Machine learning based IPL match win predictor

Devarsh Jadeja
Dept. of Computer Science and Engineering
DEPSTAR -CHARUSAT
Vadodara,Gujarat
devarshjadeja09@gmail.com

Udit Kamdar
Dept. of Computer Science and Engineering
DEPSTAR- CHARUSAT
Rajkot,India
uditkamdar@gmail.com

***Abstract*—In recent times, machine learning has become crucial providing us with insights using the information.To achieve our goal, we should gather information from a variety of reputable sources and create multiple models. This paper centers around the usage of Machine Learning in the field of crickets.Cricket is an uncertain sport, especially in the T-20 format, where the total gameplay can be altered by the impact of a single over, making it a constant challenge to devise a strategy for predicting match outcomes. Many people watch the Indian Premier League (IPL) on a regular basis. A cricket match's outcome is determined by a variety of factors. Cricket is an unsure game, especially in the T-20 format, chances are there that a single over can make a huge impact on the total gameplay. Millions of people watch the Indian Premier League (IPL) consistently, thus it turns into a constant issue to create a strategy that will gauge the finish of matches. Numerous viewpoints and highlights decide the consequence of a T20 cricket match.**

Keywords—Cricket,Indian Premier League,Machine Learning

I. Introduction

A large amount of money is on the line with respect to the outcome of IPL cricket matches in terms of match winnings, awards, and even external bets. Using match information gathered from all the seasons of the Indian Premier League, several variables were developed that could independently explain large proportions of uncertainty in predicted run totals and match outcomes. Such variables include home ground advantage, Toss wins, Runs remaining at two stages of the same game, balls remaining at those two stages along with wickets lost at those stages, and performance against specific opposition. These variables are individually weighed based on their significance and are key in predicting the outcome of the game. Studies suggest that, in most cases, the audience keeps getting increasingly engaged as the match progresses as it manages to keep everyone on the edge of their seats. A win predictor in cricket is a very important tool used by statisticians to help broadcasters engage the audience by closely predicting the outcome of a game at different stages as it is being played. This not only helps the audience but also the teams and players because they get to understand what their strengths and weaknesses are and overcome them using data.

The industry around cricket is a highly competitive one and is growing each day. However, win predictors have only recently shown up in different tournaments, so they are still pretty new, use different techniques, and aren't very precise. Hence, taking up this project with multiple variables helps us get one step closer to the current market standards and further updates of this can potentially help us build a product that can be used by various bodies in the industry. For the same, we have used a lot of different methods and models in order to find the most suitable and accurate one. We have worked mainly with Python for all the prediction models and have used Python as well as R for data visualization and analysis. The models incorporated to build the predictor system are Decision Tree / Random Forest Classification and XGBoost.

Since the need to accurately predict the outcome has piqued the attention of various people in the industry, the media, and the audience, it is absolutely important to create an intelligent and viable prediction model.

II. Objective

To predict the result of a match from any point of situation in the second innings on the basis of factors such as (balls remaining, runs required , wicket lost etc.) using the previous years' data.

III. About The Dataset

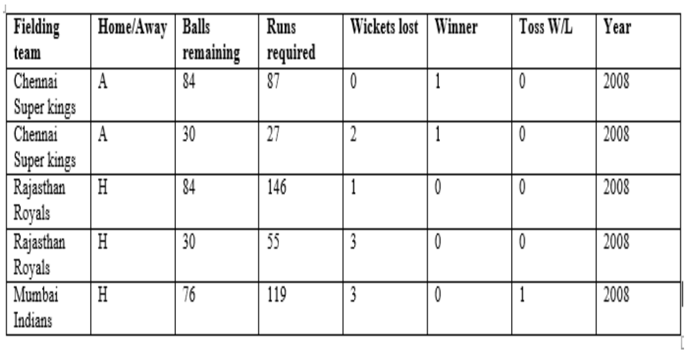
A. The Data

The sample of the full primary dataset that we generated for the research study is shown below. The original dataset was ball by ball data for every match throughout 12 seasons. We cleaned the data and turned it into something that could be used to produce the win or lose calculator.

We had to first measure the total first innings score set by the team batting first, and then divide the second innings into different periods to figure out how many balls were left, how many runs were needed, and how many wickets were lost. This allows the model to take in data when a team wins or loses based on the factors' values.

We also took into account whether the team batting second is playing on their home field or not, as this is a major factor in teams winning or losing, as well as the toss outcome. We believe that these variables provide us with sufficient data to reliably predict the outcome of a future match.From each season, we have carefully selected the matches to ensure that the number of data points for a particular team did not increase and that the number of data points for a particular fixture did not decrease. There would be no prejudice in this dataset since every team has at least two entries in both batting first and batting second.

We have excluded teams such as Deccan Chargers, Pune Warriors India, Gujrat Lions, and Rising Pune Supergiants from the dataset which are no more playing in IPL.



Fielding team	Home/Away	Balls remaining	Runs required	Wickets lost	Winner	Toss W/L	Year
Chennai Super kings	A	84	87	0	1	0	2008
Chennai Super kings	A	30	27	2	1	0	2008
Rajasthan Royals	H	84	146	1	0	0	2008
Rajasthan Royals	H	30	55	3	0	0	2008
Mumbai Indians	H	76	119	3	0	1	2008

Figure 1: Some records from the dataset

B. Predictor affecting variables

Input variables:

- Home/Away – If the team batting second is playing in their home ground or not
- Balls remaining – Number of deliveries left at that point in the game
- Runs required – Number of runs required for the team batting second at that point in the game
- Wickets lost – Number of wickets lost by the team which is batting second
- Toss W/L – If the team who is batting second won the toss or not

Outer Variables:

- Winner – If the team batting second won the game or not.

C. Dataset visualization

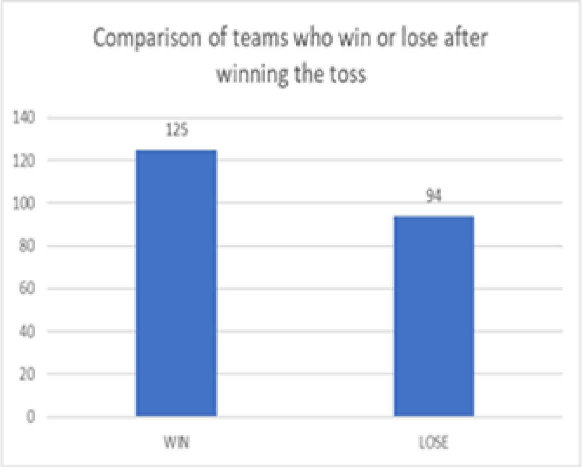


Figure 2: Comparison of teams who win or lose after winning the toss

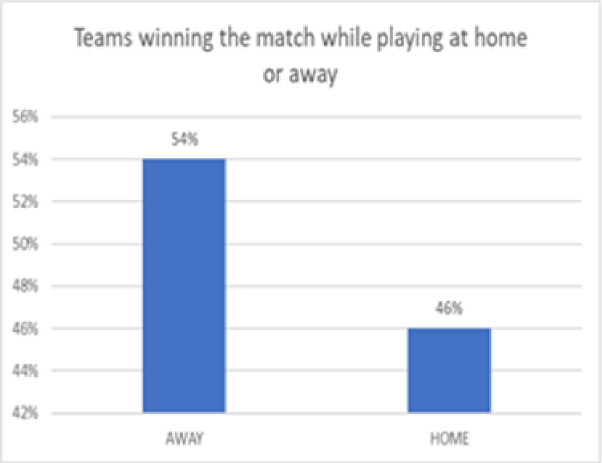


Figure 3: Comparison of teams winning the match while playing at homeground vs away

IV. Methodology and Classification

A. Random Forest Classifier

Random Forest is a supervised Machine Learning algorithm which can be used for both classification and regression. We use this Classifier to construct the model since the outcome of our predictive model must be either a win or a loss for the batting team. The Decision Tree Classification algorithm is usually extended with this approach.To get a decision tree in python, we partition the entire dataset into 2 sections: training and testing. Typically, the training information includes 80% of the total rows and the test information is made of the other 20%. Python at that point builds a tree on its own utilizing the training dataset contemplating all the independent variables dependent on the probability of a specific outcome because of that factor at each phase of the tree. This calculation is then utilized on the test dataset to anticipate the result and test the precision of the calculation. A Random Forest classifier considers various of these decision trees dependent on a random order of rows selected to build the algorithm and finally deduces an algorithm that has the most accuracy.

Following is the predictive model built by the Random Forest Classifier on our dataset. This model has an accuracy of 88.6 percent. This means that this model correctly forecasts approximately 88 out of 100 matches.

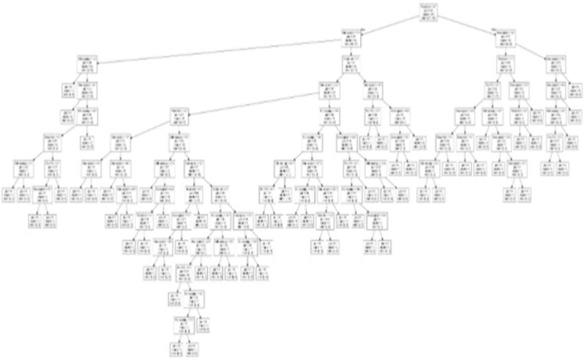


Figure 4 : Decision Tree

This model can be cross validated by using the K-fold Cross-validation where the model is again split into train and test datasets and then validated based on the results obtained from both. The cross-validation score for the Random Forest Classifier is 82% with SD = 0.04. This means that the predictive model works efficiently 82 on 100 times. The confusion matrix obtained for this algorithm is:

		Predicted	
		0	1
Actual	0	31	5
	1	5	31

Table 1: Confusion Matrix

B. XGBOOST

XGBoost stands for e**X**treme **G**radient **B**oosting which is a very efficient implementation of the gradient boosted decision tree algorithm designed for speed and performance. It is a supervised machine learning algorithm, which aims to accurately predict a target variable by combining the estimates of a collection of weaker models.

Though the decision tree algorithm forms a basis for XGBoost as well, this method is particularly more accurate as gradient boosting repetitively leverages the patterns in residuals, i.e., it strengthens the weak models and hence, lowers the overall error. The prediction made by the XGBoost model is thus expected to be much more accurate.

In the XGBoost algorithm also, the whole dataset is split into 2 parts: training and test data. The models are first built and tried on the training data, following which the best model is then selected to be applied to the test data.

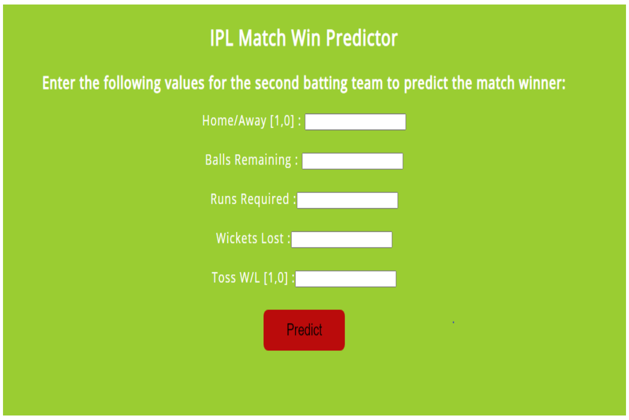
XGBoost, when applied to the dataset, produces an accuracy score of 82% whereas The K-fold cross-validation applied on the XGBoost model gives a cross-validation score of 81% with SD = 0.05.

	RFC	XGBoost
Accuracy	88.6	81.4

Table 2: Accuracy of RFC and XGBoost

V. Results

The following are the implementation screenshots of our web application which takes input from users such as balls remaining, wickets lost, who won the toss ?.. and predicts whether the match will be won by the second batting team or not.



IPL Match Win Predictor

Enter the following values for the second batting team to predict the match winner:

Home/Away [1,0] :

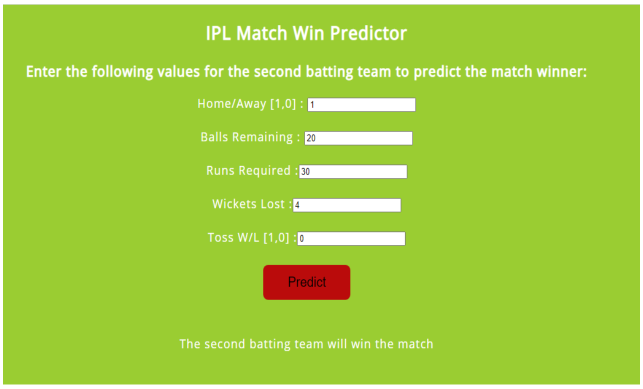
Balls Remaining :

Runs Required :

Wickets Lost :

Toss W/L [1,0] :

Figure 5: The Basic Web UI



IPL Match Win Predictor

Enter the following values for the second batting team to predict the match winner:

Home/Away [1,0] : 1

Balls Remaining : 20

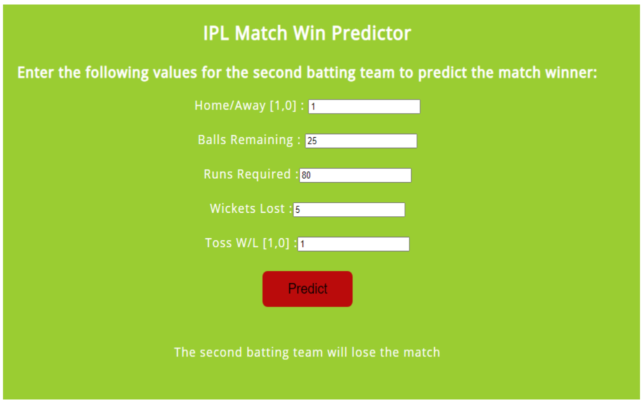
Runs Required : 30

Wickets Lost : 4

Toss W/L [1,0] : 0

The second batting team will win the match

Figure 6: Second batting team winning the match



IPL Match Win Predictor

Enter the following values for the second batting team to predict the match winner:

Home/Away [1,0] : 1

Balls Remaining : 25

Runs Required : 80

Wickets Lost : 5

Toss W/L [1,0] : 1

The second batting team will lose the match

Figure 7: Second batting team losing the match

VI. Conclusion

In this project , ***IPL match win predictor*** using RFC and XGBOOST machine learning algorithm has been implemented. Accuracy when using RFC(88.6%) is higher than XGBOOST(81.6%) which shows us that Random Forest Classification is more suitable for this type of predictions.We have embedded our machine learning model in web application to provide the graphic user interface for the user. The project was a great opportunity to learn about different ML algorithms. The importance of time bound and coordinated implementation of work was realized.

VII. References

[1] Passi, Kalpdrum & Pandey, Niravkumar. (2018) “Predicting Players' Performance in One Day International Cricket Matches Using Machine Learning” 111-126. 10.5121/csit.2018.80310.

[2] I. P. Wickramasinghe et. al, "Predicting the performance of batsmen in test cricket," Journal of Human Sport & Exercise”, vol. 9, no. 4, pp. 744-751, May 2014.

[3] R. P. Schumaker, O. K. Solieman and H. Chen, "Predictive Modeling for Sports and Gaming” in Sports Data Mining, vol. 26, Boston, Massachusetts: Springer, 2010.

[4] J. McCullagh, "Data Mining in Sport: A Neural Network Approach," International Journal of Sports Science and Engineering, vol. 4, no. 3, pp. 131-138, 2012.

[5] Bunker, Rory & Thabtah, Fadi. (2017) “A Machine Learning Framework for Sport Result Prediction. Applied Computing and Informatics”, 15. 10.1016/j.aci.2017.09.005.

[6] Ramon Diaz-Uriarte and Sara, “Gene selection and classification of microarray data using random forest, BMC Bioinformatics”, doi:10.1186/1471-2105-7-3.

[7] Rabindra Lamsal and Ayesha Choudhary, “Predicting Outcome of Indian Premier League (IPL) Matches Using Machine Learning”.

[8] Akhil Nimmagadda et. Al, “Cricket score and winning prediction using data mining”, IJARnD Vol.3, Issue3.

[9] Ujwal U J et. At, “Predictive Analysis of Sports Data using Google Prediction API” International Journal of Applied Engineering Research”, ISSN 0973-4562 Volume 13, Number 5 (2018) pp. 2814-2816.

[10] Rameshwari Lokhande and P.M.Chawan, “Live Cricket Score and Winning Prediction”, International Journal of Trend in Research and Development, Volume 5(1), ISSN: 2394-9333.

[11] Abhishek Naiket. Al, “Winning Prediction Analysis in One-Day-International (ODI) Cricket Using Machine Learning Techniques”, IJETCS, vol. 3, issue 2, ISSN:2455-9954, April 2018.

[12] Esha Goel and Er. Abhilasha, “ Random Forest: A Review”, IJARCSSE, Volume 7, Issue 1, DOI: 10.23956/ijarcsse/V7I1/01113, 2017.

[13] Amit Dhurandhar and Alin Dobra, “ Probabilistic Characterization of Random Decision Trees”, Journal of Machine Learning Research, 2008.

[14] Lamsal, Rabindra & Choudhary, Ayesha. (2018). Predicting Outcome of Indian Premier League (IPL) Matches Using Machine Learning.