Predicting Outcome of Indian Premier League (IPL) Matches Using Classification Based Machine Learning Algorithm

Rabindra Lamsal\textsuperscript{1} and Ayesha Choudhary\textsuperscript{2}

School of Computer \& Systems Sciences
Jawaharlal Nehru University, New Delhi, India
\textsuperscript{1}rabindralamsal@outlook.com, \textsuperscript{2}ayeshac@mail.jnu.ac.in

\textbf{ABSTRACT}

Cricket, especially the twenty20 format, has maximum uncertainty, where a single over can completely change the momentum of the game. With millions of people following the Indian Premier League, therefore developing a model for predicting the outcome of its matches beforehand is a real-world problem. A cricket match depends upon various factors, and in this work various feature selection methods were used to reduce the number of features to 5 from 15. Player’s performance in the field is considered to find out the overall weightage (relative strength) of the team. A Linear Regression based solution is proposed to calculate the weightage of a team based on the past performance of its players who have appeared most for the team. Finally, a dataset with the features: home team, away team, stadium, toss winner, toss decision, home-team-weightage and away-team-weightage, is fed to a Random Forest Classifier to train the model and make prediction on unseen matches. Classification results are satisfactory. Problem in the dataset and how the accuracy of the classifier can be improved is discussed.

\textbf{KEYWORDS}

Multiple Linear Regression, Random Forest Classifier, Scikit-learn

\section{1. INTRODUCTION}

Indian Premier League (IPL) is a professional cricket league based on twenty20 format and is governed by Board of Control for Cricket in India. The league happens every year with participating teams representing various cities of India. There are many countries active in organising t20 cricket leagues, and when most of them are being over hyped; team franchises routinely losing money, IPL has stood out as an exception \cite{7}. As reported by espncricinfo, with Star Sports spending $2.5 billion for exclusive broadcasting rights, the latest season of IPL (2018, 11\textsuperscript{th}) saw 29\% increment in number of viewers including both the digital streaming media \& television. The 10\textsuperscript{th} season had 130 million people streaming the league through their digital devices and 410 million people watching directly on television \cite{8}. So, with millions of people eying the league, it would be an interesting problem to make use of statistics and machine learning to predict the outcome of IPL matches.

As of now, there are eight teams that compete with one another in a double round-robin fashion during the league stage. After the league stage, the top four teams in the league points table qualify to the playoffs. In playoffs: The winner between first \& second team qualify for the final, while the loser gets an extra chance to qualify by playing against
the winner between third & fourth team that qualified for the playoffs. Finally, the two qualifying teams go against each other for the league title.

Among all formats of cricket, t20 format sees a lot of turnarounds in the momentum of the game. An over can completely change a game. So, predicting an outcome for a t20 game is quite a challenging task. On top of that, developing a prediction model for a league which is completely based on auction is another hurdle. Indian Premier League’s games cannot simply be predicted by just making use of statistics over participating team’s past games data solely. Because of players going under auction, players are bound to change their teams; which is why every player’s ongoing performance must also be taken into consideration.

In this work, we have analysed various factors that might affect the outcome of a cricket match, and found out that home team, venue, team strength, toss decision i.e. first batting or first fielding play respective vital role in determining the winning side. The proposed prediction model makes use of Linear Regression to find out the average strength of each team; while the final classification, whether the home team is likely to win the match, is based on Random Forest Classifier.

Some works have been published in this area of predicting outcome in sports. In our literature review, we found out that the published works were for test or one-day-international (ODI) cricket format. There were none, who have considered their prediction model for auction based cricket league, and also haven’t modelled player’s performance to calculate team’s strength based on their previous matches.

Bandulasiri [1] has analysed the factors like home field advantage, winning the toss, game plan (first batting or first fielding) and the effect of D/L (Duckworth Lewis) method [2] for one day cricket format. There were none, who have considered their prediction model for auction based cricket league, and also haven’t modelled player’s performance to calculate team’s strength based on their previous matches.

The rest part of the paper is as follows. Section 2 is about designing the dataset required for the model. In section 3 we discuss the use of machine learning algorithms in developing the model. Results are discussed in section 4. Finally, section 5 concludes the overall work.

2. Designing the Dataset

The main source of past matches data is official website of Indian Premier League [6]. The data is scrapped from the site and maintained in a Comma Separated Values (CSV) format.

2.1. Features Selection

The initial dataset after extracting various data from [6] had many features including: date, season, home team, away team, toss winner, man of the match, venue, umpires, referee, home team score, away team score, powerplay score, overs details when team
reached milestone of multiple of 50 (i.e. 50 runs, 100 runs, 150 runs), playing 11 players, winner, won by details.

Trying to feed all these features into the model is not completely necessary. Features selection methods is implemented to select out the only features which can help generate better results. We use scikit-learn machine learning library [10] to pre-process the extracted data, and applied three feature selection models.

1. Removed Low Variance Features:

   Sklearn.feature_selection.VarianceThreshold feature selector is used to remove the features that has low variance. This selector is effective in removing zero-variance features.

2. Univariate Feature Selection:

   Scikit-learn provides SelectKBest class to help us in removing all but k highest scoring features. The k needs to be specified beforehand.

3. Recursive Feature Elimination:

   Sklearn.feature_selection.RFE facilitates in selecting best features by recursively removing features and building a model on the remaining features. Least important features are removed until the desired k numbers of features are reached.

Using these feature selection models, we are able to reduce the number of features from initially 15 to 5. The selected five features are: Home Team, Away Team, Toss Winner, Toss Decision, and Stadium

2.2. Organizing the Dataset

In a single season, a team has to play with other teams in two occasions i.e. once as a home team and next time as an away team. For example, once KKR plays with CSK in its home stadium i.e. Eden Gardens, next time they have to play against CSK in CSK’s home stadium i.e. M Chinnaswamy Stadium. So, while making the dataset the concept of home team and away team is being considered and match data are placed in their specific columns accordingly to prevent the redundancy. In Table 1, every column specifies its own meaning.

- 1\textsuperscript{st} column represents \textit{Home Team} (team which is playing in their own ground)
- 2\textsuperscript{nd} column represents \textit{Away Team} (team which is playing in another team’s ground)
- 3\textsuperscript{rd} column represents \textit{Toss Winner} (a coin toss is done in presence of both the team’s captains, and one of them wins the toss)
- 4\textsuperscript{th} column represents \textit{Toss Decision} (toss winner gets to either first go with batting or fielding)
- 5\textsuperscript{th} column represents \textit{Stadium} (venue)
- 6\textsuperscript{th} column represents \textit{whether home team wins or not}. 1 means yes, 0 means no. (the team which scores the most runs wins the match)
Table 1. Listing 7 past matches data played by Kolkata Knight Riders.

| Col1     | Col2  | Col3  | Col4 | Col5           | Col6 |
|----------|-------|-------|------|---------------|------|
| KKR      | CSK   | KKR   | bat  | Eden Gardens  | CSK  |
| KKR      | DD    | DD    | bat  | Eden Gardens  | KKR  |
| KKR      | KXIP  | KXIP  | bat  | Eden Gardens  | KKR  |
| KKR      | MI    | KKR   | bat  | Eden Gardens  | MI   |
| KKR      | RR    | RR    | field| Eden Gardens  | RR   |
| KKR      | RCB   | KKR   | bat  | Eden Gardens  | KKR  |

For better understanding and to make the dataset look somehow cluttered-free, acronym is used for every team instead of their complete name. The acronyms used in the dataset are the official ones.

Table 2. Acronym used in the dataset. Teams in grey shaded area are not active at present.

| Team Name                  | Acronym |
|----------------------------|---------|
| Chennai Super Kings        | CSK     |
| Delhi Daredevils           | DD      |
| Kings XI Punjab            | KXIP    |
| Kolkata Knight Riders      | KKR     |
| Mumbai Indians             | MI      |
| Rajasthan Royals           | RR      |
| Royal Challenger Bangalore | RCB     |
| Sunrisers Hyderabad        | SRH     |
| Rising Pune Supergiant     | RPS     |
| Deccan Chargers            | DC      |
| Pune Warriors India        | PWI     |
| Gujrat Lions               | GL      |
| Kochi Tuskers Kerala       | KTK     |

Until now, we just have 5 features in our dataset. Later in section 4, we’ll discuss in detail how we make use of Multiple Linear Regression model to calculate home-team’s and away-team’s relative strength, using each player’s past performance. Finally, at the end, our classification-based machine learning algorithm will be fed with a dataset of 7 features for training purpose.

2.3. Problem with the Dataset

Indian Premier League is just 11 years old, because of which only 634 match data became available after pre-processing phase. This number is quite less with comparison to the data available for prediction work in ODI or test cricket.

Due to certain issues with the team franchises, in some seasons there have been participation of new teams, which are inactive as of now. Also, some teams have discontinued. Presences of these inactive teams in the dataset are not significant, but if those match details are removed where these teams appear, we would be missing important information about the teams which are still active in the league.
3. USE OF MACHINE LEARNING MODELS

3.1. Calculating Player’s Points

There are various ways a player can be awarded extra points for their performance in the field. The official website of IPL has a Player Points section where every player is awarded points based on these 6 features: (1) number of wickets taken, (2) number of dot balls given, (3) number of fours, (4) number of sixes, (5) number of catches & (6) number of stumpings.

To find out how IPL is assigning points to each player based on those 6 features, we use Multiple Linear Regression to the available dataset of Player Points. Freedman [9] has beautifully explained the mathematics behind the Linear Regression models.

Let us consider a dataset \{y_i, x_{i1}, x_{i2}, x_{i3} ..., x_{im}\} where \(i = 1\) to \(n\), with \(n\) other statistical units. In Linear Regression the relation between dependent variable i.e. \(y_i\) and independent variables \{\(x_{i1}, x_{i2}, x_{i3} ..., x_{im}\)\} is considered linear [9].

For our problem with 6 independent variables, multiple Linear Regression model takes the following form:

\[ y = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \beta_4 x_{i4} + \beta_5 x_{i5} + \beta_6 x_{i6} \]  

Where, \(y\): points awarded to a player, \(\beta_0\): Intercept term, \(\beta_1\): number of wickets taken, \(\beta_2\): number of dot balls given, \(\beta_3\): number of fours, \(\beta_4\): number of sixes, \(\beta_5\): number of catches, \(\beta_6\): number of stumpings.

When Multiple Variable Linear Regression was running on the dataset of Player Points, we obtained following relation:

\(\beta_0 = 0, \beta_1 = 3.5, \beta_2 = 1, \beta_3 = 2.5, \beta_4 = 3.5, \beta_5 = 2.5, \beta_6 = 2.5\)

Finally, for calculating each player’s performance points we have the working formula:

\[ \text{Player points} = (3.5 \times \text{number of wickets}) + (1 \times \text{number of dot balls given}) + (2.5 \times \text{number of fours}) + (3.5 \times \text{number of sixes}) + (2.5 \times \text{number of catches}) + (2.5 \times \text{number of stumpings}) \]

3.2. Calculating a Team’s Weightage

For a team there can be as many as 25 players. This is a limit put on by IPL governing council to the franchises. To find the average relative strength of a team, every players of the team should be sorted in the descending order by their number of appearances in previous matches in the same season. Once the team players have been sorted, the top 11 players are considered for calculating the weightage of the team because these players have played more games for the team and their individual performance adds to give us the team strength.

Now team’s weightage is calculated using:

\[ \frac{\sum_{i=1}^{11} (\text{6th player’s points})}{\text{Team’s total appearance till the moment}} \]  

Now two more features: home-team-weightage & away-team-weightage are added to the previously designed dataset for all match data in the dataset, formula (2) is used recursively to calculate the respective playing team’s weightage. Calculating team
weightage for all 634 matches for the dataset is a tedious task. So, for simulation, we’ve considered final results of each season and calculated the team weightage for each team accordingly and used the same score for all the matches in that particular season.

For better working of the classifier, the team weightage should be calculated after the end of each match. This way we can obtain the real-time performance of each team, and the newly calculated weightage can be used in predicting upcoming games.

3.3. Use of Dummy Variable

There are some categorical variables in our dataset. So, whenever there is absence of numeric value we should convert the categorical variables to dummy variables. Suits [11] have introduced Dummy variables as a simple way of introducing information contained in a variable which is not measures in continuous fashion; e.g. race, sex, religion etc. For broader view on Dummy Variables visit [11].

Here’s a quick example how categorical variable is converted to dummy variables.

Table 3. Example of conversion of categorical variable to dummy variables.

| Home Team | KKR | CSK |
|-----------|-----|-----|
| KKR       | 1   | 0   |
| CSK       | 0   | 1   |

Initially, there was one column i.e. one categorical variable Home Team. When it is converted to dummy variable two columns are generated i.e. KKR and CSK. Instead of previous Home Team column, the newly formed columns KKR & CSK are now used.

There are some constraints to be considered while using dummy variables. We should not include all the dummy variable columns in our dataset [11]. Any one column should be removed from a set of dummy variables to prevent falling into dummy variable trap.

If there are many columns that take on categorical variables in our dataset, then for each column we should consider converting it to dummy variables, and use the newly created set of dummy variables as a representative of previous categorical variable.

3.4. Using Random Forest Classifier for Classification

Random Decision Forests [12] are method of ensemble learning [13] which constitutes the concept of building N number of decision trees using random k datapoints from a training set of D, and whenever a new data point arrives each of these decision trees are used to classify it and is classified to that class which has the majority vote.

The N number of trees help in getting rid of biasness and reducing the uncertainty in the algorithm. In Empirical comparison done by Caruana & Niculescu-Mizil [14] between various supervised algorithms, the Random Forests came on second place just after boosted trees.
During formation of decision trees, we look for best split through use of information gain or Gini index [15].

3.5. Feeding Dataset to the Classifier

Our dataset final dataset (where categorical features were converted to dummy variables) is fed to this classifier. The dataset contains all the match data since the beginning of Indian Premier League till 2017. This will be the training data for the algorithm.

Study carried out by Kohavi [16] indicates that for model selection (selecting a good classifier from a set of classifiers) the best method is 10-fold stratified Cross Validation (CV). Scikit-learn library provides us StratifiedKFold class which helps with designing dataset that contains approximately the same percentage of samples of each class. [10]

4. RESULTS

The trained classifier correctly predicted 38 IPL 2018 matches out of 59. The confusion matrix (M) of results obtained from the classifier, as shown in Fig. 2, when interpreted deduces following information:

---Training dataset---

| 1st iteration | test |
|---------------|------|
| 2nd iteration | test |
| 3rd iteration | test |
| 4th iteration | test |
| . . . Training data . . . |
| test | 10th iteration |

Figure 1. An example of 10-fold CV.

This algorithm, splits the whole dataset into k=10 equal partitions (folds), and uses 1 fold as testing set and union of other folds as training set. The creation of folds is random. This process repeats for every fold. That means each fold will be a testing set for once. Finally, the average accuracy can be calculated out of the sample accuracy from each iteration.

\[
\text{Random Forests Algorithm}
\]

Let us consider \((x_1, y_1), \ldots, (x_n, y_n)\) be data points D.

for i = 1 to N
   Choose random sample \(D_k\) from D
   Use \(D_k\) to construct decision tree \(T_i\) such that we consider splitting on those features that have best split
Use all \(T_i\) to classify a new data point
Take majority vote for classification

During formation of decision trees, we look for best split through use of information gain or Gini index [15].
Figure 2. Confusion matrix

- Element $M_{1*1}$ shows that 6 times the classifier has correctly predicted the case away-team-winning.
- Element $M_{2*2}$ shows that 32 times the classifier has correctly predicted the case home-team-winning.
- Element $M_{2*1}$ shows that 4 times the classifier has incorrectly predicted the case home-team-winning.
- Element $M_{1*2}$ shows that 17 times the classifier has incorrectly predicted the case away-team-winning.

5. CONCLUSION

In this work, we’ve made use of two machine learning algorithms, Multiple Linear Regression and Random Forests respectively, in order to predict the outcome of Indian premier League match. We used the dataset containing 634 match data prior season 11. These number of matches details were obtained after putting the dataset through cleaning and pre-processing.

Multiple Linear Regression was used to calculate points for each player based on their performance in the field, and the points of those players who have appeared the most for a team were used in calculating the relative weightage (relative strength) of the team. For classification purpose, we used Random Forest Classifier and got a satisfactory result of correctly classifying 38 IPL 2018 matches correctly out of 59 total matches. The accuracy of the classifier would have improved further, if the team weightage was calculated immediately after a match ends.

ACKNOWLEDGEMENTS

The authors would like to thank Intel AI DevCloud for providing a computing cluster.

REFERENCES

[1] A. Bandulasiri, “Predicting the Winner in One Day International Cricket,” *Journal of Mathematical Sciences & Mathematics Education*, Vol. 3 (1).

[2] F.C. Duckworth, A.J. Lewis, “A fair method for resetting the target in interrupted one-day cricket matches,” *Journal of the Operational Research Society*, Vol. 49 (3), pp. 220-227, 1998.

[3] M. Bailev, S.R. Clarke, “Predicting the Match Outcome in One Day International Cricket Matches while the Game is in Progress,” *Journal of Sports Science & Medicine*, Vol. 5 (4), pp. 480–487, 2006.

[4] V.V. Sankaranaravanan, J. Sattar, L.V. Lakshmanan, “Auto-play: A Data Mining Approach to ODI Cricket Simulation and Prediction,” *SDM*, 2014.
A. Kaluarachchi and S.V. Aparna, "CricAI: A classification based tool to predict the outcome in ODI cricket," *2010 Fifth International Conference on Information and Automation for Sustainability*, Colombo, 2010, pp. 250-255.

[6] IPLT20, Official website of Indian Premier League, [online] https://www.iplt20.com

[7] T. Wigmore, "How can the IPL become a global sports giant?," 28 June 2018. [Online]. Available: http://www.espncricinfo.com/story/_/id/23931646/how-ipl-become-global-sports-giant.

[8] V. Choudhary, "Star India eyes 700 million viewers during IPL 2018," 20 December 2017. [Online]. Available: https://www.livemint.com/Companies/BrvDru6CWopjSh7guUH0N/Star-India-eyes-700-million-viewers-during-IPL-2018.html.

[9] D.A. Freedman, Statistical Models: Theory and Practice, Cambridge University Press, 2009.

[10] F. Pedregosa et al., “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[11] D.B. Suits, "Use of Dummy Variables in Regression Equations," *Journal of the American Statistical Association*, vol. 52 (280), pp. 548-551, 1957.

[12] T.K. Ho, "Random Decision Forests," in *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, Montreal, QC, 1995.

[13] R. M. Daavid Opitz, "Popular ensemble methods: An empirical study," *Journal of Artificial Intelligence Research*, vol. 11, pp. 169–198, 1999.

[14] R. Caruana, A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *Proceedings of the 23rd International Conference on Machine Learning (ICML 2006)*, Pittsburgh, PA, USA, New York, 2006.

[15] S.K. Tan, "Data Mining Classification: Basic Concepts, Decision Trees, and Model Evaluation," in *Introduction to Data Mining*, p. Chapter 4.

[16] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," in *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, San Mateo, 1995.

Authors

**Rabindra Lamsal** is a research scholar, pursuing his M.Tech in Computer Science and Technology degree at School of Computer & System Sciences, Jawaharlal Nehru University. His major field of interest are Machine Learning and Natural Language Processing.

**Ayesha Choudhary** is an assistant professor at School of Computer & System Sciences, Jawaharlal Nehru University. She did her PhD from Indian Institute of Technology, New Delhi. Her areas of specialization are Computer Vision, Machine Learning, Digital Image Processing, Linear Algebra, and Optimization.