IPL Data Analysis
Venkateswara Rao Addala

Title: IPL Data Analysis and Predicting Match Outcomes

Author: Venkateswara rao Addala

Contents:

Purpose:

Cricket isn't just a sport in India it's a religion with the IPL sitting at its heart, it has the largest cricket fan base in the world, and IPL franchises, sponsors, and enthusiasts constantly seek data-driven insights. By analyzing historical match and player stats, this project aims to:

- Help team management and auction strategists bid smartly for players (batsmen, bowlers) each season.
- Guide sponsors to identify and support high-impact players and teams based on performance metrics.
- Enable bettors and analysts to forecast match outcomes with greater confidence.

1 Introduction

In this project, I have attempted to perform Data Analysis on available data on the **Indian Premier League**, a yearly cricket tournament that started way back in year 2008. I have performed Exploratory data analysis, followed by employment of ml models to predict the winner of the match based on some pre-determined factors.

1.1 A Brief History

The Indian Premier League (IPL) is a professional men’s Twenty20 cricket league, contested by ten (or less) teams based out of Indian cities. The league was started by the Board of Control for Cricket in India (BCCI) in 2007. It is usually held between March and May of every year and has an exclusive window in the ICC Future Tours Programme.

2 Data Information

2.1 Source

Data sourced from the IPL Complete Dataset (2008–2024) on Kaggle.

- Matches.csv
- Deliveries.csv

2.2 Filters Applied

- Focused only on completed matches, merged ball-by-ball data with match details, and cleaned out missing or invalid entries.

- Identified each season’s champion by picking the final match and tallying team wins.

- Counted matches per venue and calculated team match-win and toss-win rates.

- Tracked batting milestones (50s/100s), built head-to-head win matrices, and calculated strike rates for batters (min 100 balls).

- Isolated wicket-taking deliveries to count bowler wickets and compute economy rates, seasonal top performers by run totals and strike rates.

- Analyzed fielding by counting catches and run-outs, and captured wicket-keeper stumpings and catches.

- Examined venue scoring patterns and split winning margins into batting-first vs. batting-second contexts.

- Engineered features like historical and venue-specific win rates, head-to-head records, home-ground advantage, bowling strength and batting form.

3. Method

I started by gathering IPL match dataset(2008-24) from Kaggle and ball-by-ball data, then filtered out super-overs and incomplete games to focus only on clear results. For each match, I created a single record packed with practical features like who won the toss and what they chose, each team's runs, run rates, bowling economy figures, historical win percentages like overall, venue-specific, and head-to-head matches, simple indicators of player form and whether teams were playing at home or not.

And then, I split my dataset into training and test portions, used a pipeline to standardize the numerical values and encoded the toss information, then trained four different classifiers – Logistic Regression, Random Forest, Gradient Boosting, and XGBoost. To evaluate performance, I measured cross-validation and test-set accuracy, examined which features had the most predictive power, and compared confusion matrices and ROC curves to determine which model performed best.

4. Statement

I have built and evaluated machine-learning models to predict the winner of an IPL cricket match using only pre-game information. By combining team previous performances, venue biases, toss decisions, and simple proxies for player form etc, my goal was to see how accurately I could forecast the outcome of a match.

5.Proof

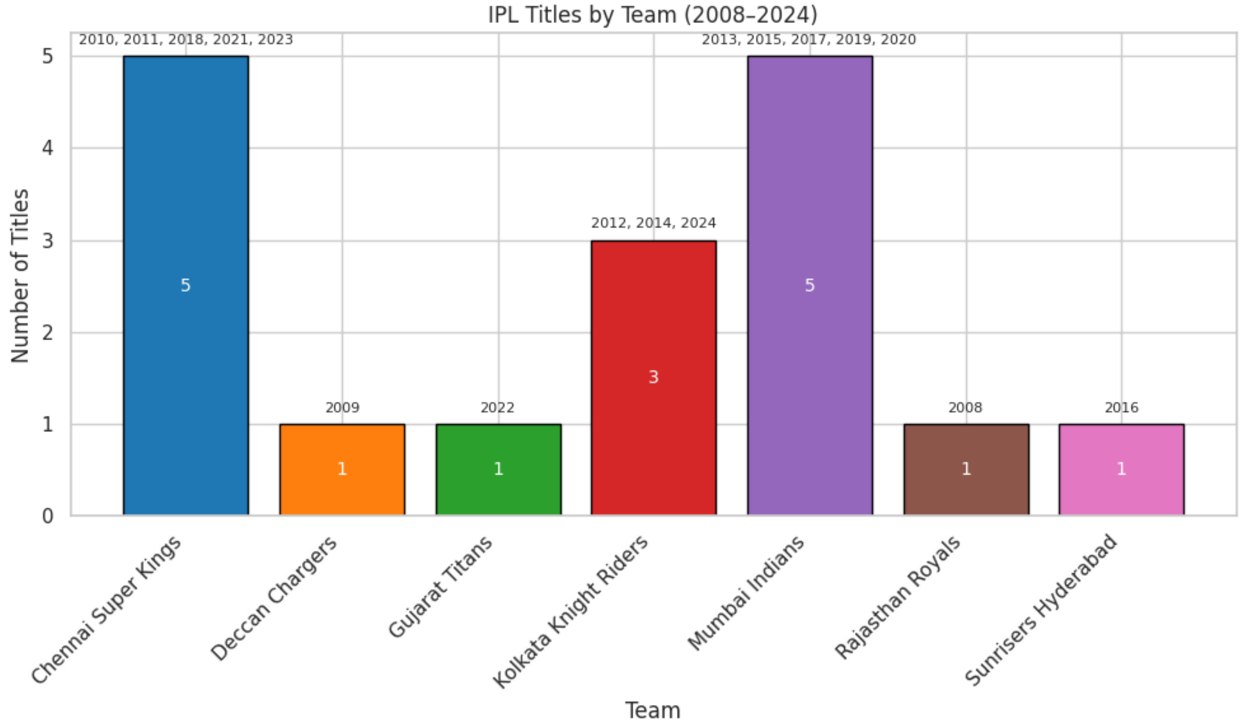
Model	**Test_Acc**
XGBoost	0.745
GradientBoosting	0.694
RandomForest	0.597
LogisticRegression	0.449

6. Exploratory Data Analysis

Exploratory Data Analysis (EDA) is an approach in analyzing data sets to summarize their main characteristics, often using statistical graphics and other data visualization methods. EDA assists Data science professionals in various ways:-

1. Getting a better understanding of data
2. Identifying various data patterns
3. Getting a better understanding of the problem statement

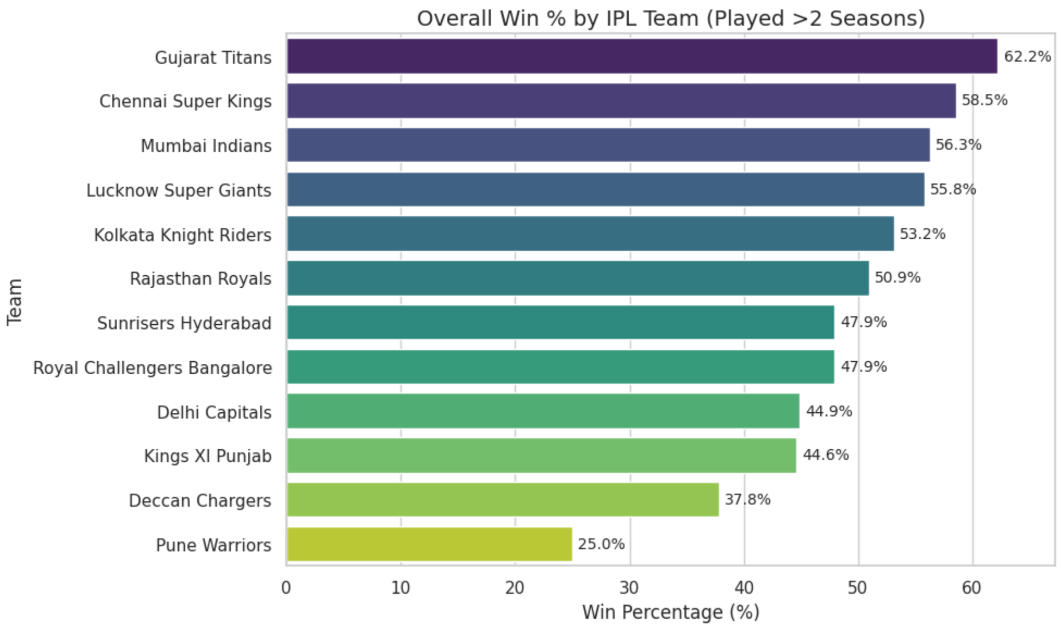
6.1 IPL Title Champions



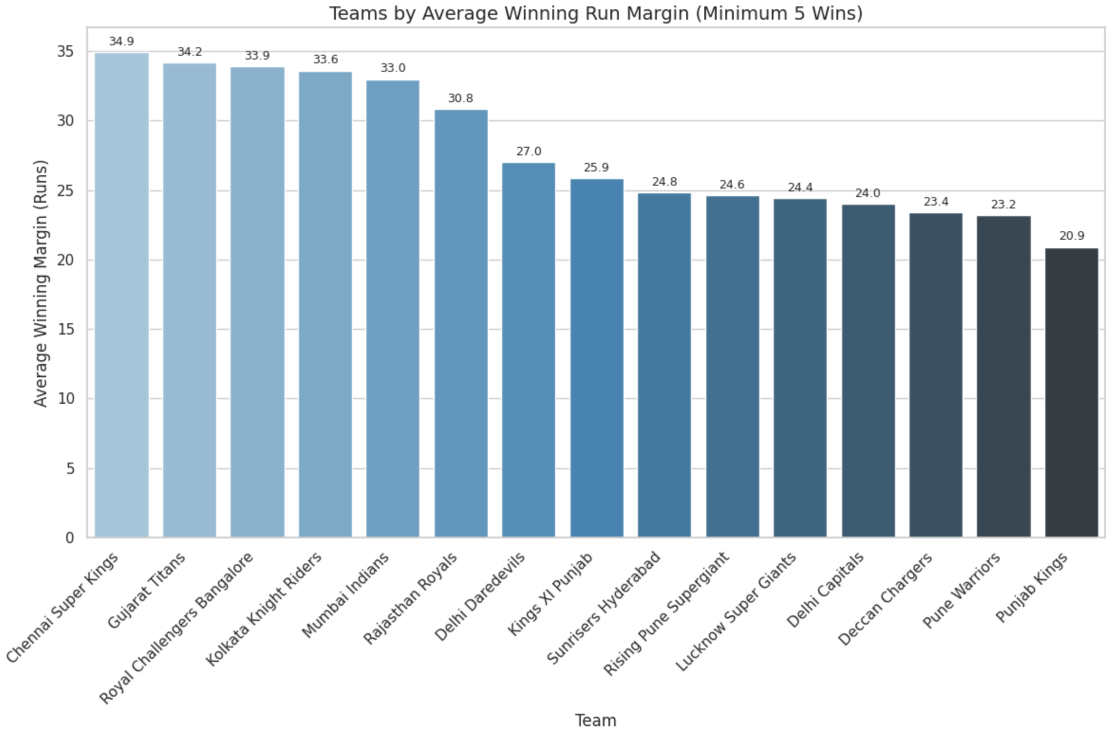
This bar chart displays each franchise’s title count from 2008 to 2024, showing Chennai Super Kings and Mumbai Indians leading with five titles each and Kolkata with 3 titles and the remaining teams with one title each

6.2 Win Percentage by Team

This Plot shows each franchise’s overall match-winning percentage, with top performers like Gujarat Titans above 60% and also highlights teams with lower consistency, such as Pune Warriors at 25%, indicating varied historical success.

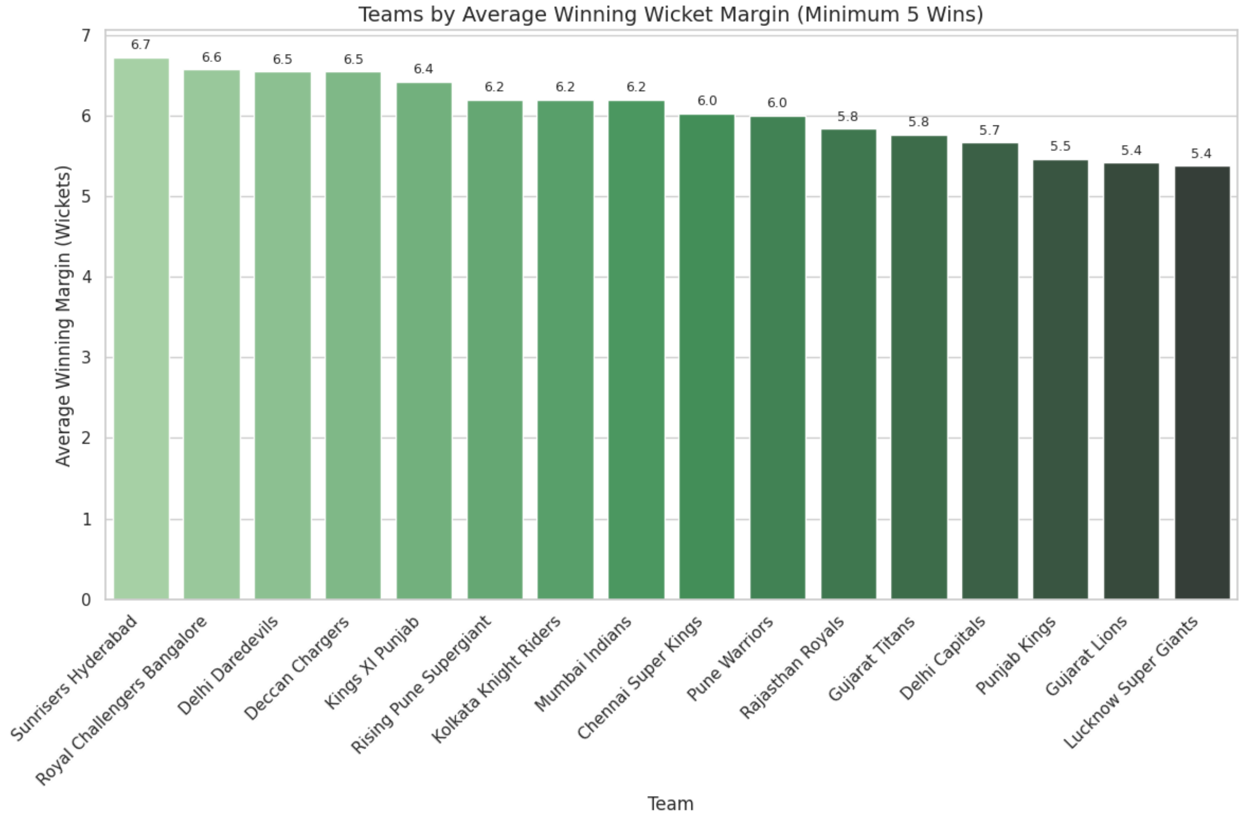


6.3 Teams Average Winning Margin by Runs



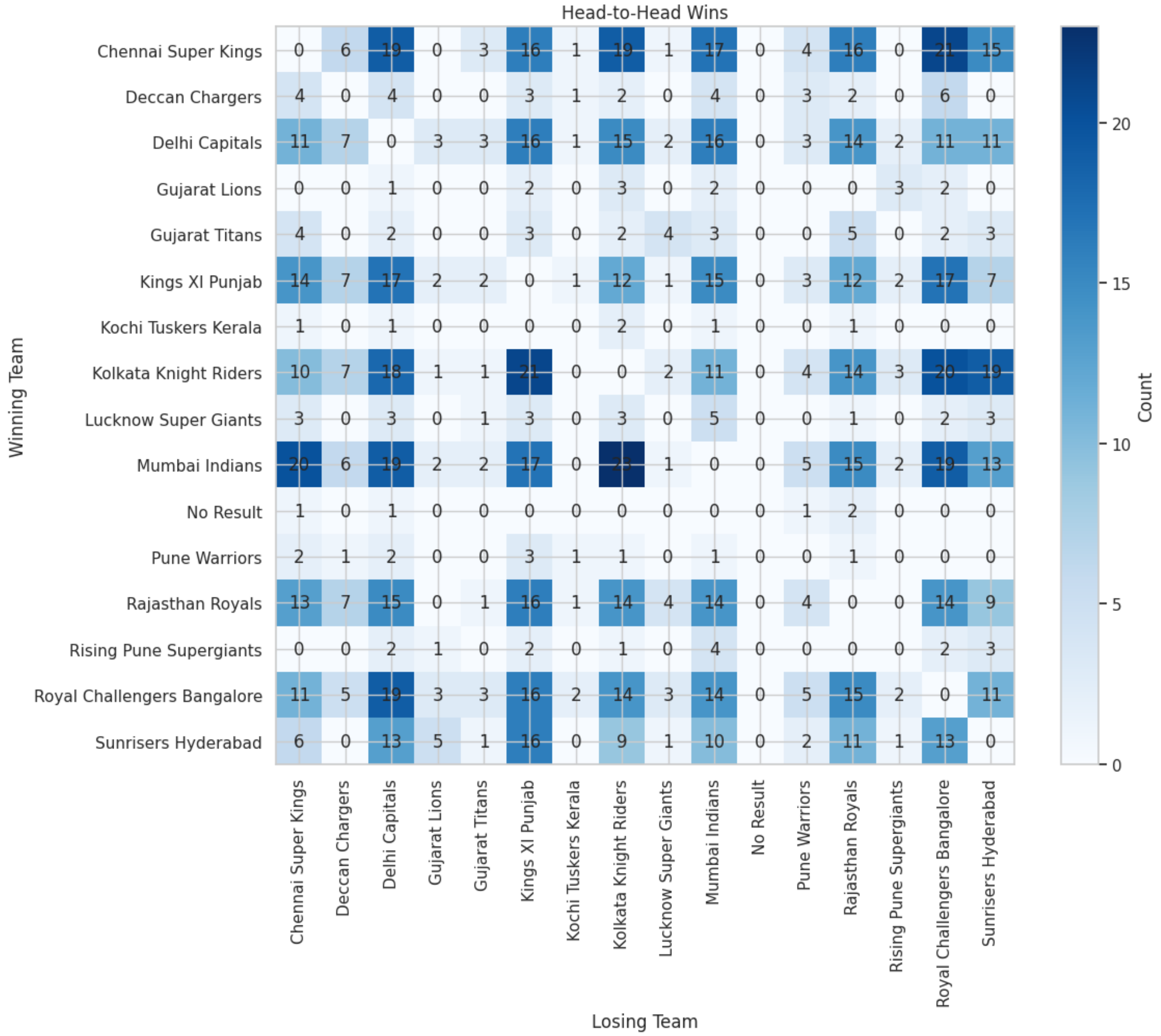
The above plot shows each team’s average run margin in victories, with Chennai Super Kings leading at about 36 runs and also highlights tighter wins by teams like Pune Warriors, who average just over 20 runs.

6.4 Teams Average Winning Margin by Wickets



It displays each team's average wicket margin in wins, with Sunrisers Hyderabad topping around 6 wickets and also shows teams like Lucknow Super Giants with narrower margins around 5 wickets.

6.5 Head-to-Head Win and Loss Heat Map

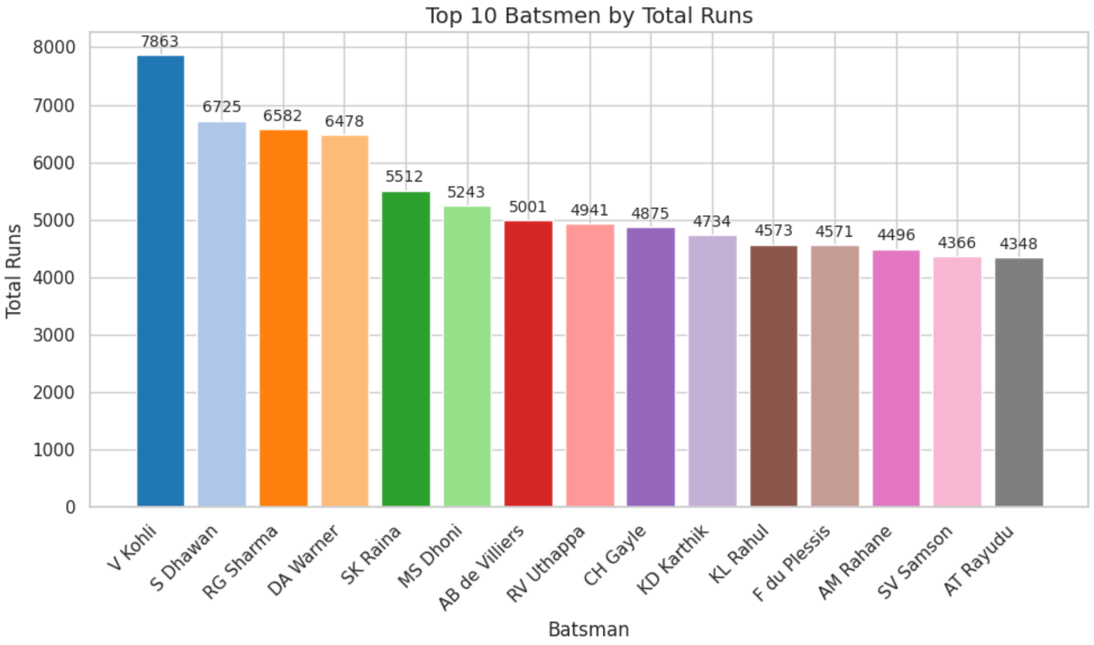


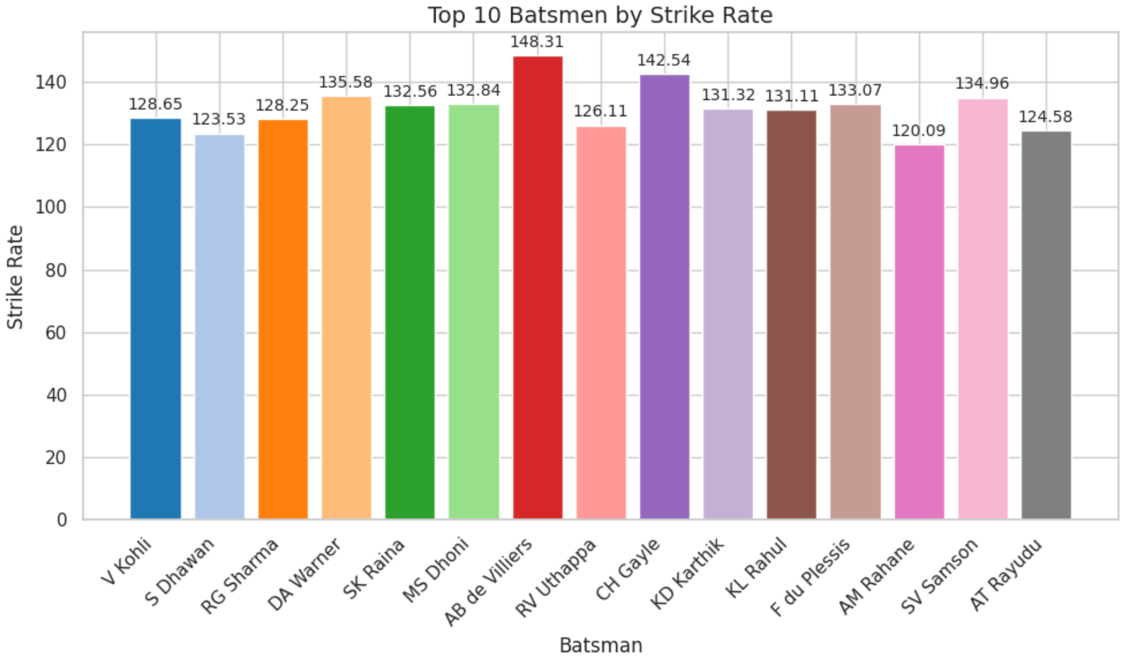
It displays win counts between every pair of teams, showing which teams consistently outperform others, the darker shades highlight the most frequent rival victories and the vice-versa for lighter shades.

For example, the cell where Mumbai Indians (row) meets Royal Challengers Bangalore (column) shows 19, indicating MI recorded 19 wins over RCB, while the cell at RCB (row) vs. MI (column) shows 14 losses for MI against RCB.

6.6 Top 10 Batsmen by Runs and Strike-Rate – Overall

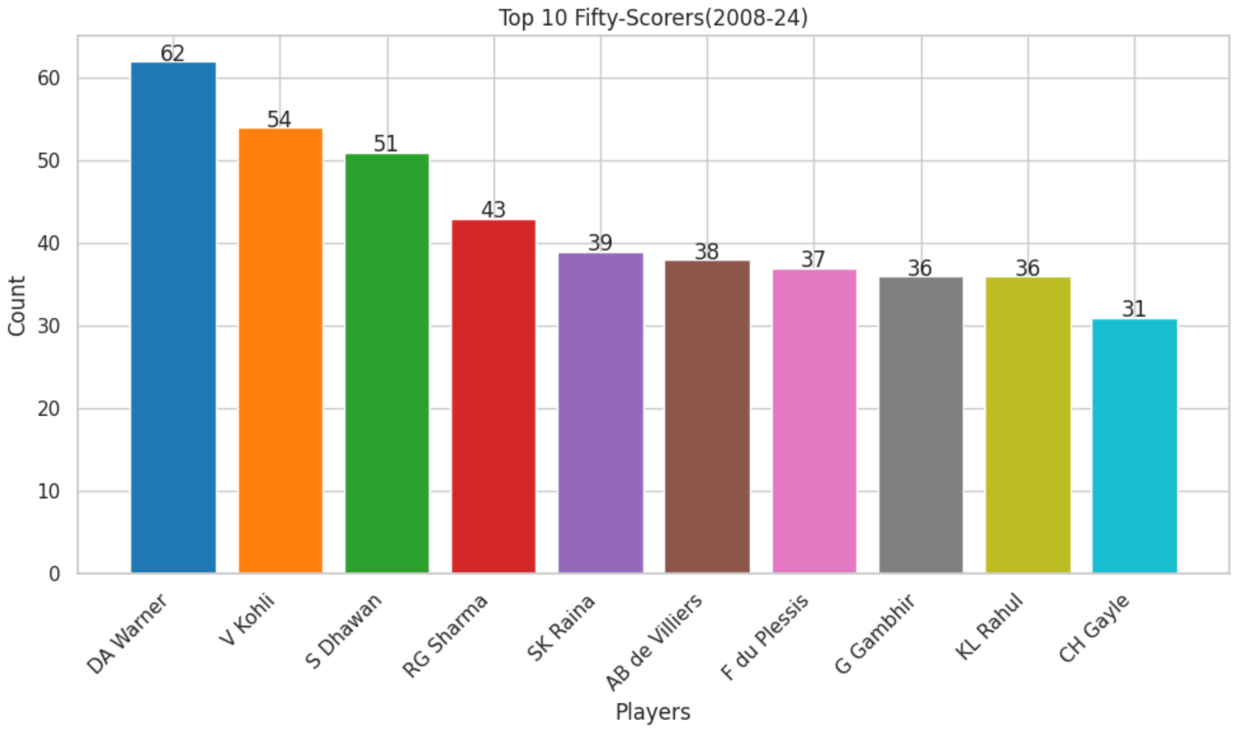
It displays the leading run-scorers from 2008–2024, with Virat Kohli topping the list at 7,863 runs and also highlights the consistency of players like Shikhar Dhawan and Rohit Sharma throughout IPL history.

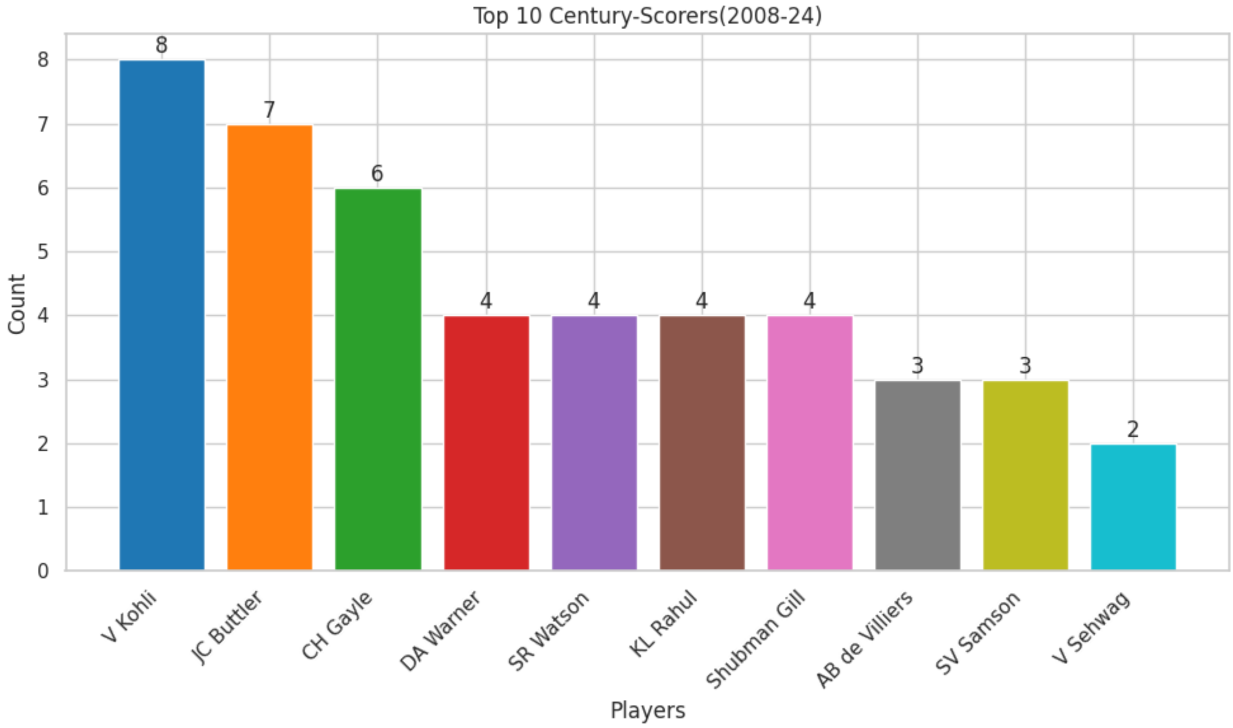




Here, **Strike rate** measures runs scored per balls faced **(i.e., runs scored ÷ balls faced × 100),** indicating scoring pace. The plot shows players with the quickest scoring rates (min 100 balls), with AB de Villiers leading at 148.3

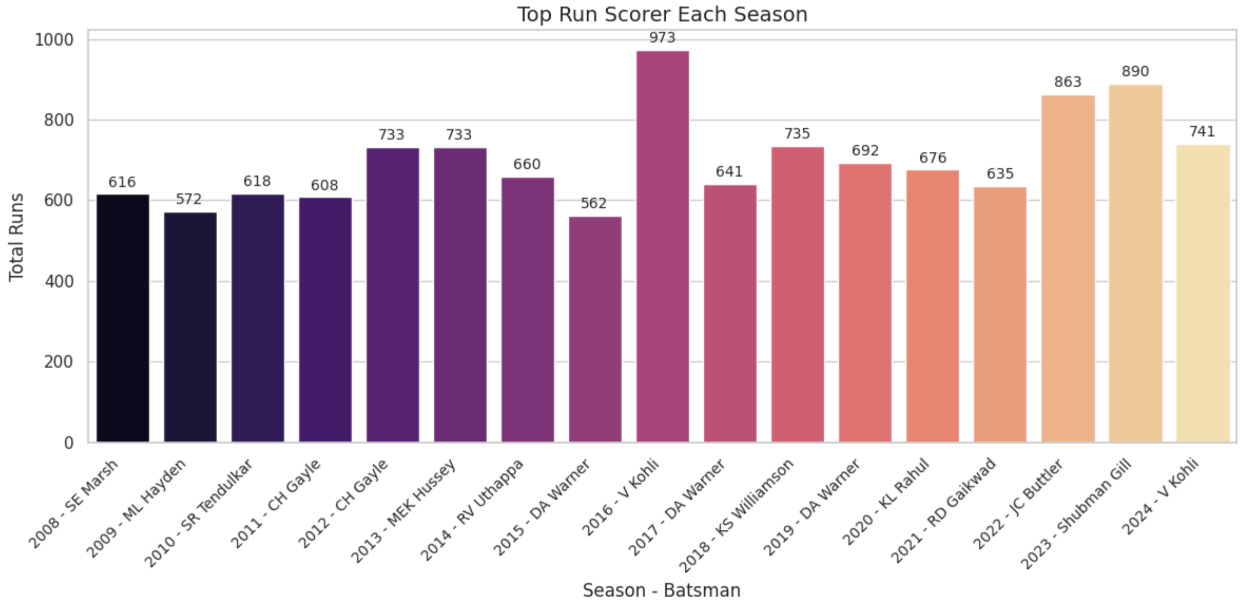
6.7 Top 10 Fifty and Century Scorers – Overall





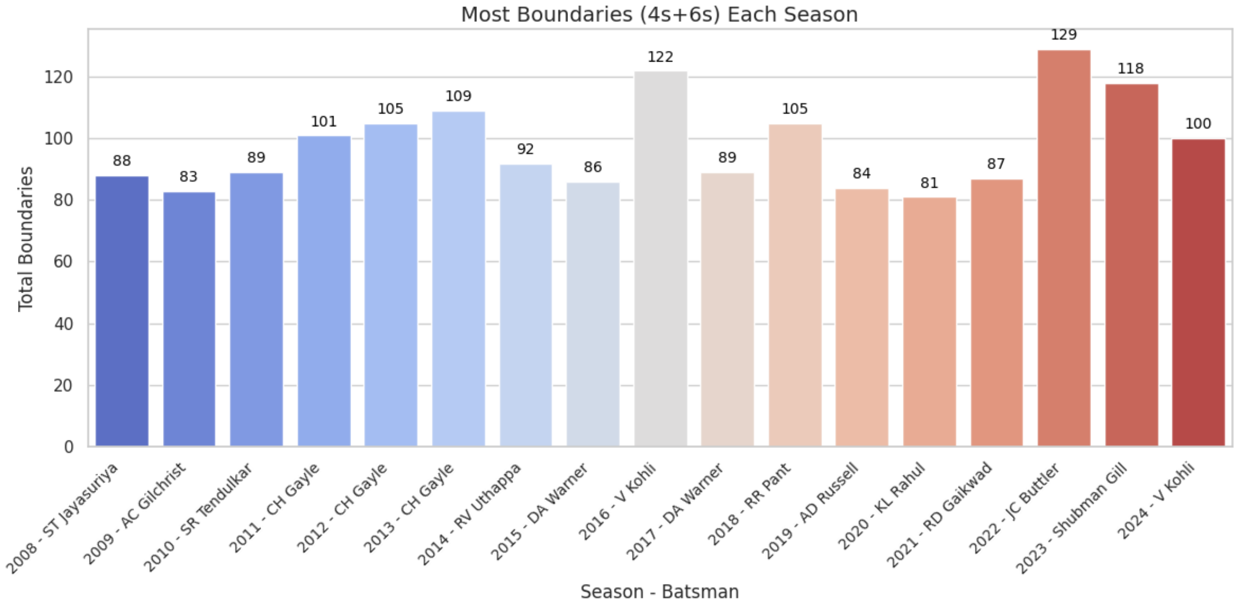
David Warner has the most 50+ scores (62), while Virat Kohli leads centuries with 8.

6.8 Top Run Scorer by Season wise (2008-24) – Orange Cap Holder’s



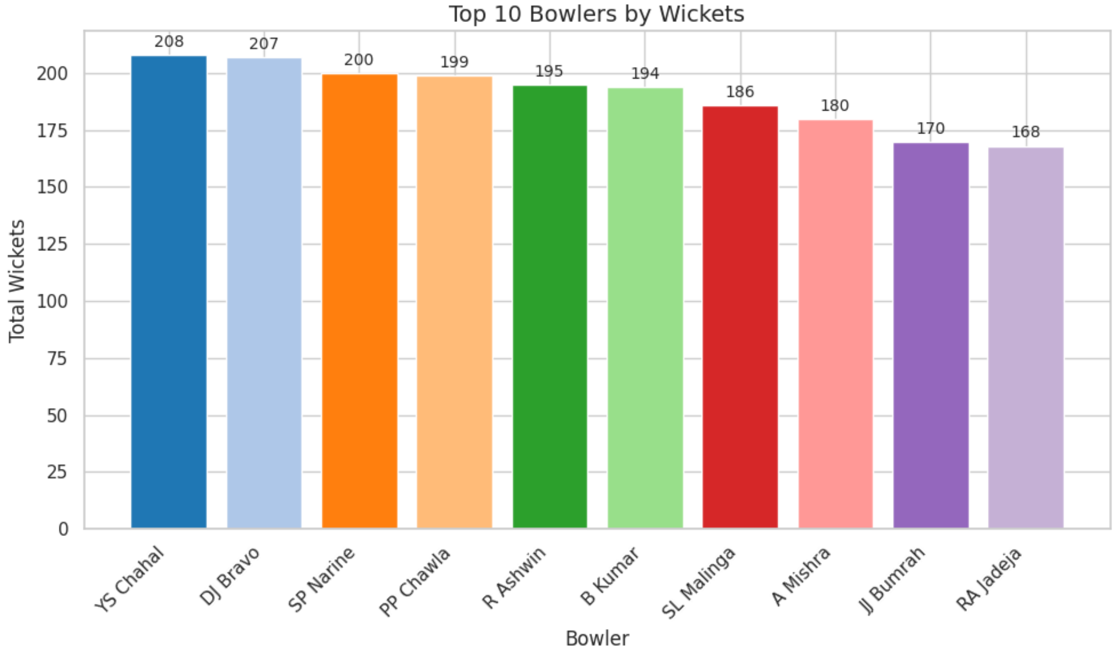
Warner is the top run scorer for three seasons followed by Kohli and Gayle who won two times

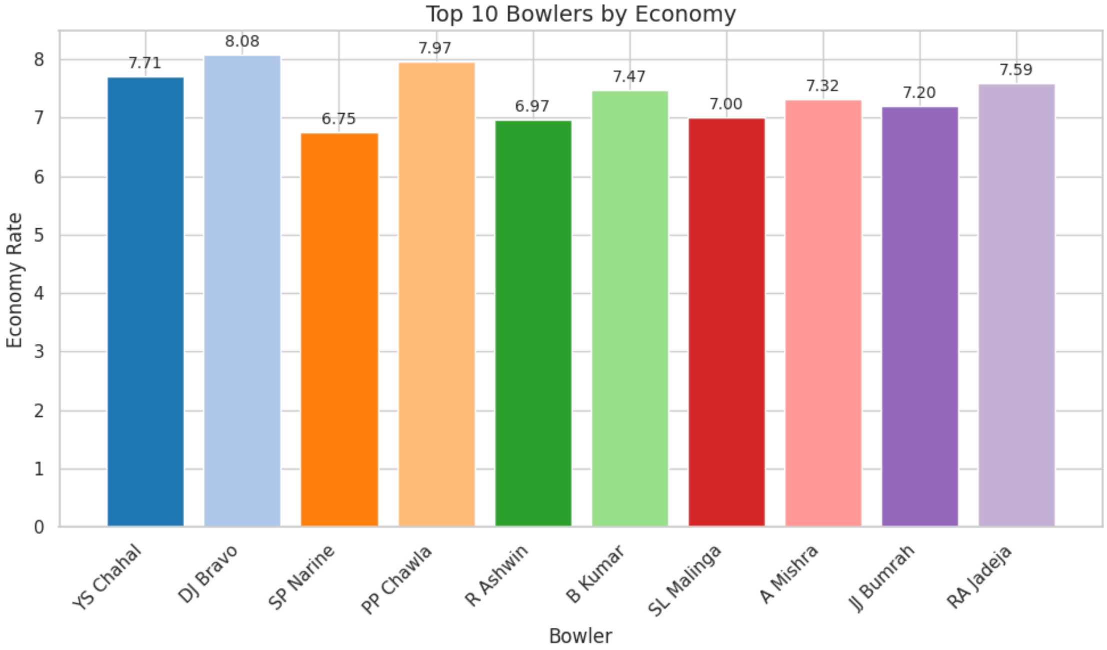
6.9 Batsmen with Most Boundaries by Season wise



It shows the highest total boundaries hit by a player each season, with Jos Butler peaking at 129 in 2013, this plot reflects boundary-hitting consistency, spotlighting aggressive season-long performance.

6.10 Top 10 Bowlers by wickets and Economy – Overall

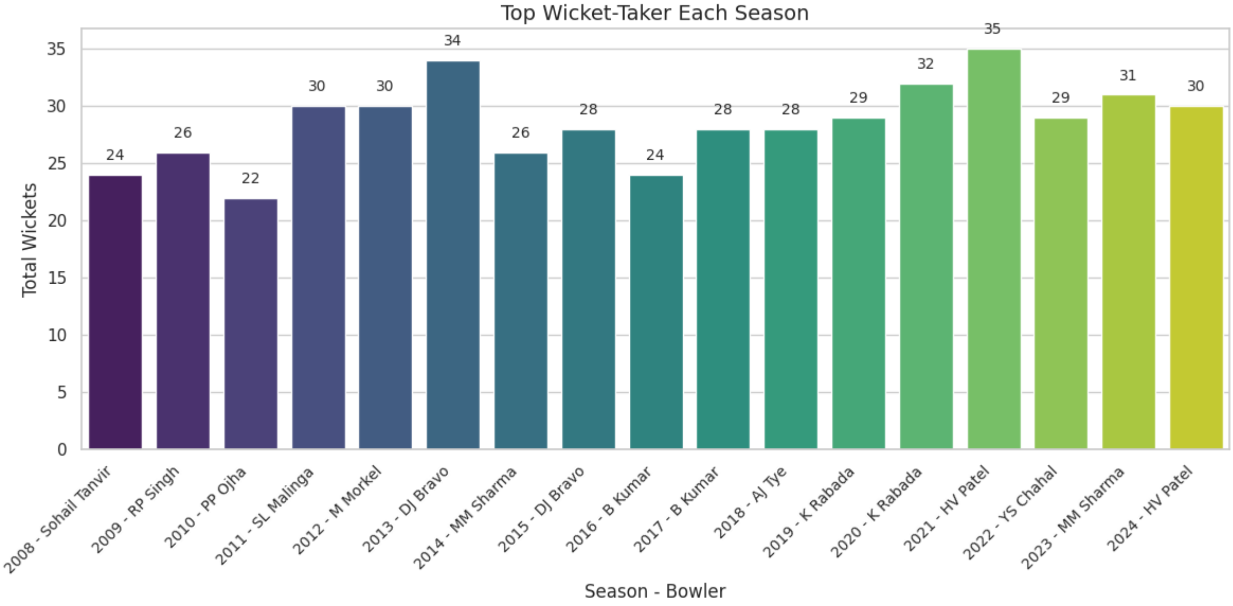




Economy rate measures the average runs a bowler concedes per over (total runs conceded ÷ overs bowled), reflecting their ability to contain scoring.

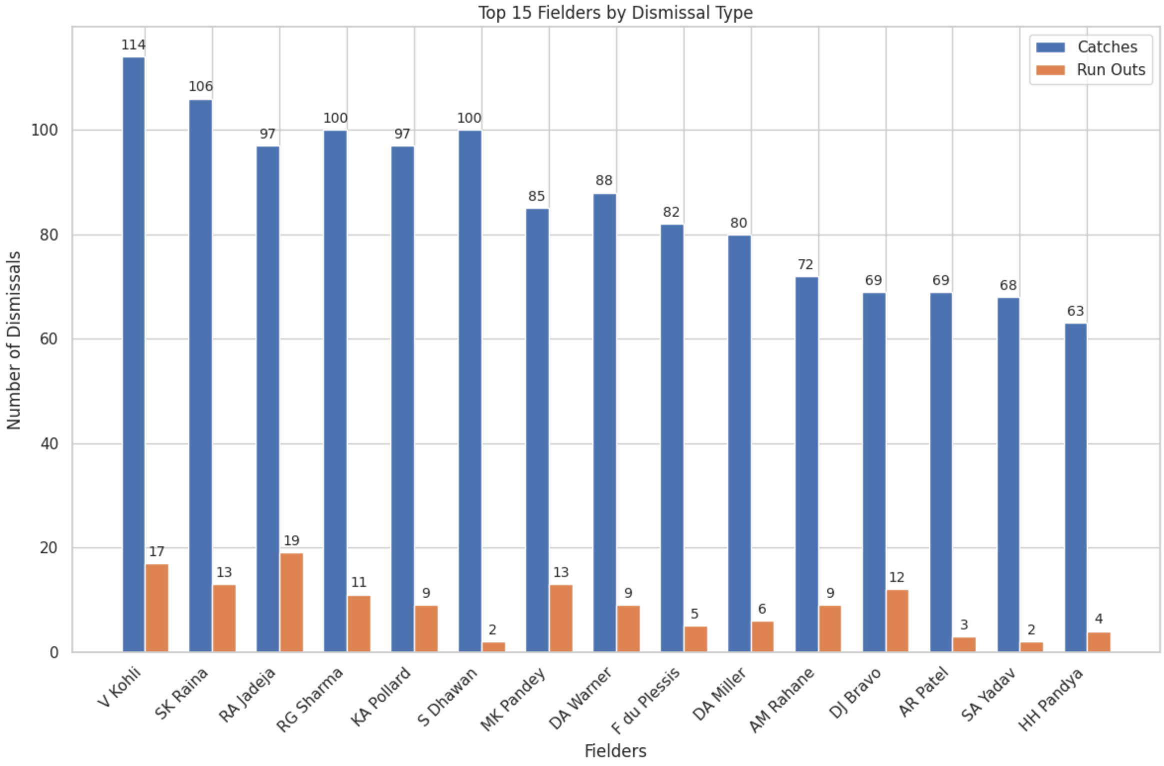
Here, Chahal tops the wicket chart with 208 dismissals, and Narine stands the best with a 6.75 bowling economy rate

6.11 Top Wicket taker by Season Wise (2008-24) – Purple Cap Holder’s

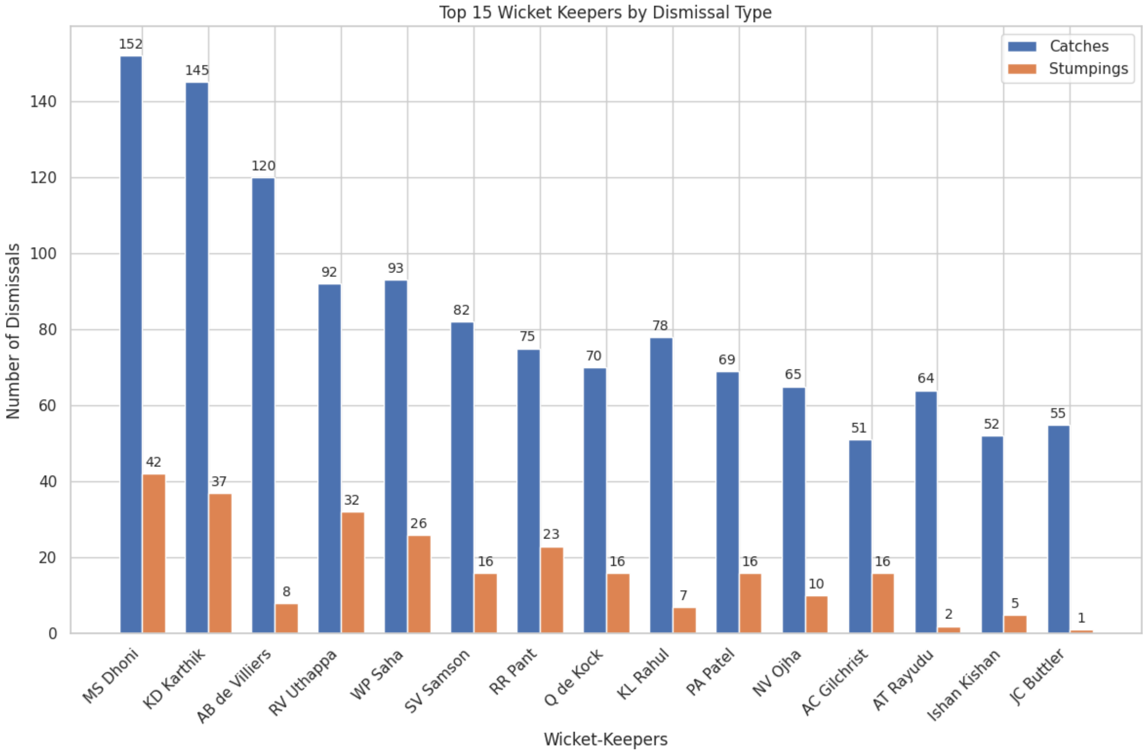


Bravo, Kumar, Rabada and Patel are the only two-time purple cap holders and the rest have won only once

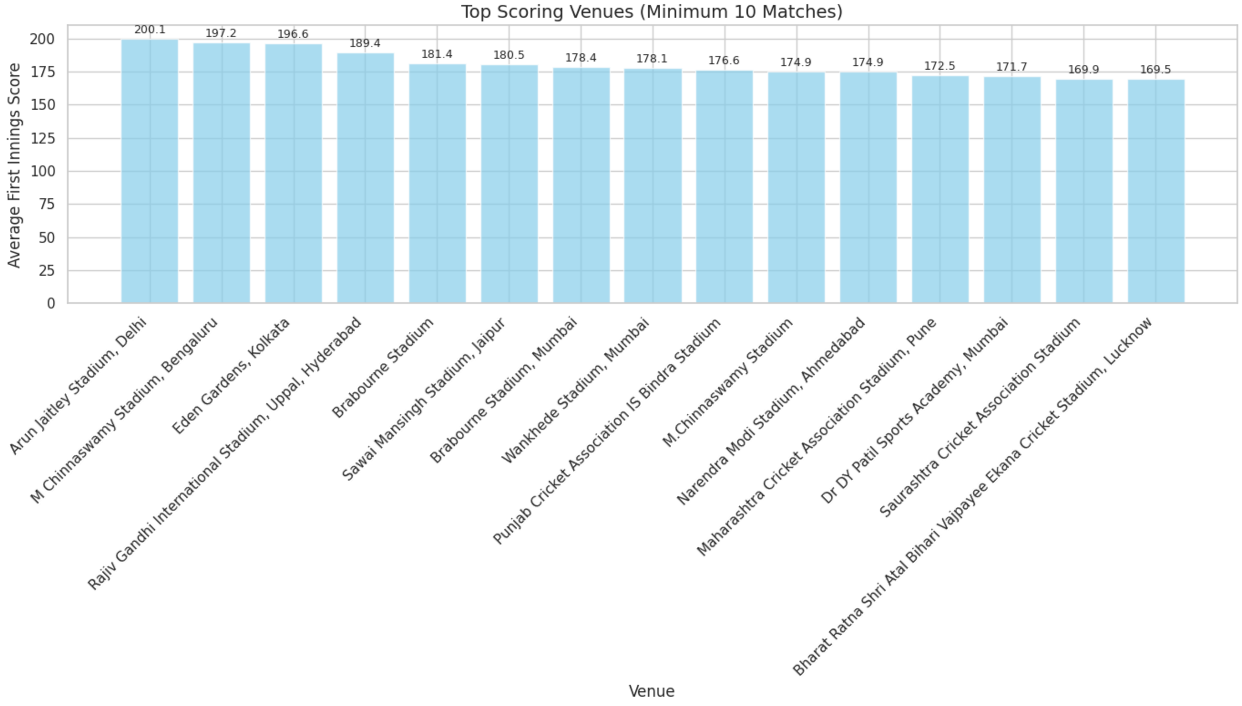
6.12 Top 15 Fielders - Overall



6.13 Top 15 Wicket Keepers – Overall

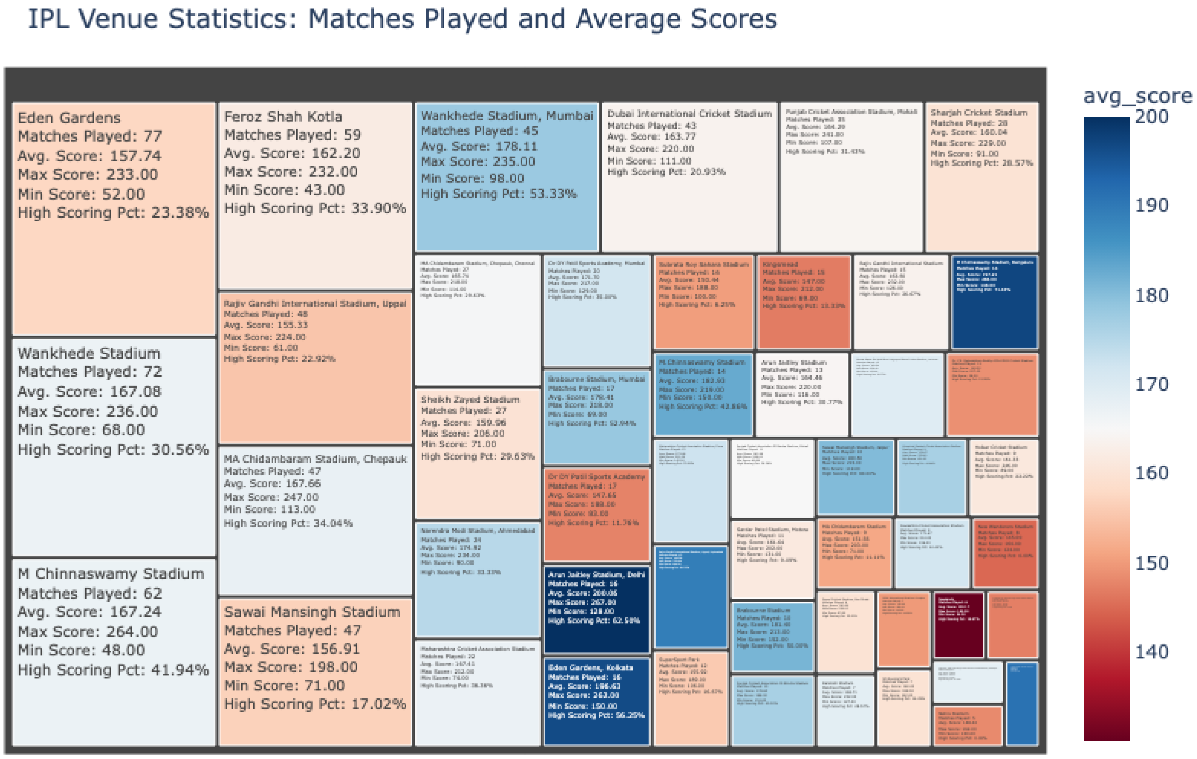


6.14 Top Scoring Venues



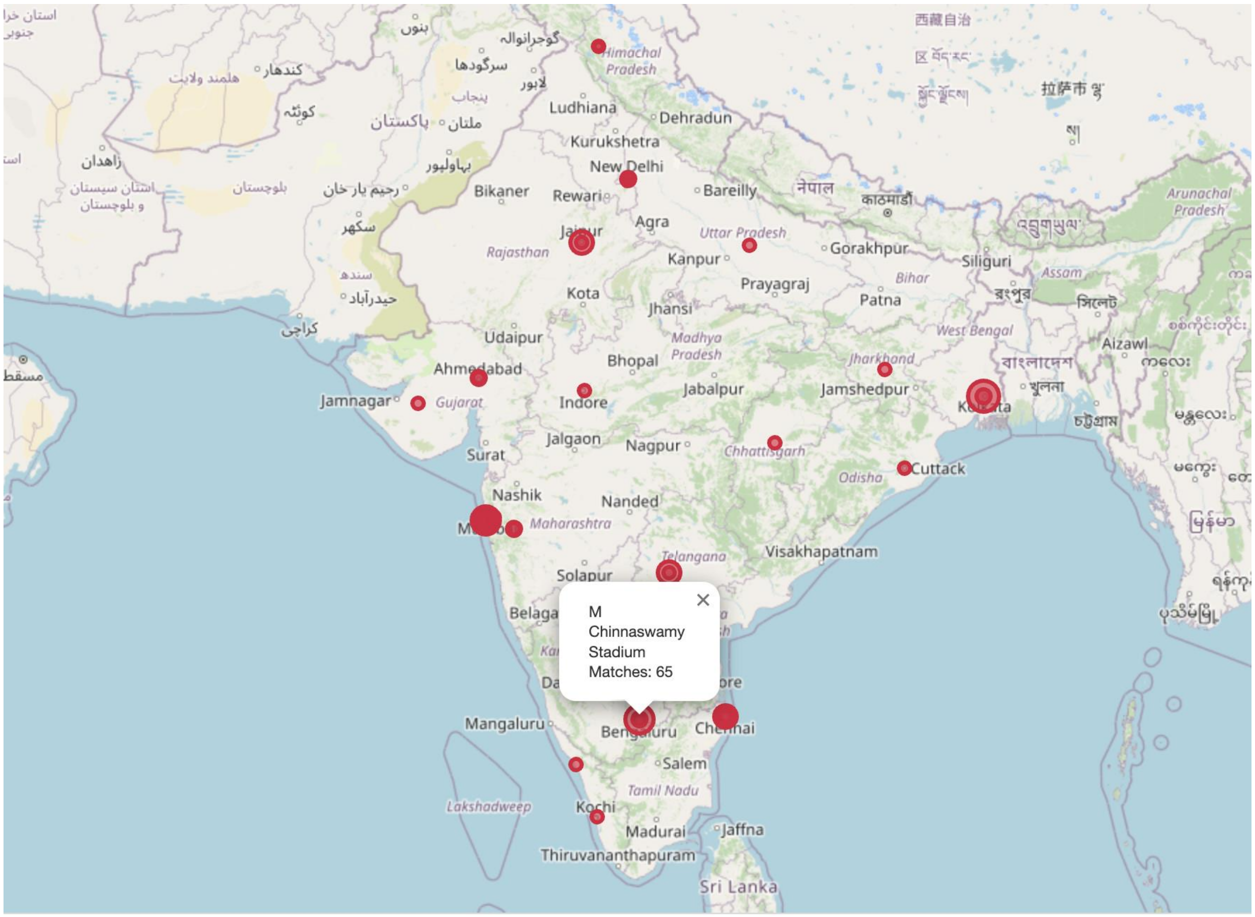
Bar chart of the top venues hosting at least 10 matches, highlighting Eden Gardens and Wankhede Stadium as most frequent grounds.

6.15 Venue Statistics



This treemap outlines each major venue by both total matches played count, average, Max, Min Score etc., easily highlighting grounds like Eden Gardens (high matches & scores). Here lighter, larger blocks indicate stadiums hosting many games with high run rates, while smaller, darker blocks show less-used or lower-scoring venues.

6.16 Matches Hosted by Venue



This map displays all IPL venues across India with circle markers sized by total matches hosted, offering a geographic view of match distribution and location.

Here, Larger, darker circles indicate the most-used grounds like Mumbai, Bangalore, and Kolkata.

7 Machine Learning Model Evaluation

The Prediction Model was build to predict the winner of an IPL cricket match using historical data and machine learning models

7.1 Data Preparation and Feature Engineering

I began by gathering ball-by-ball and match metadata, then carefully cleaned the timeline:

- **Filtering**: I dropped all super-overs and "no result" games so my model focused only on full, completed IPL encounters.
- **Computing Stats**: For each inning, I summed up batsman runs and delivery counts to calculate total **runs**, **balls**, **run rate**, and even each side's **economy rate**.
- **Creating the processed dataset**: I merged all these statistics back with match detail teams, toss winner, toss decision, venue, city, and player-of-the-match—to get one unified table per match.

From there, I created a mix of categorical and numerical predictors variables:

- **Categorical**
  - toss_winner and toss_decision capture the pre-game strategic choice.
- **Numerical**
  - Batting/bowling: team1_runs, team2_run_rate, team1_econ, etc.
  - Historical form: team1_win_pct, team2_venue_win_pct, team1_vs_team2_pct (head-to-head).
  - Proxies: team1_top_player flags if a star "Player of the Match" is on the side.
  - Contextual: team1_home marks home-ground advantage.
  - Depth metrics: team1_bowler_str averages the top three bowlers' economies; team2_bat_form averages the top five run-scorers' tallies.

7.2 Model Training and Evaluation

I used four different machine learning models to predict the match winner:

1. **Logistic Regression:** A straightforward model that estimates the probability of a binary outcome by fitting a weighted combination of inputs and squashing it through a logistic curve. It's fast, interpretable, but best suited when relationships are roughly linear.

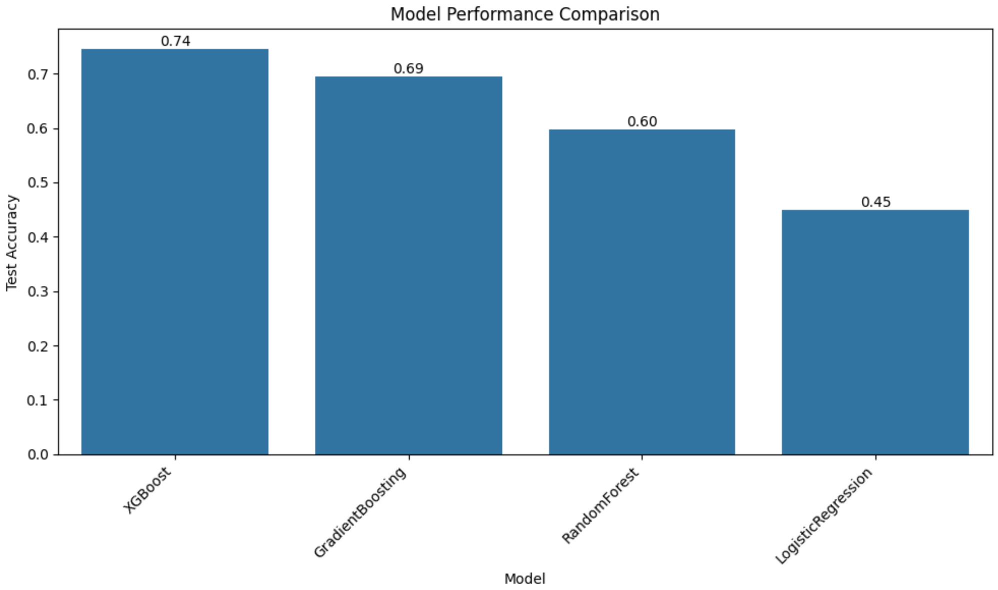
2. **Random Forest:** A combination of many decision trees trained on random subsets of data and features with their votes averaged to reduce overfitting and improve accuracy. It captures non-linear interactions without heavy tuning.
3. **Gradient Boosting:** Builds trees one after another, each new tree correcting its predecessor's errors, to gradually boost overall model performance. This sequential "learning from mistakes" yields strong predictive power but can overfit without careful regularization.
4. **XGBoost:** An optimized, scalable implementation of gradient boosting that adds regularization and clever approximations for speed and stability. It often wins data-science competitions thanks to its balance of accuracy, efficiency, and robustness.

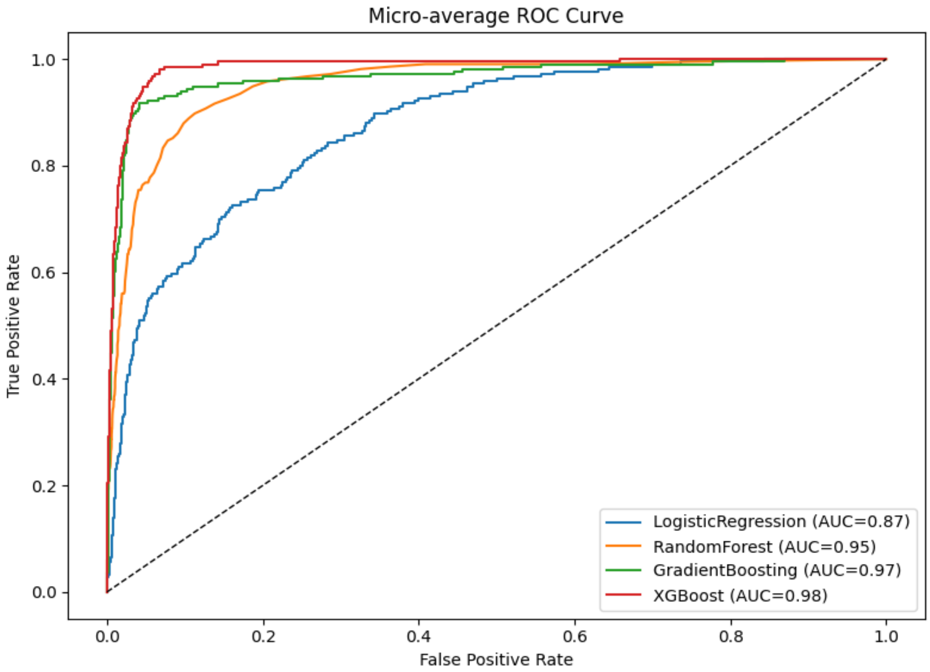
7.3 ROC Curve & AUC:

I have also examined each model's ROC curve, which plots true positive rate against false positive rate across all classification thresholds. The Area Under the Curve (AUC) summarizes discrimination ability:

- XGBoost produced the highest AUC, its curve bows closest to the top-left corner, confirming its superior capacity to distinguish winning from losing outcomes.
- Other tree-based methods (Gradient Boosting, Random Forest) follow, while Logistic Regression shows the weakest separation.

7.4 Results:





8. Conclusion

In my analysis, I successfully predicted IPL match winners with XGBoost achieving 74.5% accuracy on test data. By combining match statistics, team history, and player performance metrics, I created a reliable forecasting system.

Next Steps:

From my perspective, I could enhance this model by:

- Fine-tuning the models to improve prediction accuracy
- Adding real-time data like weather and pitch conditions
- Including player availability updates and team news
- Developing a user-friendly dashboard for live predictions

These improvements would make my predictions more accurate and useful for cricket fans, fantasy players, and team analysts.

9. References

1. **IPL Complete Dataset (2008–2024)** – Patrick B. Kumar, Kaggle
2. **Scikit-Learn: Machine Learning in Python**
3. **Pandas: Python Data Analysis Library** – Wes McKinney
4. **Matplotlib: Visualization with Python** – John D. Hunter
5. EDA Analysis - code
6. Model Evaluation - code