually win the game.
* But, winning the coin toss doesn't mean you'll always win; you can still lose a lot of games even if you win the toss.

**Teams Winning Both Toss and Match**



**Insights Gained:**

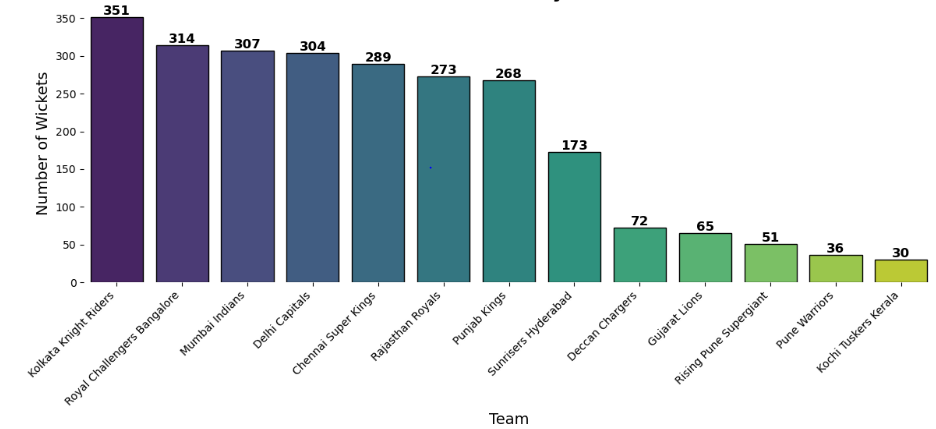
* Chennai & Mumbai dominate: These two teams win the most matches even when they win the toss.
* Most teams are pretty close: Many teams sit in the middle with similar toss-win match records.
* Some struggle even after winning the toss: A few teams, especially older, shorter-lived ones, didn't often convert the toss win into a match win.
* Winning the toss helps, but talent matters more: It's not just luck; strong teams make the most of winning the toss.

**Home Wins by Each Team** 

**Insights Gained:**

* Mumbai Indians boast the highest number of home wins near 50, showing their dominant home performance.
* Kolkata Knight Riders and Royal Challengers Bangalore follow with both teams securing over 30 home victories.
* Rajasthan Royals and Delhi Capitals also performed well, each with home wins exceeding 30.
* Rising Pune Supergiant, Punjab Kings, and Gujarat Lions recorded significantly fewer home wins, all under 10.
* The bubble size effectively visualizes the win count, making the relative home performance of each team immediately clear.

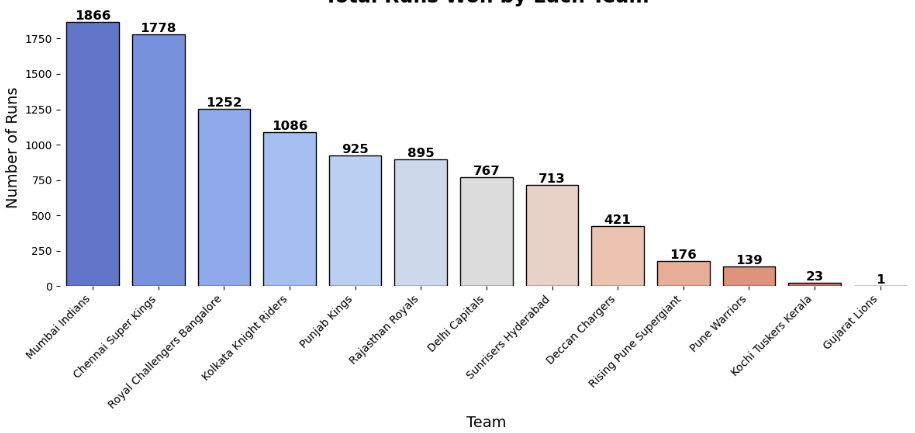
**Wickets Won by Each Team**



**Insights Gained:**

* Wicket Warriors: Kolkata Knight Riders stand tall as the ultimate wicket-takers, leading the pack!
* Top Contenders: Teams like Mumbai Indians and Chennai Super Kings also show strong wicket-taking performances.
* Tail-enders: Newer or less successful teams like Kochi Tuskers Kerala lag far behind in wicket counts.

**Total Runs Won by Each Team**

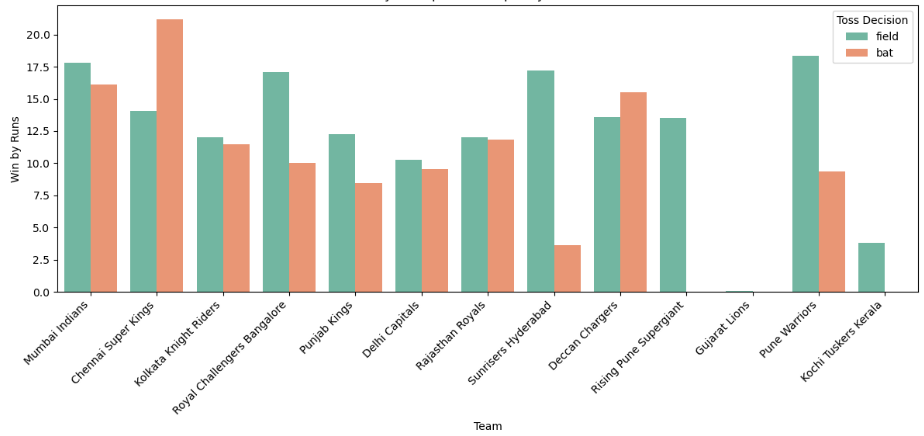


**Insights Gained:**

* Mumbai & Chennai: These two teams are the undeniable kings of run-winning.
* The Chasing Pack: RCB and KKR consistently lead the pursuit, but haven't quite matched the top two.
* The Rest of the League: A mix of steady performers and teams with shorter or less impactful tenures.

**MULTIVARAIATE ANALYSIS**

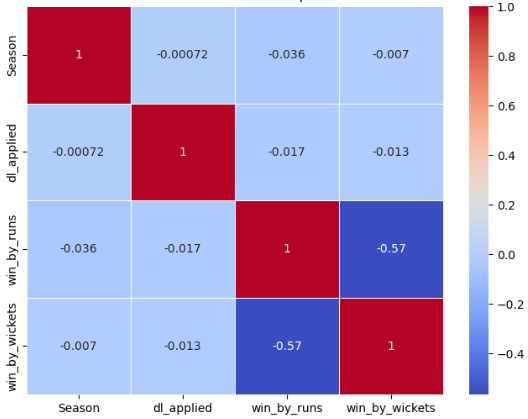
**Wins by Runs per Team Split by Toss Decision**



**Insights Gained:**

* Toss Twist: Batting or fielding first matters differently for each team!
* CSK Loves Batting First: Chennai Super Kings clearly thrives setting a target.
* MI & SRH Field First Favers: Mumbai Indians and Sunrisers Hyderabad shine when chasing.
* Not All Teams are Toss-Sensitive: For many, the toss decision has a less clear impact.

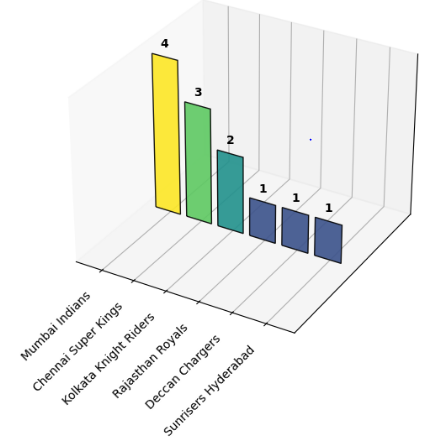
**Correlation Heatmap**



**Insights Gained:**

* The correlation heatmap shows that win\_by\_runs and win\_by\_wickets are moderately negatively correlated (-0.56), which aligns with cricket rules (a team can either win by runs or by wickets, but not both).
* The dl\_applied feature has negligible correlation with both, suggesting it independently affects match outcomes without directly influencing the margin of victory.

**IPL Trophies Won by Each Teams**



**Insights Gained:**

* Mumbai and Chennai: The IPL Kings! They've won the most trophies by far.
* It's a Two-Team Show: Mumbai Indians and Chennai Super Kings dominate, grabbing almost 60% of all titles shown.
* Kolkata's Strong Presence: Kolkata Knight Riders are a clear third in terms of wins.
* Other Champions: Rajasthan, Deccan, and Hyderabad each have one title under their belt in this chart.

1. **DATA PREPROCESSING**

Before training the model, the dataset was preprocessed as follows:

1. **Dropping Irrelevant Columns**Certain columns were not required for prediction (as they directly reveal the outcome or are not useful for the model).



1. **Encoding Categorical Variables**  
   Since machine learning models require numerical input, all categorical columns were encoded using LabelEncoder.



1. **Splitting Data into Training and Testing Sets**  
   To evaluate the model’s performance, the dataset was split into training (80%) and testing (20%) sets using train\_test\_split.

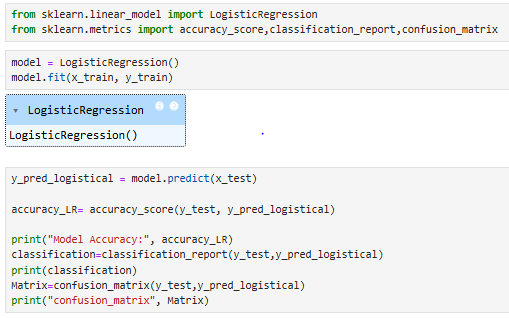


1. **MODEL TRAINING**

After preprocessing, machine learning models were trained to predict the match winner. Each model was trained using the training set and evaluated on the testing set.

**1.Logistic Regression (LR)**

Logistic Regression was used as a **baseline model** due to its simplicity and interpretability.



**2. Support Vector Classifier (SVC)**

SVC was chosen for its effectiveness in classification tasks, especially with non-linear decision boundaries.



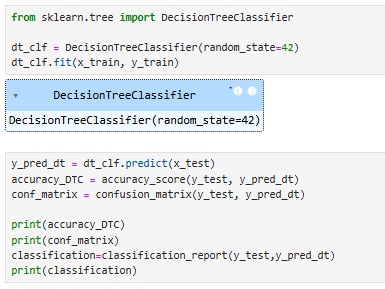
**3. Random Forest Classifier (RFC)**

Random Forest, an ensemble method, was used for better accuracy and handling of categorical variables.



**4. Decision Tree Classifier (DTC)**

Decision Tree was tested for interpretability and ease of visualization.



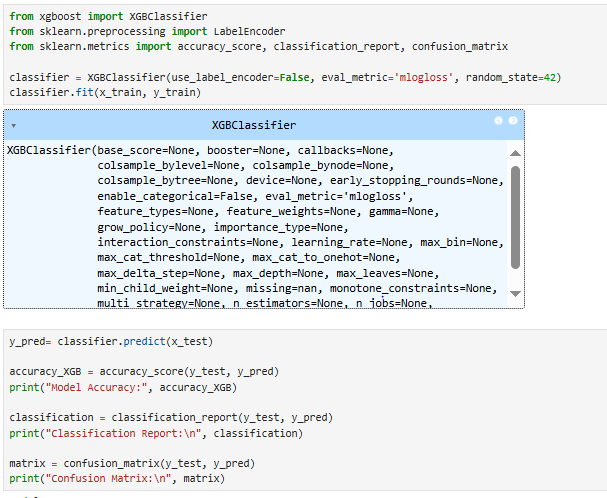
**5. K-Nearest Neighbors (KNN)**

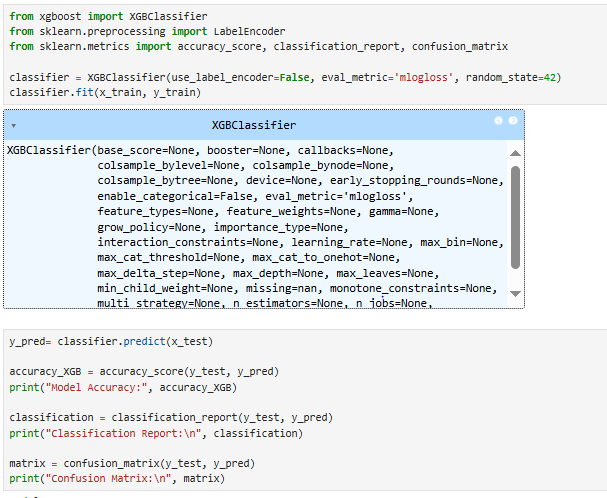
KNN was applied as a distance-based method to compare with other classifiers.



**6. Extreme Gradient Boosting (XGB)**

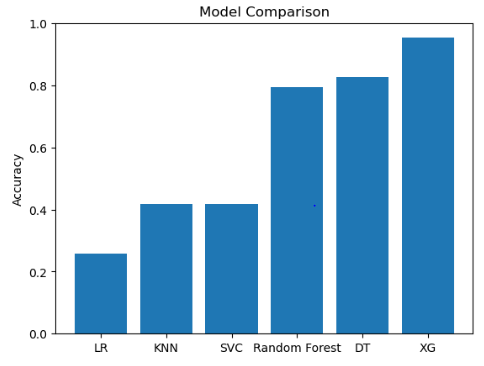
XGBoost was included as a powerful boosting algorithm commonly used in Kaggle competitions and sports analytics.





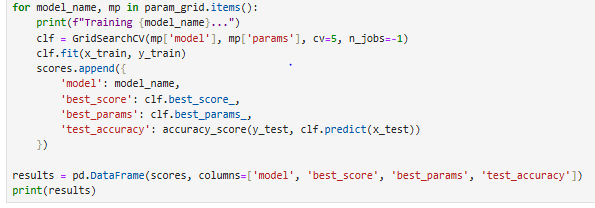
**9.MODEL COMPARISON**

|  |  |
| --- | --- |
| **Model** | **Accuracy** |
| Logistic Regression | 0.25 |
| SVC | 0.41 |
| Random Forest | 0.79 |
| Decision Tree | 0.82 |
| KNN | 0.41 |
| XGBoost | 0.95 |



**10.HYPERPARAMETER TUNING**

To improve model performance beyond the default parameters, GridSearchCV was used to perform hyperparameter tuning. Each model was optimized with respect to important hyperparameters, and the results were compared with baseline models.



|  |  |
| --- | --- |
| **Model** | **Accuracy** |
| Logistic Regression | 0.27 |
| SVC | 0.54 |
| Random Forest | 0.80 |
| Decision Tree | 0.95 |
| KNN | 0.54 |
| XGBoost | 0.98 |

 Hyperparameter tuning improved model performance compared to default values.

 **XGBoost** achieved the highest accuracy after tuning, followed by **Decision Tree**.

 Simpler models like Logistic Regression and KNN showed lower improvements.

**11.CONCLUSION**

The objective of this project was to develop predictive models for forecasting match winners. From the analysis, it was observed that matches are typically played between March and June, with most wins achieved by chasing teams. The Duckworth–Lewis method was applied in only a small percentage of games.

Six machine learning models were trained and evaluated: Logistic Regression, SVC, Random Forest, Decision Tree, KNN, and XGBoost. After hyperparameter tuning, **XGBoost** achieved the highest accuracy (98%), followed closely by Decision Tree, while simpler models like Logistic Regression and KNN performed comparatively lower.

These results demonstrate that ensemble models, particularly XGBoost and Decision Tree, are well-suited for sports prediction tasks. In the future, incorporating **player performance metrics, toss impact analysis, and weather conditions** could further enhance predictive accuracy, practical applications in cricket analytics and match forecasting."