IPL Prediction Using Machine Learning

*Abhineet Menon**[1], *Dhruv Khator*[2], *Dhru Prajapati*[3], *and Archana Ekbote*[4]

Abstract

Cricket is amongst the most popular sports in the world. Indian Premier League, more commonly known as IPL is the biggest domestic cricket league in the world. It generates a lot of revenue along with excitement among fans. Many bookers, bettors, and fans like to predict the outcome of a particular match which changes with every ball. This project studies and compares different Machine Learning techniques that can be applied to predict the outcome of a match. Features like team strength and individual strength of a player are also included along with conventional features like toss, home ground, weather and pitch conditions that are taken into account for predicting the result. Machine Learning algorithms such as Naïve Bayes, Random Forest Classifier, Logistic Regression, XGBoost, AdaBoost, and Decision Tree are selected to determine the predictive model with highest accuracy.

Keywords : AdaBoost, Decision Tree, Indian Premier League, Machine Learning, Naïve Bayes, Logistic Regression, Random Forest Classifier, XGBoost

I. INTRODUCTION

A. Problem Statement

IPL is considered to be the most popular T20 league in the world and it creates a lot of excitement amongst the fans of the game. Many people like to predict the outcome of the game beforehand and like to formulate their own strategies and create their own fantasy teams. There are many systems already present that predict the outcome of an IPL match, but all the factors affecting a particular match are not taken into consideration. The main aim of this work is to create a prediction system with accuracy as high as possible, which will help users to create their fantasy teams with much ease.

B. Motivation

There is always a rush in our heads regarding the results of a cricket match. Friendly betting was one of the factors which motivated us for predicting the winners of IPL matches. As the law-obliged sports gambling industry is growing at a faster rate than ever, it can be useful for people to get the idea of the outcome before a match begins. So, we thought of creating a prediction system with accuracy as high as possible which will make it easy to create fantasy teams.

II. LITERATURE REVIEW

Many researches have contributed towards predicting the results of cricket matches. Here, some of the prominent works done by researchers are discussed.

Manuscript Received : May 3, 2022 ; Revised : May 12, 2022 ; Accepted : May 15, 2022. Date of Publication : June 5, 2022.

A.Menon[*1], *Student*, Email : menonabhineet@gmail.com ; ORCID iD : https://orcid.org/0000-0002-2641-651X
D. Khator[2], *Student*, Email : dhruvkhator19@gmail.com ; ORCID iD : https://orcid.org/0000-0002-1659-7230
D.Prajapati[3], *Student*, Email : dhruprajapati18@gmail.com ; ORCID iD : https://orcid.org/0000-0002-4627-2883
A. Ekbote[4], *Assistant Professor*, Email : archana.ekbote@vcet.edu.in ; ORCID iD : https://orcid.org/0000-0001-8342-3612
[1,2,3,4]Vidyavardhini's College of Engineering and Technology, K.T. Marg, Vartak College Campus Vasai Road, Vasai-Virar, Maharashtra - 401 202.

DOI : https://doi.org/10.17010/ijcs/2022/v7/i3/171267

In [1] the researchers investigated the match by considering multiple features. The authors have taken two cases into consideration:

(1) Home ground advantage and

(2) Winning the toss. When the algorithms were applied to the given two cases, the results were as follows:

Case 1 : Home Ground Advantage:

(1) Naïve Bayes – 57%
(2) Model decision tree – 56%
(3) Random forest – 54%
(4) KNN –52%

Case 2 : Winning the toss:

(1) KNN – 62%
(2) Naïve Bayes – 52%

Their model provided better prediction results by using the second case of winning the toss and applying KNN algorithm to it. They made their model on the basis of past results of matches and have not included the player's data. In [2] the researchers applied different algorithms on their dataset and the results they obtained were as follows:

(1) Naïve Bayes – 71%
(2) KNN – 66%
(3) Logistic Regression – 64%
(4) Random Forest – 63%
(5) Support Vector Machines – 59%

They did not use features like weather conditions, venue, and pitch conditions which play a vital role in a particular cricket match.

In [3] the researchers applied various Data Science techniques to foretell the result of an IPL match. The datasets were scraped and obtained from espncricinfo [4], iplt20 [5], and Kaggle [6]. Batting, bowling, and team strength of the top 11 players who have played most matches for a particular team were considered and the strength was calculated using Dream11 points calculation system [7]. Cumulative team strength was also considered (i.e. the team strength of previous years). The analytic hierarchy process was used for getting priority order and weights of attributes that are considered for calculating batting and bowling average. Win rate strength was calculated by multiplying team strength with win rate which was then used to predict the outcome of a particular match. The results were as follows:

(1) Naïve Bayes – 58.23% ± 5.5%
(2) Adaboost – 60.03% ± 6.2%
(3) Logistic Regression – 57.77% ± 5.8%
(4) Support Vector Machines – 58.42% ± 5.69%
(5) KNN – 53.47% ± 5.2%
(6) XGBoost – 55.42 ± 5.9%
(7) Extra Trees Classifier – 59.51% ± 5.9%
(8) Random Forest Classifier – 60.04 ± 6.3%

Highest accuracy was provided by Random Forest and AdaBoost (60.4% ± 6.3% and 60.3% ± 6.2% respectively), and lowest accuracy of 53.47% ± 5.2% was provided by KNN. Even if you consider the best case result, the maximum accuracy will not go beyond 67% which is not that good a result. Though the methodology was good as compared to other papers, there were a few drawbacks related to team strength calculation and win rate calculation. Nothing was mentioned about the debutants of the competition. Win rate was figured out by fractionating the total number of matches won by the team to the total number of matches played by the team, but there were few factors that affected the match like weather conditions, home ground advantage, and toss winning advantage which were not considered.

The authors of [8] used various Machine Learning algorithms to predict results and analyze an IPL match. Two datasets were used, one having ball-to-ball information of the whole IPL till 2019 and other one was having information of all the matches. The authors made use of various external factors like weather, venue, etc. that affect a match were taken into consideration to make a training model. Batting and bowling performances of individual players were also taken into consideration in addition to the past performances of the team. Other factors were also considered by them. The results were as follows:

(1) Logistic Regression – 95.91%
(2) Decision Tree – 87.75%

(3) SVM – 83.67%

(4) Random Forest – 83.67%

(5) Naïve Bayes – 81.63%.

III. COMBINED STUDY

On studying the given related papers thoroughly, we got some useful information, which is presented in Table I.

TABLE I.
COMPARISON OF ALGORITHMS USED IN PREVIOUS PAPERS

Algorithms	Maximum Accuracy
Naïve Bayes	81.63%
Random Forest Classifier	83.67%
Logistic Regression	95.91%
XGBoost	55.42 ± 5.9%
AdaBoost	60.4 ± 6.3%
DecisionTree	87.5%
SVM	83.67%
KNN	66%

IV. PROPOSED METHODOLOGY

A. Block Diagram

Fig. 1 shows the block diagram.

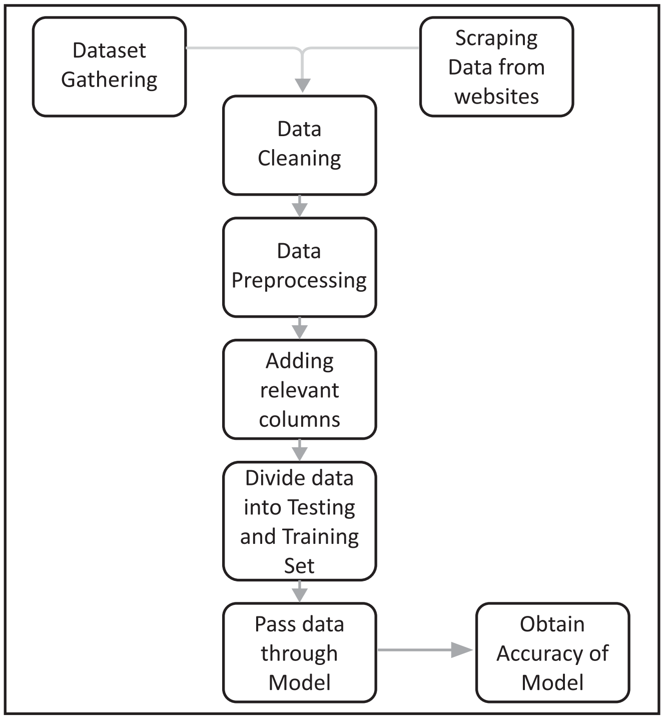

Fig.1. Block Diagram

B. Dataset

The dataset used for analysis and prediction was collected from Kaggle [6] and furthermore, data was scraped from espncricinfo [4] and iplt20 [5] using the Beautiful Soup library of Python [9].

C. Data Preprocessing

For applying the Machine Learning algorithms, string data will be converted into numerical data as the algorithms works better with numerical values.

All the unnecessary attributes such as names of umpires, venue, date, player of the match, method, and eliminator were removed from the dataset to get more accurate results.

Further, all the match rows that were dismissed, drawn, or null were eradicated.

Newer needed features such as team strength and team points were added to the dataset. Team points were calculated on the basis of previous matches of each team.

D. Features

1) ***Toss :*** Toss is considered as the most important factor sometimes. After winning a toss, a team can select either to bat or bowl first, which when analyzed can provide an upper hand to the toss winning team over the other team even before the match starts. Toss analysis of our dataset tells us that 55.8% times toss winning team wins the match (Fig. 2). However, toss is not the only factor on which the outcome of a match is decided.

2) ***Team Points :*** Team points are used to represent the performance of a team by using the results of the team's

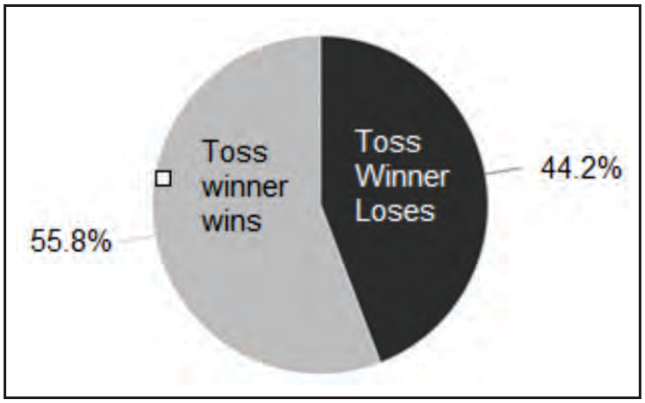

Fig. 2. Toss Factor

previous matches. The previous 5 matches of each team were considered for calculating team points. A win was counted as 2 points and a loss was counted as 0 points as it is done officially in the IPL. Team points analysis of our dataset tells us that 59.3% times team having greater points wins the match (Fig. 3).

3) ***Team Strength :*** Team strength is used to represent the cumulative strength of all the players of a team. For calculating team strength we made use of the player-points data of every year available on IPL's official website. Each team's top 11 players were selected and their average points were taken to get team strength. Team strength analysis of our data-set tells us that 60.9%(Fig. 4) times team having greater strength wins the match.

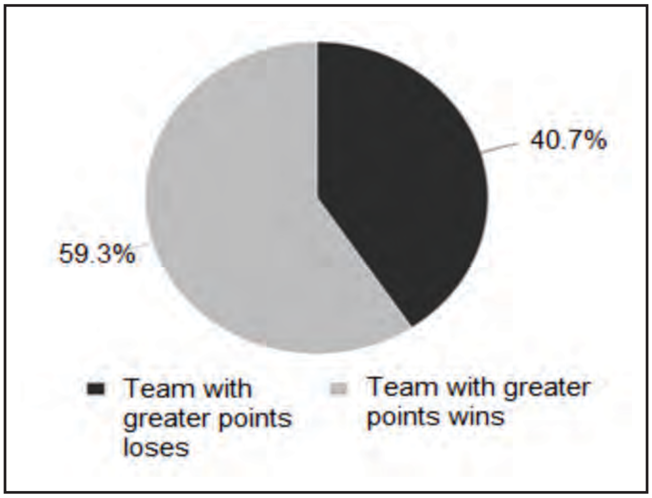

Fig. 3. Team Points Factor

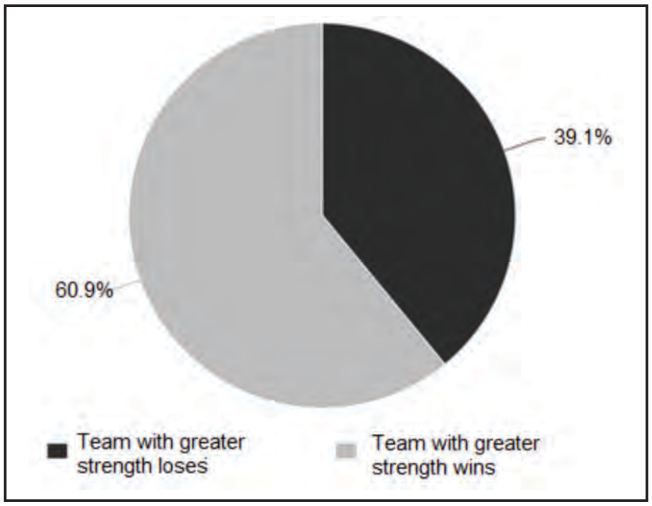

Fig. 4. Team Strength Factor

E. Algorithms

1) ***Ada Boost :*** Ada Boost is short for Adaptive Boosting. It is an ensemble ML method. It learns from the mistakes of weak classifiers in every iteration and hence, converts the same to strong classifiers.

Parameters of AdaBoost algorithm used are:

(a) n_estimators
(b) learning_rate

2) ***Decision Tree :*** Decision Tree algorithm is used for regression as well as classification. It begins with root node and finishes with decision node called leaf. It has root, decision, and terminal nodes. It is very convenient in case of decision based examples.

Parameters of Decision Tree algorithm used are:

(a) criterion
(b) min_samples_split
(c) min_samples_leaf
(d) max_features
(e) max_depth

3) ***Random Forest:*** The Random Forest classifier contains numerous separate decision trees. Each tree in the forest calculates a class prediction and the class that has highest votes will be given out as the prediction of the whole model. It operates like an ensemble. For higher accuracy the number of trees should be higher.

Parameters of Random Forest algorithm used are:

(a) n_estimators
(b) min_samples_split
(c) min_samples_leaf
(d) max_features
(e) max_depth
(f) bootstrap

4) ***XgBoost :*** XgBoost stands for Extreme Gradient Boosting that uses decision tree algorithm to predict small to medium structured or tabular data. Ensemble method is used which gives us predictive power of multiple iterations. Using Bagging or Boosting, highly variant

behaviour is reduced, which in turn helps to increase the accuracy.

Parameters used for XGBoost algorithm are:

(a) n_estimators
(b) learning_rate
(c) max_depth

5) ***Logistic Regression :*** This algorithm is used in cases where the target variable is categorical. It is predicted using independent variables. It is used in the case of classification and not regression problems.

Parameters of Logistic Regression that were used are:

(a) penalty
(b) tol
(c) solver
(d) max_iter
(e) C

6) ***Naïve Bayes :*** This algorithm is based on the Bayes Theorem of conditional probabilities. Classification problems can be solved using this algorithm. In this, each feature is supposed to be independent of the other and it contributes to the result independently.

Parameter used for Naïve Bayes algorithm are:

(a) var_smoothing

F. Hyper Parameter Tuning

Hyper-parameters of algorithms are the parameters the values of which need to be set before building the model and beginning the Machine Learning process. Their values cannot be obtained from training. Tuning a model for a certain problem can be done using Random Search or Grid Search. By doing this, the most efficient prediction of the learning model can be obtained.

➯ ***RandomizedSearchCV :*** RandomizedSearchCV (Cross-Validation) is used to basically optimize the hyper- parameters. Random combinations are used to train the model. A sampling distribution is used for every hyper-parameter to do random search which allows us to administer the attempted combinations.

➯ ***GridSearchCV :*** GridSearchCV (Cross-Validation) is used to optimize the hyper-parameters of a model. It uses traditional trial and error to get all the combinations. After this cross validation is done, it helps us to get the best accuracy. Cross Validation checks show how a model generalizes itself to an unconstrained dataset.

For tuning the algorithms, GridSearchCV was used in the case of AdaBoost and Naïve Bayes algorithm while the other algorithms were tuned using RandomizedSearchCV.

V. RESULTS AND CONCLUSION

The conclusion that was obtained after building the models of all the algorithms mentioned earlier is presented in Table II.

It can be seen in Fig. 5 that the highest accuracy achieved by the current investigation is 89.04% provided by XGBoost algorithm and the highest accuracy achieved was 95.91% provided by Logistic Regression algorithm for the previous investigation. Dataset can be a huge

TABLE II.
RESULTS AND COMPARISON OF VARI.OUS MLTECHNIQUES

Algorithm	MaximumAccuracy
Naïve Bayes	59.366%
Random Forest Classifier	84.726%
Logistic Regression	56.196%
XGBoost	89.049%
AdaBoost	59.942%
Decision Tree	70.317%

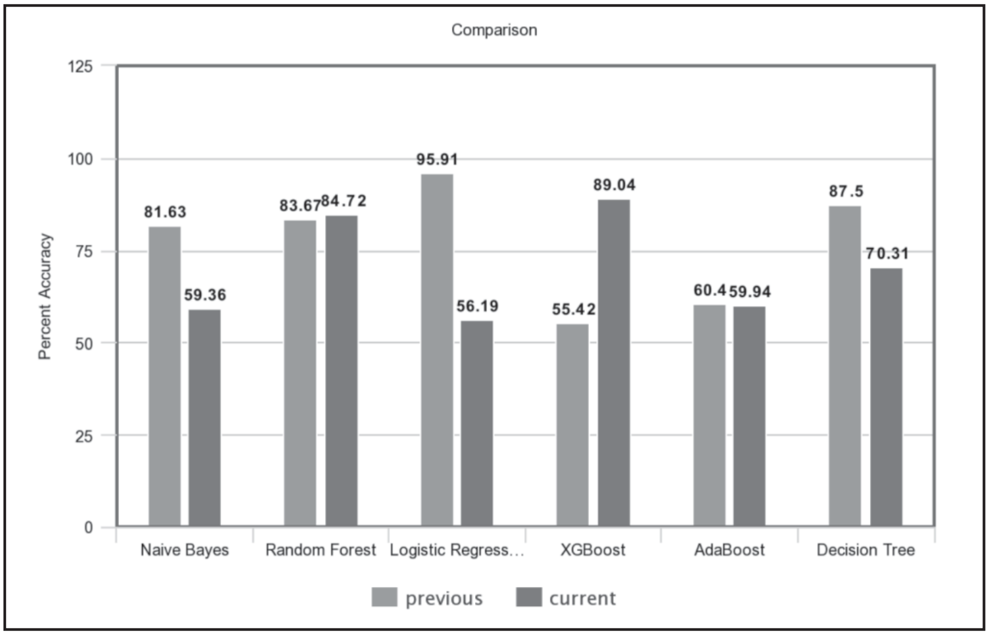

Fig. 5. Comparison of Studies

factor because of which the accuracy of current investigation is differing from previous investigation. Other than that, parameters of a particular algorithm chosen in current investigation may be different from the parameters chosen in previous investigation for that algorithm. There are many different features selected by previous investigation as compared to current investigation which can also be the reason of difference in accuracy for same algorithms.

VI. FUTURE SCOPE

As the IPL seasons go on, the dataset can be updated to the current season to get more precise results and for fantasy team builders the best 11 players of the match can also be predicted based on the fantasy points they have previously obtained.

ACKNOWLEDGMENT

The authors would like to thank Dr. Harish Vankudre, their Principal for extending his support to carry out this research. They express deep sense of gratitude and thanks to Dr. Ashish Vanmali, their Head of Department.

They would also like to thank some other faculty members of Information Technology Department, Prof. Anagha Patil and Dr. Madhavi Waghmare for their valuable ideas and timely assistance.

AUTHORS' CONTRIBUTION

Abhineet Menon, Dhruv Khator, Dhru Prajapati, and Archana Ekbote are the authors of this paper. The dataset was cleaned and modified by Dhru Prajapati. Dhruv Khator scraped data from the internet and added it to the dataset. Abhineet Menon ran the various models and did the GUI work. Analysis work was done by all three. Dr. Archana Ekbote provided timely guidance and assistance when required.

CONFLICT OF INTEREST

The authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in the manuscript.

FUNDING ACKNOWLEDGEMENT

The authors have not received any financial support for the research, authorship, and/or for the publication of the article.

REFERENCES

[1] K. Kapadia, H. Abdel-Jaber, F. Thabtah, and W. Hadi. "Sport ana lytics for cricket game results using machine

learning: An experimental study," *Appl.Comput. Inform.,* vol. ahead-of-print, no. ahead-of print, 2019, doi: 10.1016/j.aci.2019.11.006.

[2] P. K. Dubey, H. Suri, and S. Gupta, “Naïve Bayes algorithm based match winner prediction model for T20 Cricket," in S. S. Dash, S. Das, B. K. Panigrahi (eds) *Intell. Comput. Appl.. Advances Intell. Syst. Comput.,* vol 1172, 2021. Springer, Singapore, doi: 10.1007/978-981-15-5566-4_38.

[3] A. Tripathi, R. Islam, V. Khandor, and V. Murugan, "Prediction of IPL matches using Machine Learning while tackling ambiguity in results," *Indian J. Sci. Technol.,* vol. 13, no. 38, pp. 4013–4035, 2020, doi: 10.17485/IJST/v13i38.1649.

[4] Espncricinfo. [Online].Available: https://www.espncricinfo.com/

[5] Indian Premier League. https://www.iplt20.com/

[6]“kaggle”.[Online]. Available: https://www.kaggle.com/datasets

[7] Dream11. https://www.dream11.com/

[8] H. Barot, A. Kothari, P. Bide, B. Ahir, and R. Kankaria, "Analysis and prediction for the Indian Premier League," *Int. Conf. Emerg. Technol.*, 2020, pp. 1–7, doi: 10.1109/INCET49848.2020.9153972.

[9]Beautiful Soup Python library. https://pypi.org/project/beautifulsoup4/

About the Authors

Abhineet Menon is pursuing B.E. (Information Technology) from the University of Mumbai, India. His research interest includes Machine Learning and Data Analytics.

Dhruv Khator is pursuing B.E. (Information Technology) and from the University of Mumbai, India. His research interests include User Experience and Machine Learning.

Dhru Prajapati is pursuing B.E.(Information Technology) from the University of Mumbai, India. His research interests include distributed systems, security, and Machine Learning.

Archana Ekbote is Assistant Professor with the Department of Information Technology at Vidyavardhini's College of Engineering & Technology, University of Mumbai, India. Her research interests include signal and image processing, and pattern recognition.