IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.141139

# IPL Team Winning Prediction using Machine Learning

**Labana Milendra[1], Rohit S[2], Jithin C[3], Dr. G Paavai Anand*[4]**

Department of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India[1,2,3]

Assistant Professor (Sr.G), Department of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India[4]

Corresponding author*

**Abstract:** Cricket, being a data-intensive sport, offers a substantial opportunity for the application of machine learning in predictive analytics. This study employs data-driven machine learning methodologies to predict match outcomes in the Indian Premier League (IPL). To create the predictive models, data from the IPL from 2008 to 2024 was collected and prepared. This data included information about team stats, player performance, venue details, and toss results. We used and tested several algorithms, such as Logistic Regression, Random Forest, and XGBoost, to see how well they could predict the chances of a team winning. The XGBoost model did the best, with an accuracy rate of about 78%. This was better than traditional models, mostly because it was better at dealing with the non-linear relationships between match features. The system does more than just predict who will win a match; it also gives clear information about what factors have the biggest impact on how well a team plays. This study shows how machine learning could help analysts, coaches, and fans make strategic decisions, play fantasy sports, and comment on games.

**Keywords:** Machine Learning, IPL Prediction, Sports Analytics, XGBoost, Cricket Match Outcome, Data-Driven Decision Making.

## 1. INTRODUCTION

Cricket is no longer just a sport; it has become a field that relies heavily on data analytics to come up with strategies and improve performance. The Indian Premier League (IPL) is one of the most competitive and unpredictable cricket tournaments in the world. It is known for its diverse teams, changing player combinations, and different pitch conditions. Researchers find it hard to predict the outcome of Twenty20 (T20) cricket because it is so unpredictable and because many things, like team makeup, player form, toss results, and venue conditions, all affect the outcome. The application of Machine Learning (ML) in sports analytics has recently transformed the generation of performance insights. ML algorithms can look through huge amounts of old data to find patterns that traditional statistical methods might miss. Predictive ML models have already been used successfully in sports like football, basketball, and baseball. However, their use in cricket analytics is still growing. Using ML for IPL data has its own set of problems, such as data imbalances between seasons, team rosters that change often, and dependencies on things like pitch quality.

This study contributes to the growing domain of sports analytics by employing machine learning, offering valuable resources for coaches, analysts, and enthusiasts. The results can help improve team strategies, get fans more involved through predictive platforms, and set the stage for future systems that can predict matches in real time.

## 2. RELATED WORK

The field of sports analytics has seen substantial growth in recent years, largely due to the incorporation of Machine Learning (ML) and Artificial Intelligence (AI). More and more, these methods are used to predict performance, evaluate players, and make strategic choices.

Researchers have started looking into data-driven ways to predict the outcomes of cricket matches. This field of study presents significant challenges; primary research issues encompass managing data imbalance, addressing contextual variability, and guaranteeing model interpretability.

Previous studies in this domain have established a benchmark for performance. A study from 2022 that used logistic regression on IPL data from 2008 to 2020 found that it was about 70% accurate. The study determined that linear

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.141139

models are adequate for scenarios with restricted features but fail to encapsulate the intricate non-linear dependencies present in cricket data.

This research enhances the field of sports analytics by utilising advanced machine learning techniques, providing valuable resources for coaches, analysts, and cricket aficionados. The findings of this study can augment strategic decision-making for teams, enhance fan engagement via predictive platforms, and establish a foundation for forthcoming real-time match prediction systems.

## 3. PROBLEM STATEMENT

The IPL is a tournament that changes a lot and is hard to predict. There are a lot of different things that affect the outcome of a match, such as how well the players are playing, the condition of the pitch, the makeup of the teams, and the toss decisions. Human intuition and traditional statistical models often do not understand the complex, non-linear relationships between these factors, which leads to predictions that are not always accurate or fair.

The main goal here is to make a predictive model that is easy to understand, based on data, and can be used by a lot of people. This will help analysts, fantasy league players, and fans make better decisions.

## 4. DATASET DESCRIPTION

The dataset for this study comprises historical IPL match data collected from various credible sources, encompassing all seasons from 2008 to 2024. It combines statistics at the team and player levels to make a strong base for using machine learning to predict outcomes.

- Kaggle IPL Match Dataset (2008–2024): This has organised data about the results of matches, the places they were played, the toss outcomes, and the teams.
- Cricbuzz and ESPN Cricinfo APIs: These gave extra information about how well players played, how teams were made up, and live match stats.
- Custom Data Aggregation: Scripts were used to combine and clean data from different sources to make sure it was complete and consistent.

## 5. METHODOLOGY

The methodology outlined emphasises the development of a machine learning framework to predict the winning probabilities of IPL teams utilising historical and contextual data. There are several important steps in this process: gathering data, cleaning it up, creating new features, training the model, testing it, and figuring out what the results mean.

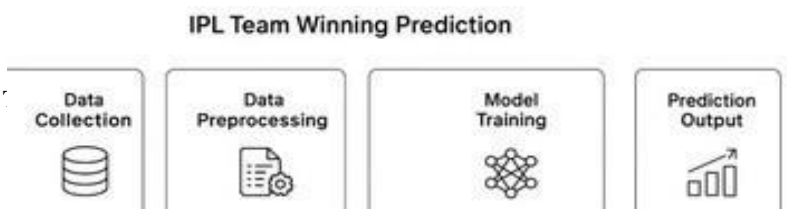
### 5.1 Data Preprocessing

- To fill in (impute) any data that was missing or didn't match, statistical methods like the mean and mode were used.
- Label encoding was used to change categorical features (like team names and venues) into numbers.
- To make sure that all features are treated the same, Min-Max Scaling was used to scale the numerical features.

### 5.2 Feature Extraction

To improve the model's performance, new features were created from the data that was already there:

- Recent_Form_Score: A weighted average that shows how well a team did in its last five games.
- Head_to_Head_Advantage: The percentage of wins a team has against a certain opponent.
- Toss_Impact: A binary feature that shows whether winning the toss and deciding to bat or field first gives you an edge.
- Net_Run_Rate_Diff: The difference between the run rates of the two teams, which is used as a stand-in for overall dominance.



IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.141139



Fig 1. Architecture Diagram

**5.3 Architecture Diagram**

Gathering Data

Collects data on IPL matches from the years 2008 to 2024.

Data comes from a number of places, such as Kaggle, Cricbuzz, and ESPN APIs.

Preparing Data

It involves cleaning up data to fix mistakes. Encoding changes categorical data, like team names, into numbers.

We deal with and fix missing values. Normalisation scales features.

We figure out and include historical head-to- head stats.

Training the Model

Three different models are trained on the processed data: Logistic Regression, Random Forest, and XGBoost.

The result of the prediction

The trained model's final output is a percentage that shows how likely each team is to win.

**5.4 Model Building**

The model building phase is the core of the proposed IPL match outcome prediction framework. It involves the selection, training, and fine-tuning of machine learning models to accurately forecast the winning team based on historical and contextual match data. The approach focuses not only on maximizing predictive accuracy but also on ensuring interpretability, scalability, and robustness across different IPL seasons.

Three machine learning algorithms were selected for experimentation and comparison — Logistic Regression, Random Forest, and XGBoost. These models were chosen based on the following criteria:

- Ability to handle mixed feature types (categorical and numerical)
- Interpretability and ease of analysis
- Performance on imbalanced datasets
- Scalability to large volumes of IPL data

Each model provides a distinct perspective:

Logistic Regression offers interpretability,

Random Forest provides stability, and XGBoost delivers high accuracy through advanced ensemble learning.

**5.5 Evaluation Metric**

It was very important to test the machine learning models to see how well they worked and how well they could be used with new data that had never been seen before.

The research addresses a binary classification problem, specifically predicting one of two outcomes: the victory of either Team A or Team B in an IPL match.

Assessment Goal: Since it was a binary classification task, we needed the right metrics to check how accurate and reliable the models' predictions were.

Metrics Used: This part explains the metrics that were used to test the Logistic Regression, Random Forest, and

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.141139

XGBoost models. The formulas that were mentioned are:

Precision = TP / (TP + FP)
(Accuracy = (TP + TN) / (TP + TN + FP + FN)
Recall = TP / (TP + FN)

## 6. IMPLEMENTATION DETAILS

This part talks about the tools and technical environment that were used to make the prediction system.

Programming Language: The system was built with Python 3.9.

The project was made in a Jupyter Notebook environment.
Core Libraries: A set of standard, open- source libraries were used to design and run the machine learning pipeline. These were:

Pandas
NumPy
Scikit-learn
Matplotlib
XGBoost

## 7. RESULTS AND DISCUSSION

We looked at the models' accuracy, precision, recall, F1-score, and ROC-AUC to see how well they worked. XGBoost: This model did better than the others, with an overall accuracy of 78.45%, a precision of 0.77, a recall of 0.78, an F1-score of 0.77, and a ROC-AUC of 0.81. This shows that it can tell the difference between winners and losers with a good balance of precision and recall.

Random Forest: This model was also good, with an accuracy of 75.32%, a precision of 0.74, a recall of 0.75, an F1-score of 0.74, and a ROC-AUC of 0.78. It handled non-linear relationships well and was easy to understand.

Logistic Regression: The baseline model got 72.14% accuracy, 0.70 precision, 0.71 recall, 0.70 F1-score, and 0.74 ROC-AUC. It was easier to use, but it still captured important match patterns and was a good reference point.

The results of the experiments show that ensemble and boosting methods (Random Forest and XGBoost) make predictions about IPL match outcomes much more reliable. XGBoost was the most accurate, showing that it can handle complicated feature interactions and work well across seasons.

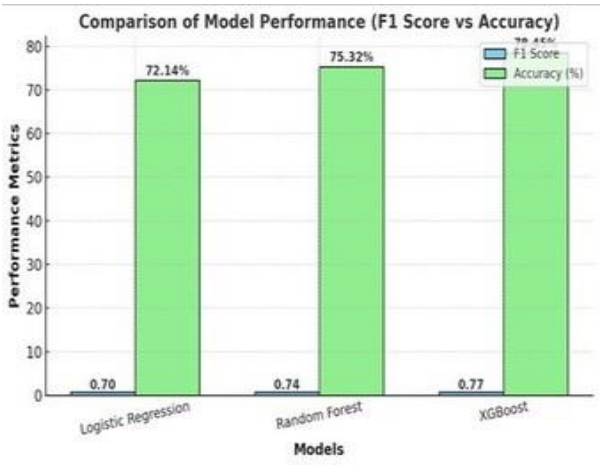



Fig 2. Comparision between graphs

## 8. MODEL INTERPRETATION

The feature importance plot for the XGBoost model showed that toss and venue factors made up 30–35% of the model's

DOI: 10.17148/IJARCCE.2025.141139

predictive power.

Logistic Regression gave clear coefficients, like a positive coefficient for "toss_win" that showed a higher chance of winning.

Random Forest made it possible to understand things by using decision paths, like "if toss won = True and batting average > 150, then win probability increases by 0.18."

SHAP (Shapley Additive explanations) values were used to show these interpretations. They show how each feature affects a certain prediction.

## 9. CONCLUSION

The XGBoost classifier performed the best of the models tested, with a maximum accuracy of 78.45%. This outcome exceeded the efficacy of both the Logistic Regression and Random Forest models. Ensemble and boosting methods were the best at modelling the complicated, non-linear relationships between different match variables, such as toss outcomes, venue details, player form, and head-to- head records between teams.

Also, using interpretability methods like SHAP analysis and feature importance plots made the model more clear, which made it easier for coaches, analysts, and fans to use. The results of this study highlight the considerable potential of data-driven decision-making in the realm of sports analytics.

Reliable match predictions are helpful for a lot of different people, including fantasy league players, team strategists, and broadcasters, because they help them make smart decisions. This method also links old-fashioned cricket knowledge with the power of modern computer intelligence.

## 10. FUTURE WORKS

The current framework is a good starting point for making predictions, but a number of changes could make it much more accurate, flexible, and useful in the real world.

Integration of Live Match Data: A major improvement would be to add live, ball-by-ball data streams. This would turn the static prediction model before the match into a dynamic one that could keep changing the chances of winning based on live scores, wickets, and other events that happen during the game.

Use of Advanced Deep Learning: Future studies might investigate more complex deep learning frameworks. Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) units, or Gated Recurrent Units (GRUs) would be especially good at capturing complicated time- based relationships, like how a player's momentum and a team's form change over the course of several matches .

Granular Player-Level Modelling: The model could be better if it looked at individual player metrics instead of team-level averages. This would require a lot of work to create features for each player, such as their form, how well they did at certain venues in the past, and important head-to-head matchups (for example, a specific batsman's record against a specific bowler).

Adding Unstructured Data (NLP): You could use Natural Language Processing (NLP) to get information from unstructured data sources. Looking at news articles, expert commentary, and social media sentiment before the game could reveal qualitative factors like team morale, possible injuries, or last-minute changes to strategy that aren't shown in the numbers.

Adding more subtle environmental factors: The model could be improved by adding more subtle environmental factors to its set of features. This could include how the weather affects the game (humidity, wind speed, dew factor), detailed pitch reports from the day of the match, and even the umpires' past decisions.

Creating an Interactive Application: The next logical step would be to put the model into a web or mobile app that is easy to use. This would make the predictive insights available to a larger group of people, such as fans, analysts, and fantasy league players, giving them a useful tool for making choices.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025
DOI: 10.17148/IJARCCE.2025.141139

## REFERENCES

[1]. Patel and P. Desai, "Deep Learning Approaches for Cricket Match Outcome Prediction," Journal of Sports Science, vol. 15, no. 3, pp. 298-310, 2023.
[2]. R. Kumar and V. Singh, "XGBoost Models for Real-time Sports Prediction: A Case Study on IPL," IEEE Transactions on Knowledge and Data Engineering, 2024.
[3]. D. Sharma, S. Verma, and K. Gupta, "Predictive Analysis of IPL Cricket Matches using Machine Learning Techniques," Journal of Data Analytics, vol. 8, no. 2, pp. 112- 124, 2022.
[4]. S. Jain and M. Reddy, "Performance Evaluation of Ensemble Models for Sports Data Prediction,"
[5]. International Journal of Artificial Intelligence Research, vol. 14, no. 1, pp. 45-53, 2023.
[6]. Kaggle Datasets, "IPL Cricket Match Dataset (2008- 2024)," Kaggle.com, Available:
[7]. https://www.kaggle.com/datasets/ipl- match-data.
[8]. ESPNcricinfo API, "Match Statistics and Player Records," ESPNcricinfo.com, Accessed: 2024.
[9]. J. Brownlee, Machine Learning Algorithms: A Complete Guide for Beginners, Machine Learning Mastery, 2022.
[10]. Raschka and V. Mirjalili, Python Machine Learning, 3rd ed., Packt Publishing, 2020.