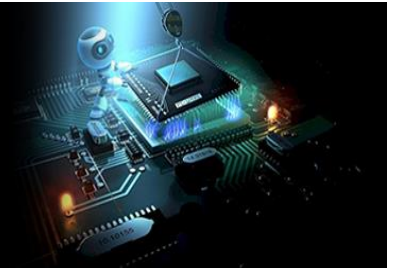


# International Journal of Engineering in Computer Science



E-ISSN: 2663-3590  
P-ISSN: 2663-3582  
IJECS 2023; 5(2): 13-20  
Received: 25-05-2023  
Accepted: 03-07-2023

**Chandra Sekhar Sanaboina**  
Department of Computer  
Science and Engineering,  
University College of  
Engineering Kakinada,  
Jawaharlal Nehru  
Technological University  
Kakinada, Andhra Pradesh,  
India

**Kalaparthi Vikram Kumar**  
PG Student, Department of  
Computer Science and  
Engineering, University  
College of Engineering  
Kakinada, Jawaharlal Nehru  
Technological University  
Kakinada, Andhra Pradesh,  
India

**Corresponding Author:**  
**Chandra Sekhar Sanaboina**  
Department of Computer  
Science and Engineering,  
University College of  
Engineering Kakinada,  
Jawaharlal Nehru  
Technological University  
Kakinada, Andhra Pradesh,  
India

## Win probability prediction for IPL match using various machine learning techniques

**Chandra Sekhar Sanaboina and Kalaparthi Vikram Kumar**

**DOI:** <https://doi.org/10.33545/26633582.2023.v5.i2a.94>

### Abstract

The paper's primary objective is to predict the win probability for both the batting and bowling teams in the IPL (Indian Premier League) match using various machine learning algorithms. This paper tackles the challenge of forecasting the result of a match determined by the Target, Net Run Rate (NRR), Current Run Rate (CRR), Score, Fall of Wickets, and the Technique each team employs during each game. For making predictions, three machine learning models Logistic Regression, Random Forest, and Naive Bayes were utilized. The Logistic Regression and Random Forest Algorithms provide the best results and have an accuracy rate of 82% and 96% respectively whereas Naive Bayes has an accuracy rate of 63%.

**Keywords:** IPL, Probability, NRR, CRR, Naive Bayes (NB), Logistic Regression (LR), Random Forest (RF), ICC, BCCI, Machine Learning (ML)

### 1. Introduction

People of all ages are absolutely obsessed with watching and playing cricket, the most thrilling and entertaining game. In 2008, BCCI (The Board of Control for Cricket in India) got the opportunity and conducted a T-20 league known as IPL, which is approved by ICC (International Cricket Council). The IPL is among the top T-20 cricket competitions going on right now. Every IPL season begins with a match between eight teams; more recently, in 2022, two extra teams were added. Four teams go from the first stage to the next level, the elimination round, and two teams advance to the championship match, where one winner will be determined. The Location, Player Performance, NRR, CRR, Target, Fall of Wickets, Toss, Power Play Performance, etc., all have an impact on the outcome of each IPL match.

This project falls under the supervised machine learning area. As a result, we may choose the right models from supervised learning. The primary goal of this project is finding probabilistic values which can be provided by Naive Bayes and Logistics Regression. Additionally, Random Forest can be used for comparison. The project's results are predicted using logistic regression and Naive Bayes models with respective accuracy rates of 82% and 63%. Additionally, random forest also predicts the results with 96% accuracy.

### 2. Related Work

Data science has infused itself into every industry since the year 2000, including cricket. In the past, a number of researchers have made contributions to outcome prediction. Agrawal *et al.*,<sup>[1]</sup> explored the issue of anticipating the match's unknown winner in the IPL. They applied machine learning models like Support Vector Machine, Classification Tree, and Naive Bayes Algorithm. According to Barot *et al.*,<sup>[2]</sup> report, the IPL is the world's most-watched domestic T20 league. Along with the customary qualities like the toss, location of the games, etc., several important factors like team form and team strength have been included to forecast the outcome of the match. Match predictions have been made using machine learning algorithms including SVM, Logistic Regression, Random Forest, and Naive Bayes. Forecasting game outcomes have been identified by Hodge *et al.*,<sup>[3]</sup> as the objective of esports analytics research. They presented the initial comprehensive analysis of a real-time win prediction in a competitive e-sport. Using traditional machine learning techniques, feature engineering, and optimization, their model is up to 85% accurate after 5 minutes of games.

A series of multinomial logistic regression models were used in Ref. 4 to forecast match outcome probability at the beginning of each session. They look into how session-to-session variations in outcome (chance of a victory, tie, or defeat) and covariate impacts. The probability of winning a match in tennis is determined hierarchically by Campus<sup>[5]</sup> approach. The probability of each player winning a point is determined from player data. These probabilities can be used to calculate the chances of winning a game at any moment in a live game. The issue of the third innings declaration in test cricket is discussed by Scarf and Shi<sup>[6]</sup>. The primary objective of these tools provides match outcome odds based on the match's location at a potential announcement point. Through the use of a multinomial logistic regression technique, these probabilities are calculated. Match outcome probabilities are calculated based on the target and run-rate.

Hayter<sup>[7]</sup> analyses the win probabilities using both central and non-central t-distributions. It also provides various examples for those techniques. Maher<sup>[8]</sup> offers a Markovian model which provides a more precise prediction of each game's outcome in their investigation. This takes into account estimates of the probability of holes being won or halved in the various formats of fourballs, foursomes, and singles, based on historical Ryder Cup and other key professional golf event data. The model also gives a probability distribution for the result of the event. Asif and Mchale<sup>[9]</sup> outline a method for predicting cricket one-day international match results while the match is still going on. As the game goes on, the logistic regression model is let to smoothly change. They provide two matches as examples to show how to use their model and analyse the match outcome probability. They interpret the quantitative similarity of the forecasts as proof that their modelling strategy is sound.

### 3. Proposed Method

#### 3.1 Logistic Regression

Logistic regression falls within the domain of supervised learning. It anticipates a dependent categorical variable's outcome. The outcome must therefore be a definite or classification value. It may either be True or False, 0 or 1, or Yes or No. Using logistic regression, it is possible to

quickly pinpoint the elements that will be effective when categorizing observations using different types of data.

##### 3.1.1 Logistic regression equation

The logistic regression formula is produced from the linear regression formula. The preceding are the mathematical approaches to creating equations for logistic regression. The straight line's formula is shown in "Eq. (1)".

$$Y = b_0 + b_1x_1 + b_2x_2 + \dots + b_nx_n \quad \text{Eq. (1)}$$

In order to account for this, let's simplify the above formula by (1-y), since y in Logistic Regression may only be between 0 and 1 this will be shown in "Eq. (2)".

$$\text{If } y = 0, \text{ then } \frac{y}{1-y} = 0; \text{ if } y = 1, \text{ then infinity} \quad \text{Eq. (2)}$$

However, we require a spectrum between - [infinity] and + [infinity] in order to take the equation's logarithm, which is as follows in "Eq. (3)".

$$\log \left[ \frac{y}{1-y} \right] = b_0 + b_1x_1 + b_2x_2 + \dots + b_nx_n \quad \text{Eq. (3)}$$

The above "Eq. (3)" is described as the final logistic regression equation.

##### 3.1.2 Process of logistic regression

**Step 1:** Data preparation

**Step 2:** Fitting the practice set for logistic regression

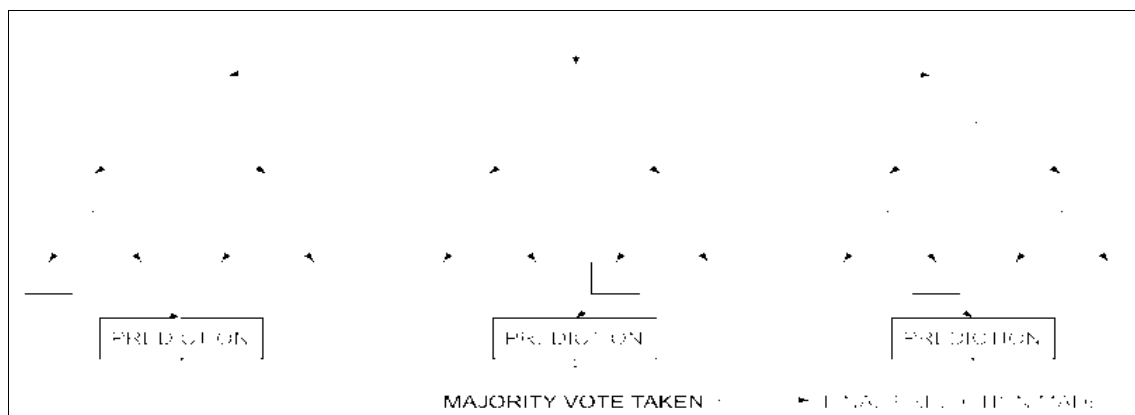
**Step 3:** Estimating the test outcomes

**Step 4:** Test the correctness of the results

**Step 5:** Displaying the outcome of the test set

#### 3.2 Random Forest Classifier

Random forest is a supervised machine-learning technique commonly used to solve classification and regression problems. Using a variety of samples, it builds decision trees and utilizes their best for classification and regression. The Random Forest algorithm's design is depicted in Fig.1.



**Fig 1:** Architecture of Random Forest Algorithm

##### 3.2.1 Random forest algorithm

The stages below can be used to demonstrate the functioning procedure

**Step 1:** Pick K sample points randomly from the given dataset.

**Step 2:** Construct the decision trees linked to the chosen

points of data.

**Step 3:** Select N to build the decision trees we desire.

**Step 4:** Steps 1 and 2 should be repeated.

**Step 5:** Find the predictions for any latest data points for each decision tree, then classify them under the most popular category.

### 3.3 Naïve Bayes Classifier

The Naive Bayes algorithm utilizes the Bayes theorem to solve classification issues. To categorize texts, it mostly uses a sizable training set. Because it uses a statistical classifier, its predictions are based on the probability that a specific event would occur. The formula for Baye's theorem is given in "Eq. (4)".

$$p\left(\frac{A}{B}\right) = \frac{P(B/A)*P(A)}{P(B)} \quad \text{Eq. (4)}$$

Where,

The posterior probability, or  $P(A|B)$ , measures the likelihood that a given hypothesis (A) will really occur.

$P(B|A)$  stands for Likelihood Probability, which measures how likely it is based on the evidence at hand that a given hypothesis is correct.

$P(A)$  stands for Priority probability, which is the likelihood of a theory before seeing the evidence.

The probability of evidence is marginal probability, or  $P(B)$ .

#### 3.3.1 Working of naïve bayes classifier

**Step 1:** Using the provided dataset, create frequency tables.

**Step 2:** Make a likelihood table by calculating the probabilities of the given attributes.

**Step 3:** Now, determine the posterior distribution using the Bayes theorem.

## 4. Methodology

### 4.1 Proposed system's architecture

The proposed system's design is depicted in Fig 2.

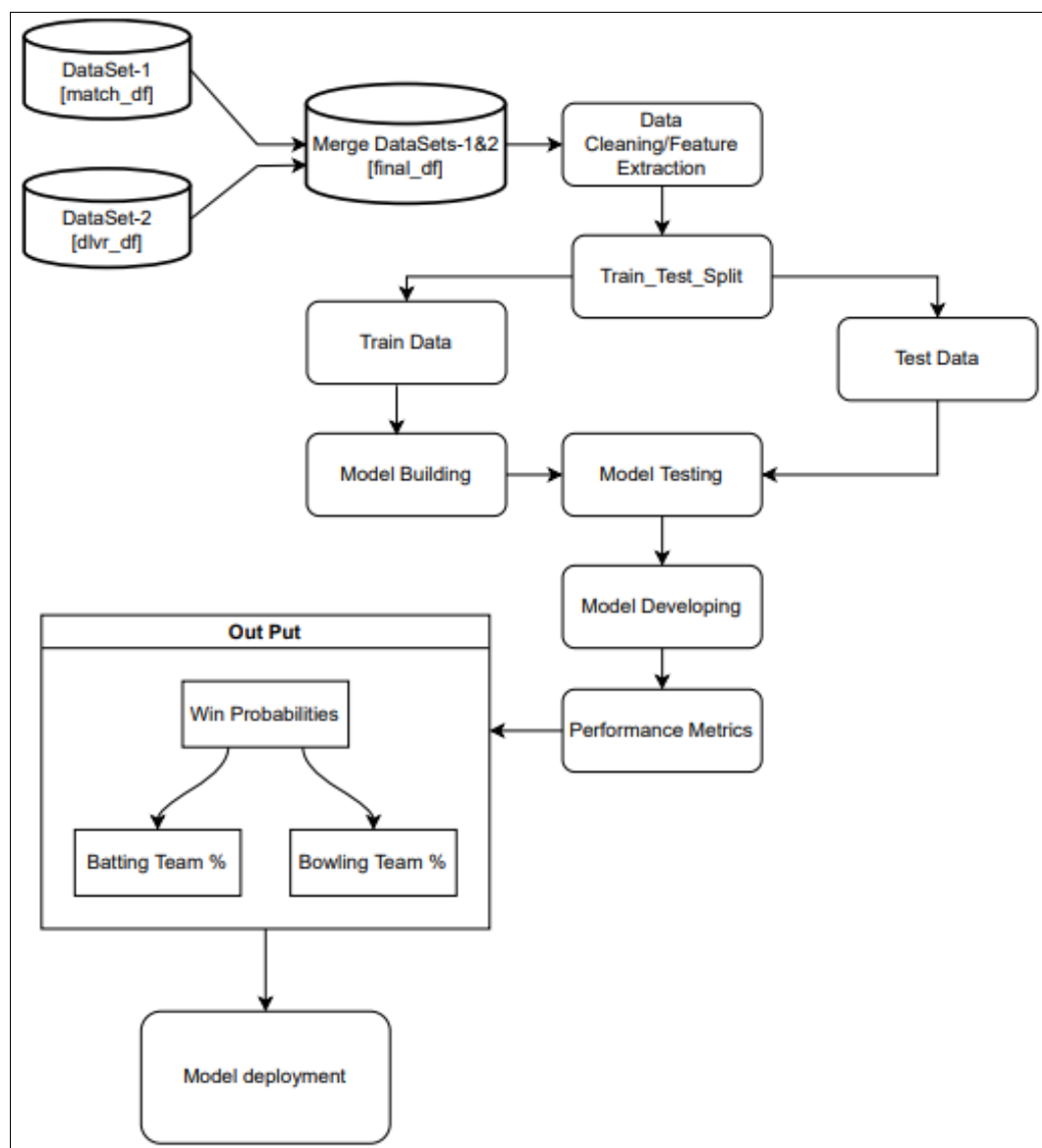


Fig 2: Proposed system architecture

#### 4.1.1 Datasets information

Two data sets are gathered for predictions in this study. They are deliveries.csv and matches.csv. Both of these data sets were gathered from kaggle.com. The matches.csv

dataset includes information on the previous 756 IPL matches. Each IPL match's results are also provided in csv format with an additional 18 attributes, as shown in Table 1.

**Table 1: Matches.csv**

Attributes of the dataset matches.csv					
1.	id	2.	Season	3.	city
4.	date	5.	team 1	6.	team2
7.	toss_winner	8.	toss decision	9.	result
10.	dl_applied	11.	winner	12.	win_by_runs
13.	win_by_wickets	14.	player of match	15.	venue
16.	umpire 1	17.	umpire 2	18.	umpire 3

The dataset deliveries.csv file depicts ball-by-ball information from both innings along with the 21 attributes listed in Table 2.

**Table 2: Deliveries.csv**

Attributes of the dataset deliveries.csv					
1.	match_id	2.	inning	3.	batting team
4.	bowling_team	5.	over	6.	ball
7.	batsman	8.	non_striker	9.	bowler
10.	is_super_over	11.	wide runs	12.	bye_runs
13.	legbye_runs	14.	noball_runs	15.	penalty_runs
16.	batsman_runs	17.	extra runs	18.	total_runs
19.	player_dismissed	20.	dismissal kind	21.	fielder

#### 4.1.2 Data merging

In the aforementioned architecture, data merging is one of the important stages. We need precise data to predict the IPL's win probability. Since no dataset provides the required attributes, there is a need to merge two datasets (i.e., matches.csv and deliveries.csv). match\_id, city, result, batting\_team, bowling\_team, inning, over, etc. Attributes are merged and formed as one data frame for predicting the IPL win probabilities. This data frame contains 72413 rows with 10 columns.

#### 4.1.3 Data cleaning/Feature extraction

Data cleaning is the process of eliminating or changing data that is inaccurate, lacking, and unnecessary, duplicated, or formatted incorrectly in order to prepare it for analysis. Feature selection or extraction enables one to pick the subset of the most important characteristics by removing the repetitive and irrelevant features from the initial data set. The variables in the dataset can only be used in a tiny section of the machine learning model. The remaining items are either unnecessary or unimportant. If all of this extraneous, meaningless data is included in the dataset, the model's overall accuracy and performance may deteriorate. It is essential to identify and pick the most suitable features from the data in order to remove the superfluous or less important information, which is performed with the help of feature selection in machine learning.

#### 4.1.4 Train\_Test\_Split

##### 4.1.4.1 Train

The data we use to train a machine learning algorithm or model is known as training data. To evaluate or prepare training data for machine learning, some human intervention is necessary. In this project, 80% of the data is used for training the ML model.

##### 4.1.4.2 Test

Following training on a particular dataset, we put the machine-learning model to the test. In this phase, we give our model a test dataset to see if it is accurate or not. According to the needs of the project or challenge, testing

the model determines its accuracy percentage. 20% of the data in this process is set aside for testing.

#### 4.1.5 Model building

Machine learning models are potent instruments used to successfully complete important jobs and resolve challenging issues. Building a machine learning model is frequently a difficult process that is overseen by data science experts. The fundamentals of creating a machine learning model are covered in this phase, which divides the procedure into six parts.

The following are the six steps to creating a machine-learning model:

1. Contextualise machine learning in your organization
2. Explore the data and choose the type of algorithm
3. Prepare and clean the dataset
4. Split the prepared dataset and perform cross-validation
5. Perform machine learning optimization
6. Deploy the model

#### 4.1.6 Model testing

In machine learning, the method of assessing a highly trained quality of the model on a testing set is known as model testing. The testing set should be kept separate from the training and validation while still having the same likelihood function. It consists of a number of testing samples. The target value is known for each testing sample. The trained model's performance can be tested by contrasting the predicted value with the actual value.

#### 4.1.7 Performance metrics

The effectiveness or caliber of the model is evaluated using a variety of measures, often known as performance metrics or evaluation metrics. These performance indicators allow us to assess how far our model processed the given data in the right way. The hyperparameters of the ML models can be tweaked to enhance the effectiveness of the model.

##### 4.1.7.1 Performance metrics for classification

###### 1. Accuracy

One of the simplest classification metrics to use is accuracy, which is calculated as the proportion of accurate predictions to all other predictions this will be depicted in "Eq. (5)".

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}) \quad \text{Eq. (5)}$$

###### 2. Precision

The precision metric overcomes the limitation of the accuracy metric. The precision determines the percentage of correct positive forecasts which is shown in "Eq. (6)".

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}) \quad \text{Eq. (6)}$$

###### 3. Recall

The recall is determined as the proportion of positively

classified positive samples to all positively classed samples which is displayed in “Eq. (7)”.

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$$

Eq. (7)

#### 4. F1-Score

Using equal weights for both precision and recall, the F1 Score can be derived as the harmonic mean of both variables and which is given by “Eq. (8)”.

$$\text{F1 score} = (2(\text{recall} * \text{precision})) / (\text{recall} + \text{precision}) \text{ Eq. (8)}$$

#### 5. Experimental Results

Before making the predictions the model was tested on the old match by selecting the match id. Fig 3 depicts those results.

##### 5.1 Prediction of old match probability

Match: Kings XI Punjab vs Delhi Capitals					
Target: 188					
Win probability for Kings XI Punjab					
end_of_over	runs_in_over	wickets_in_over	lose	win	
1	2	0	71.2	28.8	
2	4	1	82.1	17.9	
3	11	0	76.9	23.1	
4	9	1	82.9	17.1	
5	7	1	88.6	11.4	
6	7	0	87.4	12.6	
7	6	0	86.7	13.3	
8	3	0	87.7	12.3	
9	4	0	88.2	11.8	
10	11	1	90.9	9.1	
11	8	1	93.9	6.1	
12	9	0	92.7	7.3	
13	4	0	93.1	6.9	
14	5	1	96.1	3.9	
15	13	0	94.4	5.6	
16	9	0	93.6	6.4	
17	12	0	91.6	8.4	
18	6	0	92.9	7.1	
19	4	1	98.2	1.8	

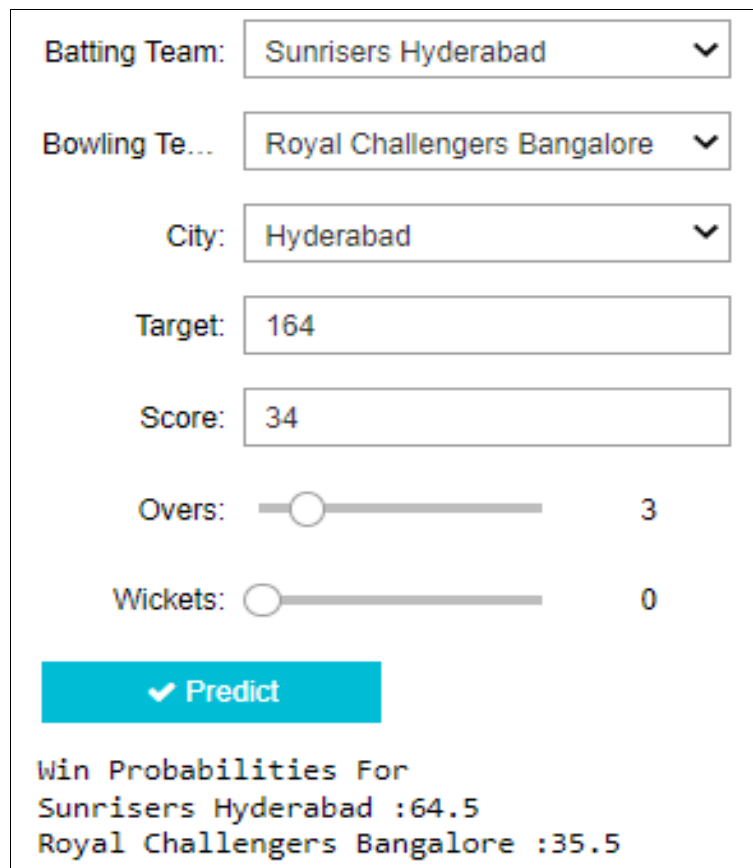
Fig 3: Old match probability predictions

Fig. 3 displays the win and loss probabilities of the batting team (Kings XI Punjab) from the first over to the 19th over. We can observe from the above results that the batting team scored 2 runs without wicket loss after the completion of the first over. At this stage, our model gives a 71.2% loss and 28.2% win probability for the batting team. After completion of the 10th over the batting team scored 64 runs with 4 wickets loss, at this stage our model gives a 90.9% loss and 9.1% win probability for the batting team. Therefore the predictions are changing over by over, hence

our model was tested successfully.

##### 5.2 Win probability prediction for live match

After testing the model on an older match, we applied it to the live IPL match. The outcomes were shown in below figures. Additionally, a Graphical User Interface (GUI) was created for simplifying the task of inputting the values to our models. The GUI accepts batting team, bowling team, city as drop-down menu, target, score as text boxes and overs, wickets as sliders.



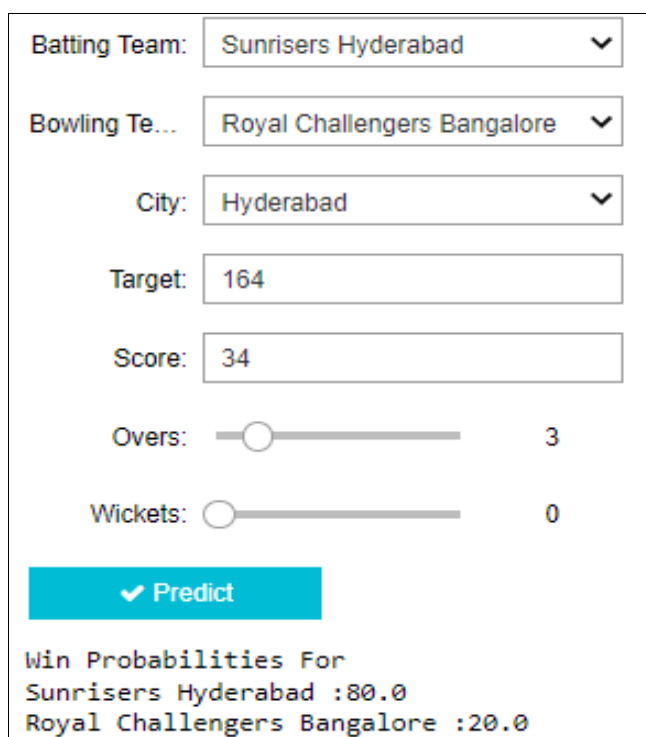
Batting Team: 
  
 Bowling Te...: 
  
 City: 
  
 Target: 
  
 Score: 
  
 Overs:  3
   
 Wickets:  0
   

  
 Win Probabilities For
   
 Sunrisers Hyderabad :64.5
   
 Royal Challengers Bangalore :35.5

**Fig 4:** Win Probability Prediction using Logistic Regression

Fig. 4 displays the win possibilities for Sunrisers Hyderabad and Royal Challengers Bangalore using Logistic Regression model based on the city, target, score, overs, and wickets. According to the results, Sunrisers Hyderabad does have a 64.5% probability of winning this match and Royal Challengers Bangalore does have a 35.5% probability of winning this match at the end of 3<sup>rd</sup> over.

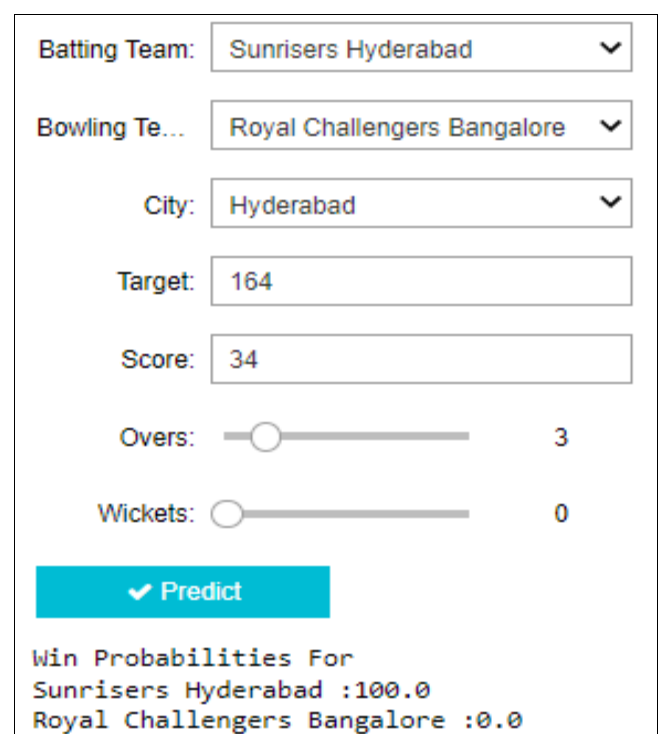
With the same inputs, Fig.5 displays the win possibilities for Sunrisers Hyderabad and Royal Challengers Bangalore using a Random Forest model. According to the results, Sunrisers Hyderabad does have an 80% probability of winning this match and Royal Challengers Bangalore does have a 20% probability of winning this match.



Batting Team: 
  
 Bowling Te...: 
  
 City: 
  
 Target: 
  
 Score: 
  
 Overs:  3
   
 Wickets:  0
   

  
 Win Probabilities For
   
 Sunrisers Hyderabad :80.0
   
 Royal Challengers Bangalore :20.0

**Fig 5:** Win Probability Prediction using Random Forest



Batting Team: 
  
 Bowling Te...: 
  
 City: 
  
 Target: 
  
 Score: 
  
 Overs:  3
   
 Wickets:  0
   

  
 Win Probabilities For
   
 Sunrisers Hyderabad :100.0
   
 Royal Challengers Bangalore :0.0

**Fig 6:** Win Probability Prediction Using Naïve Bayes



Fig.6 displays the win possibilities for Sunrisers Hyderabad and Royal Challengers Bangalore using a Naïve Bayes model based on the same inputs as those we used earlier. According to the results, Sunrisers Hyderabad does have a 100% probability of winning this match and Royal Challengers Bangalore does have a 0% probability of winning this match. These results are unrealistic because still, 17 overs are remaining but it gives a 0% win probability for the bowling team. Therefore, this model is not feasible for this project. The results show that Logistic Regression and Random Forest Algorithms are the effective models for calculating match win probabilities for our

project.

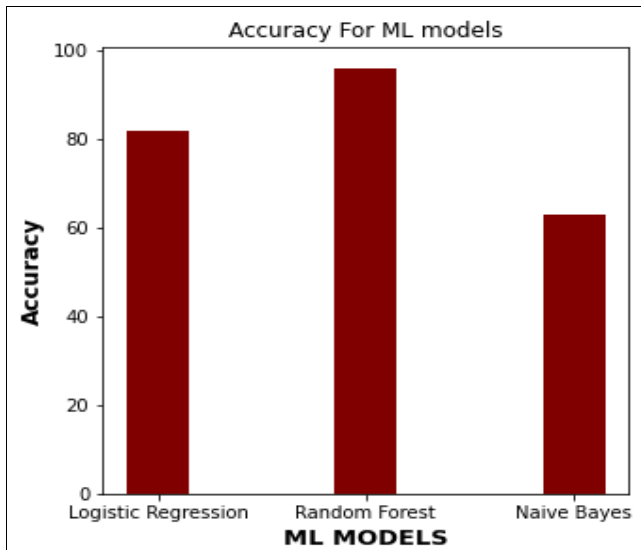
### 5.3. Performance metrics

The below table 3, provides the results of the performance metrics.

**Table 3:** Results of performance metrics

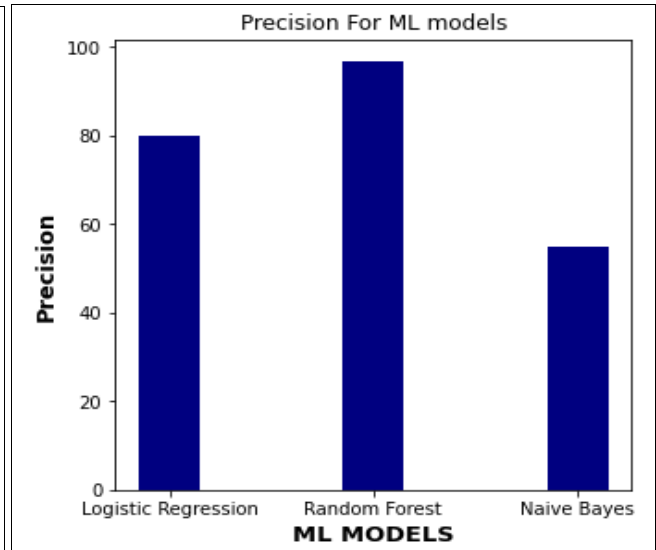
S. No.	Model	Accuracy	Precision	Recall	F1 score
1	Logistic Regression	82%	80%	79%	80%
2	Random Forest	96%	96.8%	96.5%	96.7%
3	Naïve Bayes	63%	55%	95%	70%

#### 5.3.1. Performance metrics comparison graphs



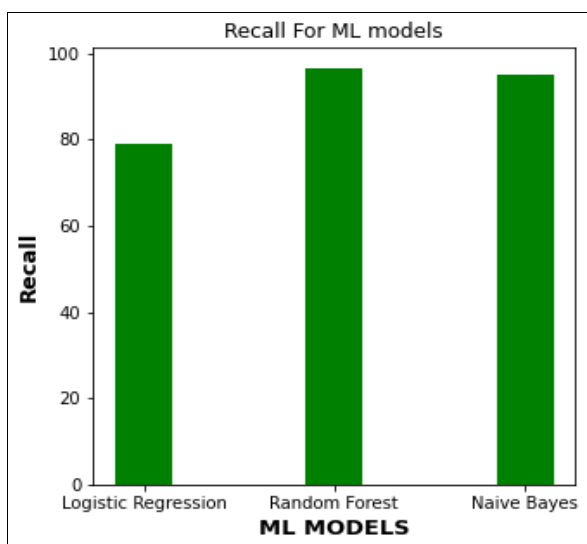
**Fig 7:** Accuracy Comparison Graph

Fig.7 shows the accuracy comparison graph between three machine learning models. As seen in the above visualization, NB, RF and LR gives 63%, 96% and 82% accuracies respectively. Fig.8 shows the precision



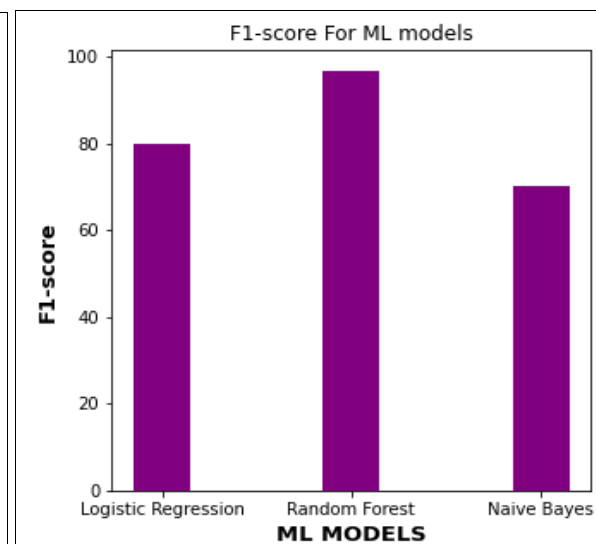
**Fig 8:** Precision Comparison Graph

comparison graph between three machine learning models. As seen in above visualization, NB, RF and LR gives 55%, 96.8% and 80% precisions respectively.



**Fig 9:** Recall Comparison Graph

Fig.9 shows the recall comparison graph between three machine learning models. As seen in the above graph, NB, RF and LR gives 95%, 96.5% and 79% recalls respectively. Fig.10 shows the F1-score comparison graph between three



**Fig 10:** F1-score Comparison Graph

machine learning models. As seen in the above graph, NB, RF and LR gives 70%, 96.7% and 80% F1-scores respectively.

## 6. Conclusion

Our major goal is not to help the gambling industry, which highlights the inherent risk in the game. However, we want to use our study to make some major findings. It can also be examined by looking at the results of IPL 2019 matches.

In this project, historical data from actual IPL cricket matches has been gathered, and after data pre-processing, relevant features have been extracted. Three classifiers LR, RF and NB were trained and used to categorize this data. Results have been obtained with an overall accuracy of 82% for the LR Classifier, 96% for the RF Classifier, and 63% for the NB Classifier. According to the experimental results, Logistic Regression and Random Forest Classifiers are the suitable models and give the best accuracies to predict the IPL match-win probabilities.

## 7. Future Scope

As our method accurately forecasts the IPL in the current scenario based on historical data, it may be expanded in a changing environment when numerous new talents join the team and their historical data becomes accessible.

Using this project, win probabilities for live IPL matches can be determined. Additionally, we can draw the comparison plot graphs for both batting and bowling team performances of the old and live matches. With precise statistics, this project can be extended to determine the win probabilities for matches played in the One-day, Test, and T-20 format cricket.

## References

1. Agrawal S, Singh SP, Sharma JK. Predicting Results of Indian Premier league T-0 Matches using Machine Learning, 2018 8<sup>th</sup> Int. Conf. Commun. Syst. Netw. Technol; c2008. p. 67-71.
2. Barot H, Kothari A, Bide P, Ahir B, Kankaria R. Analysis and Prediction for the Indian Premier League; c2020. p. 1-7.
3. Hodge VJ, Devlin S, Sephton N, Block F, Cowling PI, Drachen A. Win Prediction in Multiplayer Esports: Live Professional Match Prediction. 2021;13(4):368-379.
4. Akhtar S, Scarf P. Forecasting test cricket match outcomes in play, Int. J Forecast. 2012;28(3):632-643. DOI: 10.1016/j.ijforecast.2011.08.005.
5. Campus SK. Predicting the outcomes of tennis matches using a low-level point model; c2012-2013 Apr. p. 311-320. DOI: 10.1093/imaman/dps010.
6. Scarf P, Shi X. Modelling match outcomes and decision support for setting a final innings target in test cricket, IMA J Manag. Math. 2005;16(2):161-178. DOI: 10.1093/imaman/dpi010.
7. Hayter AJ. Win-probabilities for regression models, Stat. Methodol. 2012;9(5):520-527. DOI: 10.1016/j.stamet.2012.02.002.
8. Maher M. Predicting the outcome of the Ryder cup, IMA J Manag. Math. 2013;24(3):301-309. DOI: 10.1093/imaman/dps008.
9. Asif M, Mchale IG. In-play forecasting of win probability in One-Day International cricket: A dynamic logistic regression model, Int. J Forecast. 2016;32(1):34-43. DOI: 10.1016/j.ijforecast.2015.02.005.