Predicting the Result of a Cricket Match by Applying Data Mining Techniques

Fahim Ahmed Shakil(✉), Abu Hasnat Abdullah, Sifat Momen, and Nabeel Mohammed

Department of Electrical and Computer Engineering, North South University, Plot-15, Block-B, Bashundhara, Dhaka, Bangladesh
{fahim.shakil,hasnat.abdullah,sifat.momen,nabeel.mohammed}@northsouth.edu

Abstract. Many different factors influence the outcome of a cricket match. This paper presents the prospect of applying machine learning algorithms in predicting the outcome of an international cricket match. This study has two goals - identifying influential features that affects the outcome of a cricket match and predicting the outcome of a cricket match using machine learning algorithms. To achieve the objectives, feature selection algorithm Recursive Feature Elimination and machine learning algorithms - ZeroR, Decision Tree, Random Forest and XGBoost have been applied. To evaluate the performance of the models, dataset containing international ODI and T20 cricket matches from 2004–2018 has been used. An accuracy of 85.48% has been achieved by applying XGBoost algorithm on the test dataset.

Keywords: Classification · Data mining · Features selection · Cricket · XGBoost

1 Introduction

Cricket is one of the most popular sports in the world with most of the followers coming from the United Kingdom and Commonwealth nations. The recently held ODI cricket world cup in England and Wales in 2019 had reached an astounding 1.6 billion average viewers in live coverage [6]. There are three different types of formats according to which international cricket is played - Test, ODI and T20. Even though Test is the oldest format of this sport, the shorter versions of the game - T20 and ODI, are more beloved among the fans [8]. International One Day Cricket (ODI) has been played since 1971 and T20 since 2004. Even though few rules have been changed or introduced over time, the general rules of the game remain the same. Cricket is a team game where two teams compete with each other having eleven members each. The result of a cricket match may depend on various factors including team members, venue, pitch condition and so on. Team selection is one of the most important factors that influences the outcome of a cricket match and selecting the best players may not always result in the best combination. The best team combination may depend on many other

R. Silhavy et al. (Eds.): CoMeSySo 2020, AISC 1295, pp. 758–770, 2020.
https://doi.org/10.1007/978-3-030-63319-6_70

aspects such as the opponent's strength and weakness, the venue of the game, the weather condition, etc. It is difficult to determine the best combination for a team as the outcome of a cricket match is influenced by plenty of factors. To solve this problem machine learning algorithms can be applied to determine the outcome of a cricket match which then can be used to determine the best combination of players for team selection. Cricket boards can also use the machine learning algorithms to determine the most suitable venues and pitch conditions to ensure victory for their teams.

In this paper, we have used machine learning algorithms to predict the winner of international T20 and ODI cricket matches. We have used several machine learning techniques to get higher accuracy. There has been very little research conducted on predicting international cricket match results. Among them, most of the work has been done on ODI matches [2] and some used statistical calculation to predict the outcome [19]. But we have worked on T20 as well as ODI matches using machine learning algorithms to determine the most accurate results. We have collected and selected more interesting and effective attributes like average rating of players in a team, team rank, home team, etc. which helps our model to minimize the errors and produce more accurate results.

One of the important outcomes of this research work is that it facilitates, through the development of predictive models, the managers and coaches of different teams to plan and select team members that will favor winning a particular match.

The paper is outlined as follows: In Sect. 1, the problems, objectives and research methodologies that have been followed are presented. Section 2 reviews related articles on predicting the outcome of a cricket match using machine learning models. Section 3 discusses the methodology of the research and data and features used. Section 4 analyses the experiments and results. In Sect. 5 conclusion and future work are discussed.

2 Literature Review

Tejinder, Vishal and other colleagues worked on the prediction of score and result of a cricket match where they obtained 70% to 91% accuracy as the match progress. They have used two separate models for first and second innings respectively Linear Regression and Naïve Bayes classifier though they interestingly concluded that the game like cricket is highly unpredictable [19].

Multiple linear regression can be a fantastic method to predict probabilities of the competing teams in ODI matches showed by Bailey and Clarke where they gathered information of 2200 ODI matches played prior to January 2005 [2].

Muthuswamy and Lam creatively used a neural network approach using Backpropagation Network (BPN) and Radial Basis Function Network (RBFN) to predict Indian cricket team bowlers performance for ODI matches [14].

For Indian premier league (IPL), Singh and Kaur have predicted the outcomes up to 71% accuracy using supervised KNN with $k = 4$ where non-relational database HBase was used for scalability purpose [18].

Iyer and Sharda predicted the athlete's performance using neural networks. They have tried to obtain the batsman's and bowler's performances. Here Neural networks were able to identify 87% of actual performance before the world cup [9]. For One Day International matches Passi and Pandey forecasted the success of players by analyzing their characteristics and stats using supervised machine learning techniques. For this, they have predicted the performance of batsmen and bowlers separately as how many runs a batsman would score and how many wickets a bowler would take in a given match [15].

Another model was created by Sankaranarayanan and other colleagues using a combination of linear regression and nearest neighbor clustering algorithms. They showed the success of their algorithms in predicting the number of runs scored and the determinants of match results. They got the highest 70% accuracy according to their paper [17].

Haseeb et al. worked on predicting the top ten list of rising cricketers based on a weighted average, performance patterns and growing star ratings by applying BN, NB, CART and SVM [1].

The most interesting paper regarding the issue was written by Mustafa and other colleagues where they have collected crowd opinions from social networks and surprisingly got 75% correct prediction with the power of Support Vector Machine (SVM) over other classifiers (Naïve Bayes and Linear Regression) [13]. Munir et al. tried to predict T20 cricket match result during on going match and at the end of the game they attained 85.5% accuracy with Multiple Linear Regression algorithm [12].

3 Methodology

The goal of this paper is to build a model which can be used to predict the outcome of a cricket match. In this section, the data collection process, data pre-processing, selected features and applied machine learning models are described elaborately.

Figure 1 describes the complete workflow of the task. The raw dataset of international cricket matches has been collected from Cricsheet [16] where information about each single match is presented in separate YAML files. Then a csv file has been created containing information about all matches together. Team ranking and average of player rating have been scrapped from Reliance ICC Ranking [7]. After applying various data pre-processing methods, machine learning algorithms have been applied to generate classification models.

3.1 Data Collecting and Pre-processing

The initial raw dataset has been collected from Cricsheet [16]. The raw dataset contains information about international ODI and T20 cricket matches from 2004 to 2018. In the raw dataset, each single match is represented in a single YAML file which contains detailed information about a cricket match such as teams participating in the match, venue, city, date of the match, type of the match,

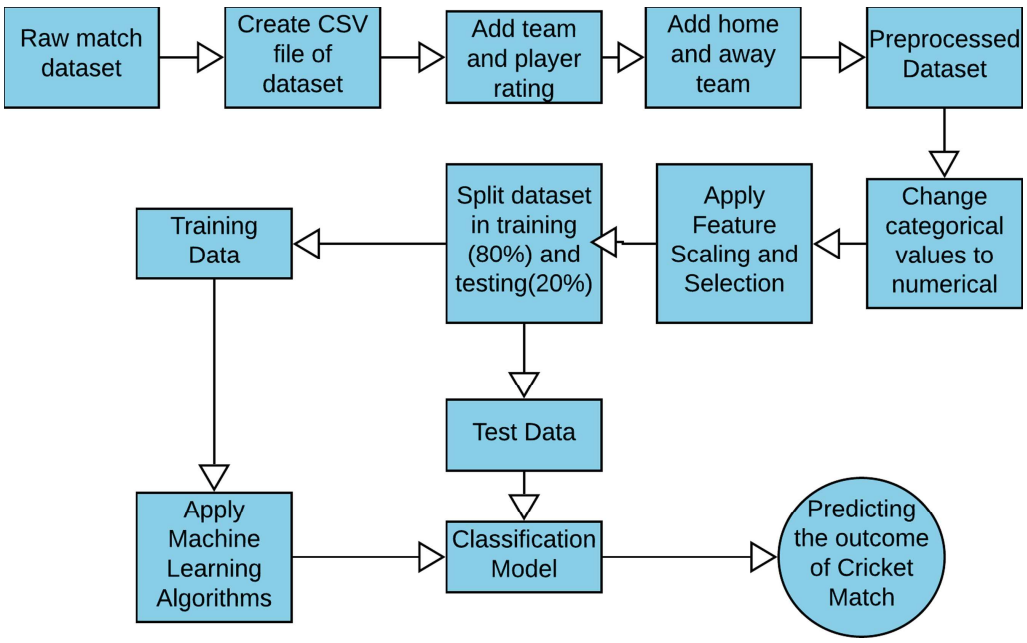

Fig. 1. Proposed framework for predicting outcome of a cricket match

toss winner, umpires, ball by ball information for both batting and bowling, tournament type, margin of winning and outcome of the match. To construct a complete dataset that contains useful information about all cricket matches in a single csv file, firstly a R package - Yorkr [4], has been used to convert YAML files to Rdata files and then convert the Rdata files to csv files. After creating single csv files representing each single match, Python Pandas packages has been used to create single csv file containing information of all the matches which contains - teams participating in the match, venue, date, home and away team, batting first and batting second team, match type and winner of the match. Using the "date" attribute, a new attribute has been generated called "season". Then to add team rating and player average rating, web scrapper has been used in Wikipedia International Cricket Season pages [20] and Reliance ICC Player Ranking website [7] respectively.

Team ranking and rating point has been collected from Wikipedia [20] according to the defined season and match type from the previously created csv file. The team ranking and rating points in the Wikipedia site has been accumulated from archived ICC website [5]. To collect average rating point of batsmen for a team in a single match, firstly the name of all players that batted for the team in that match has been collected from the previously created single match csv file. Secondly, the rating points of each of these players has been scraped from Reliance ICC Ranking [7] for the specified date from the "Date" attribute of the csv file. Finally, the total rating points of all the players that batted has been divided by the number of players that batted to determine the average batsmen rating for the team. Similar approach has been used to figure out average rating for bowlers.

$$\text{Avg. batsmen rating} = \frac{\begin{array}{c}\text{Total rating of players of} \\ \text{the team that batted}\end{array}}{\begin{array}{c}\text{Number of players} \\ \text{that batted in the innings}\end{array}} \tag{1}$$

$$\text{Avg. bowlers rating} = \frac{\begin{array}{c}\text{Total rating of players of} \\ \text{the team that bowled}\end{array}}{\begin{array}{c}\text{Number of players} \\ \text{that bowled in the innings}\end{array}} \tag{2}$$

Equations (1) and (2) have been used to calculate batting and bowling average of each team in each innings of a match.

After completing the pre-processing, the dataset have features described in Table 1.

Table 1. Description of features in the dataset

Attribute name	Description
TeamA	Name of one participating team
TeamB	Name of other participating team
Venue	Stadium name where the match was held
Season	Cricket season in which the match was played
TeamA Rank	Ranking of TeamA in given season
TeamA Rating	Rating of TeamA in given season
TeamB Rank	Ranking of TeamB in given season
TeamB Rating	Rating of TeamB in given season
Home Team	Name of home team from TeamA and TeamB or neutral
Away Team	Name of away team from TeamA and TeamB or neutral
TeamA Batting Average Rating	Average rating of batsmen of TeamA that batted
TeamA Bowling Average Rating	Average rating of bowlers of TeamA that bowled
TeamB Batting Average Rating	Average rating of batsmen of TeamB that batted
TeamB Bowling Average Rating	Average rating of bowlers of TeamB that bowled
Match Type	Either ODI or T20
Winner	Name of the winning Team

3.2 Feature Selection

Recursive Feature Elimination or *RFE* of Sklearn has been used to select the best features for predicting the outcome of a match. *Recursive Feature Elimination* works by eliminating less relevant features and then building model on the remaining features. By using model accuracy, it determines which features and combination of features contribute the most in determining the class attribute. By applying *RFE* on the dataset with *Decision Tree Classifier*, 7 features have been selected as the best features for maximizing the accuracy of prediction. The selected features are - participating teams, home team, batting and bowling averages of both teams.

3.3 Feature Scaling and Splitting Dataset

As categorical values do not work properly with many machine learning algorithms, firstly the attributes in the dataset have been converted to integers using Sklearn package - *Label Encoder*. Then another Sklearn package - *Standard Scalar* has been used to scale or standardize the values of the attributes according to the Eq. (3).

$$z = (x - u)/s \tag{3}$$

The pre-processed dataset contains 1861 instances. The dataset has been split into train set and test set where train set contains 80% of the total dataset and test set contains the remaining 20% data. To ensure the same representation of all classes in the train and test dataset, stratified cross validation has been applied.

3.4 Applied Machine Learning Algorithms

Supervised machine learning algorithms have been applied to the pre-processed dataset to determine the outcome of a international cricket match. The performance of the machine leaning algorithms have been evaluated by first applying the four learning algorithms on the dataset with all features and then by applying the learning algorithms on the dataset using only the features selected by *Recursive Feature Elimination* algorithm.

ZeroR is the simplest classification method which relies on the target and ignores all predictors [10]. ZeroR classifier predicts the target which appears the most i.e. it simply predicts the majority class. Even though the ZeroR classifier does not have any actual capacity for prediction, it is very effective to establish a baseline performance. ZeroR classifier can be used as a benchmark for other classifiers.

Decision Tree is one of the most popular and used algorithm for classification problems as it is very robust against noise in dataset and generally performs well for predicting categorical outcome. In brief, decision tree algorithm chooses the best feature as root which partitions the dataset best. To choose a feature that partitions the dataset best, the concept of entropy is used. Entropy means randomness and the goal of decision tree algorithm is to eventually partition dataset such that each partition contains only one class i.e. to reduce randomness (entropy). By using *Iterative Dichotomiser 3* (ID3), decision tree algorithm calculates gain of each feature and selects the feature with highest gain as the feature with highest information gain reduces entropy the most.

$$E = \sum_{i=1}^{c} -p_i \log_2 p_i \tag{4}$$

$$Gain(S, A) = Entropy(S) - \sum \frac{\|S_v\|}{\|S\|} Entropy(S_v) \tag{5}$$

Equations (4) and (5) are used for calculating entropy and information gain respectively.

Random Forest or random decision forests are an ensemble learning algorithm. It splits dataset to multiple subset and train learning algorithms on the subsets to create multiple models. It then applies these multiple models on test set and through voting determines the class of unknown instance. In case of decision tree, classifier learns on the whole dataset whereas in case of random forests classifier learns over several different subsets. Diversity increases as the models are built using different dataset and variance decreases as all the models are ensemble together for predicting the target. As a result, the resultant ensemble model performs much better.

XGBoost stands for *eXtreme Gradient Boosting*. This algorithm is a scalable end-to-end tree boosting system and is an implementation of gradient boosting machines presented by Tianqi Chen in paper *XGBoost: A Scalable Tree Boosting System* [3]. XGBoost algorithm handles missing data automatically and support parallelization of tree construction which increases the execution speed. Boosting is an ensemble technique where new models are sequentially added to fix the errors made by previous models and it is continued until no more improvement can be achieved. In gradient boosting, the new model firstly predict the errors of existing models and then added together to make prediction over dataset. As the XGBoost algorithm uses gradient descent algorithm to minimize the loss while adding new models, it is also called gradient boosting algorithm [11].

4 Result and Analysis

The experiments have been conducted in Python environment using packages such as - Sklearn, Pandas and XGBoost. To assess the performance of the models,

evaluation metrics - accuracy, precision, recall, f1-score, AUC (area under the curve); have been used.

The dataset contains total 1861 instances which have been split into train set (80%) and test set (20%). The machine learning algorithms have been applied to the dataset initially containing all the features and then containing only the selected 7 features by RFE algorithm. Table 2 shows the obtained train and test accuracy. In Fig. 2, comparison of accuracy among the learning classifier algorithms for the selected best 7 features has been shown.

Table 2. Accuracy on training and test dataset

Algorithms	Train accuracy for all features	Test accuracy for all features
ZeroR	11.90 %	11.29 %
Decision Tree	77.2 %	65.32 %
Random Forests	92.20 %	81.18 %
XGBoost	93.39 %	85.48 %

Fig. 2. Comparison between model performances for 7 selected attributes on train and test dataset

The models performances have also been evaluated by other metrics - precision, recall, f1-score, auc. Precision is a measure of correctly positive prediction rate. Precision is calculated by determining the rate of true positive (TP) from the predicted total positive that is the summation of true positive (TP) and false positive (FP).

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

Recall is a measure of correctly-predicted positive class among all the positive class instances present in the dataset.

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

F1 Score is the weighted average of Precision and Recall. It is calculated by using the formula (8)

$$F1 - Score = \frac{2 * Recall * Precision}{Recall + Precision} \tag{8}$$

AUC or area under curve is the area under ROC curve and the bigger the value of auc the better the performance of the model. By using these metrics, performances of the models have evaluated. In Fig. 3, the precision-recall curve or PR-curve for the applied algorithms have been illustrated. As this is a multi-class classifier problem, to generate precision-recall curve for all the target attributes *OneVsAll* method has been used.

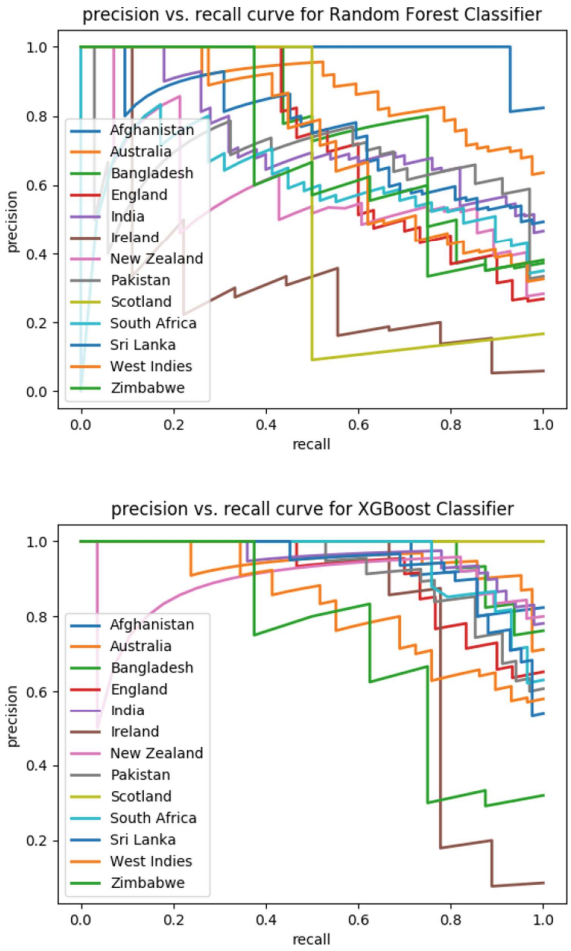

Fig. 3. PR curve for Random Forest classifier and XGBoost classifier

In Fig. 4, the receiver operating characteristic curve for Random Forest and XGBoost algorithm has been presented. By analyzing Fig. 4, it appears that area under the ROC curve for XGBoost model is larger than the area under for Random Forest model.

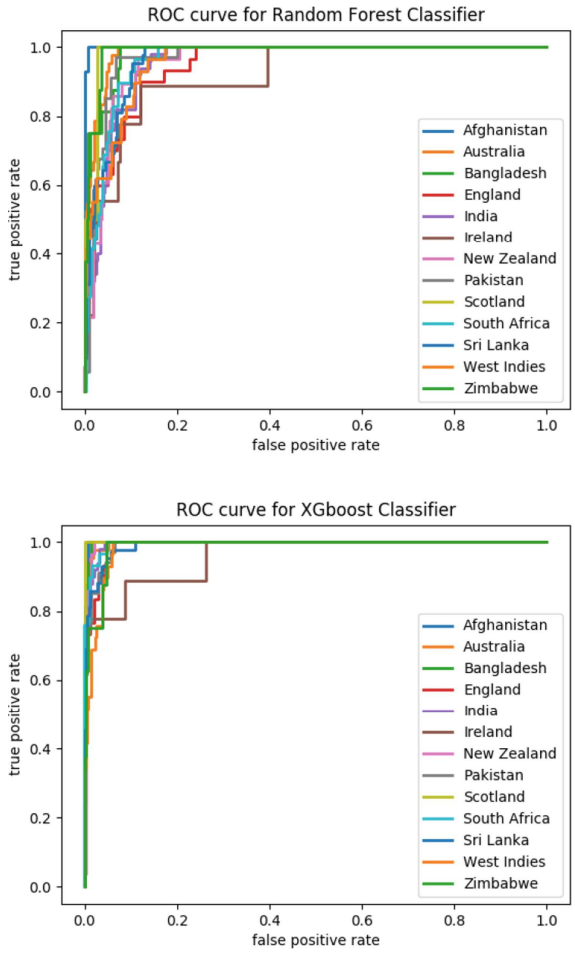

Fig. 4. ROC curve for Random Forest classifier and XGBoost classifier

As XGBoost learning algorithm has performed the best for predicting the target class, in Table 3 value of precision, recall, f1-score, AUC for XGBoost classifier obtained by using *OneVsAll* approach is presented.

The precision-recall curves for XGBoost classifier for several target attributes has been presented in Fig. 5 which illustrates that XGBoost classifier performs well in determining class attributes that is to predict which team is going to win a cricket match.

Figure 6 displays the receiver operatic characteristic curves for XGBoost classifier for different class variables.

From the presented data, it can be analyzed that the accuracy of the models can be improved by reducing the number of classes. From Table 3, we can see

Table 3. Detailed performance evaluation on test dataset for XGBoost learning algorithm

Class	Precision	Recall	F1-Score	Area under ROC
Afghanistan	0.76	0.93	0.84	0.9587
Bangladesh	1.00	0.81	0.90	0.9063
England	0.85	0.93	0.89	0.9593
Australia	0.91	0.95	0.93	0.9701
India	0.89	0.94	0.91	0.9606
Ireland	1.00	0.78	0.88	0.8888
Netherlands	1.00	0.83	0.91	0.9166
Pakistan	0.78	0.82	0.80	0.8999
Scotland	1.00	1.00	1.00	1.0000
Sri Lanka	0.87	0.93	0.90	0.9551
Zimbabwe	0.67	0.75	0.71	0.8708
New Zealand	0.93	0.89	0.91	0.9435
South Africa	0.90	0.90	0.90	0.9439
West Indies	0.76	0.76	0.76	0.8691
Canada	1.00	0.50	0.67	0.7500
Kenya	0.57	1.00	0.73	0.9959
Hong Kong	0.33	1.00	0.50	0.9973
UAE	0.75	0.60	0.67	0.7986

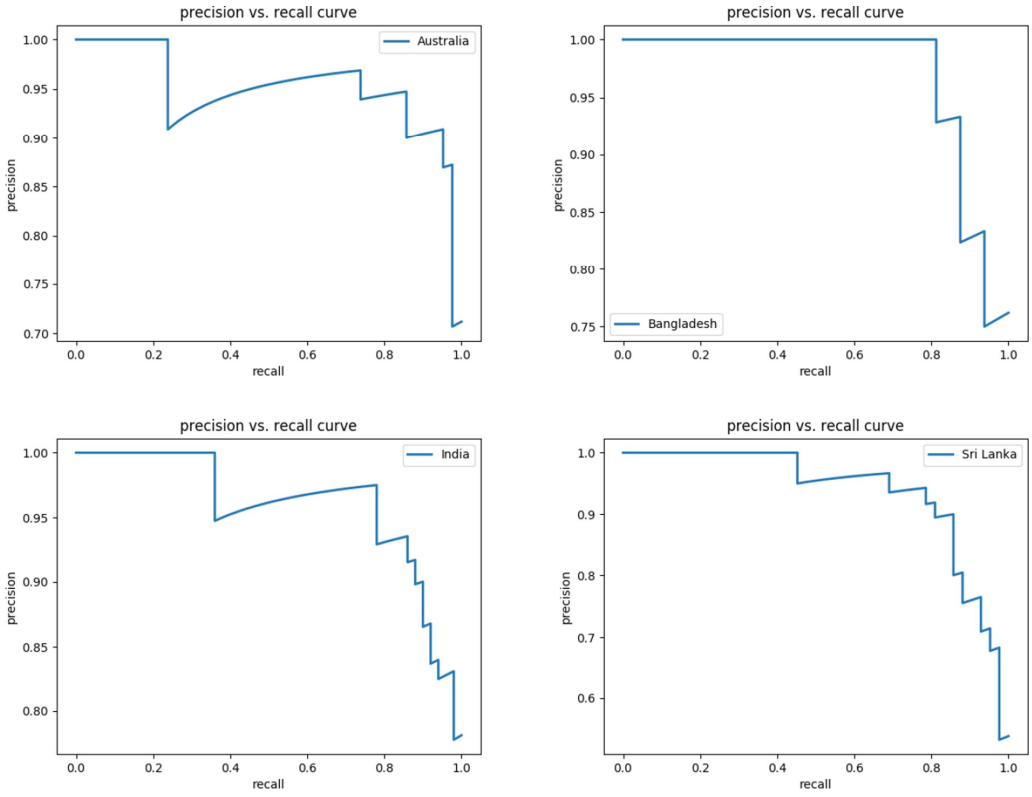

Fig. 5. PR curve for XGBoost classifier for Australia, Bangladesh, India and Sri Lanka

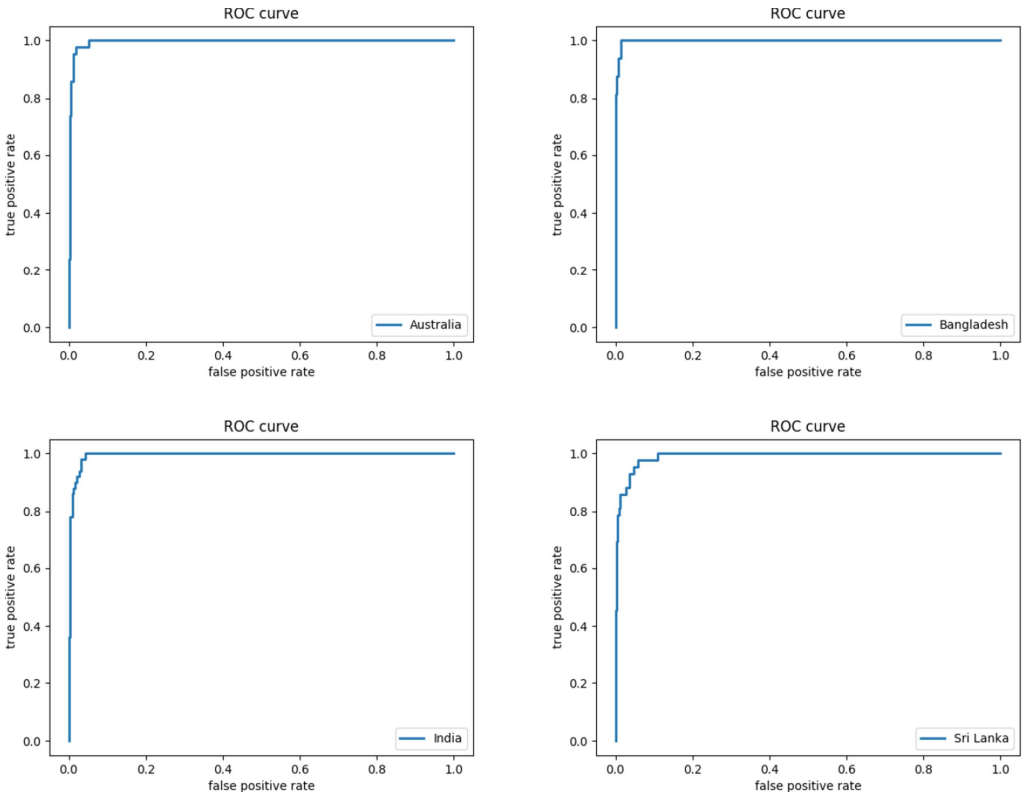

Fig. 6. ROC curve for XGBoost classifier for Australia, Bangladesh, India and Sri Lanka

that classes like - Kenya, Hong Kong, Zimbabwe; have very low precision. There are few classes similar to the mentioned ones which are very under represented and also ratings for players of these teams are not available which results in inaccurate predictions and thus decreases the average prediction of the model.

5 Conclusion and Future Work

Identifying the influential features set for predicting the outcome of a cricket match is still an ongoing study. In this paper we have presented a machine learning approach to predict the outcome of a cricket match. For selecting the influential features, *Recursive Feature Elimination* algorithm has been used. Machine learning algorithms - ZeroR, Decision Tree, Random Forest and XGBoost; have been applied over dataset to predict the result of a match and XGBoost learning algorithm has performed the best with a test accuracy of 85.48%.

Based on what we have completed in this research study, improvements can still be made. This study contains international cricket matches played between 2004–2018; by increasing the time-frame the length of the dataset can be increased which may improve the test accuracy further. The accuracy of prediction may also be increased by implementing neural network model and using different feature selection model.

References

1. Ahmad, H., Daud, A., Wang, L., Hong, H., Dawood, H., Yang, Y.: Prediction of rising stars in the game of cricket. IEEE Access **5**, 4104–4124 (2017)
2. Bailey, M., Clarke, S.R.: Predicting the match outcome in one day international cricket matches, while the game is in progress. J. Sports Sci. Med. **5**(4), 480 (2006)
3. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794 (2016)
4. Ganesh, T.: Yorkr package. https://www.rdocumentation.org/packages/yorkr. Accessed 22 Feb 2020
5. ICC archived site (2004). http://web.archive.org/web/20070301143620/http://www.icc-cricket.com/icc/test/archive/2004.html. Accessed 10 Feb 2020
6. 2019 men's cricket world cup most watched ever. https://www.icc-cricket.com/media-releases/1346930. Accessed 18 Feb 2020
7. International cricket council reliance ranking. http://www.relianceiccrankings.com. Accessed 22 Feb 2020
8. India constitutes 90 percent of one billion cricket fans: ICC research (2018). https://economictimes.indiatimes.com/news/sports/india-constitutes-90-percent-of-one-billion-cricket-fans-icc-research/articleshow/64760726.cms?from=mdr. Accessed 18 Feb 2020
9. Iyer, S.R., Sharda, R.: Prediction of athletes performance using neural networks: an application in cricket team selection. Expert Syst. Appl. **36**(3), 5510–5522 (2009)
10. Kaushik, S., Choudhury, A., Dasgupta, N., Natarajan, S., Pickett, L.A., Dutt, V.: Evaluating frequent-set mining approaches in machine-learning problems with several attributes: a case study in healthcare. In: International Conference on Machine Learning and Data Mining in Pattern Recognition, pp. 244–258. Springer (2018)
11. Kumar, A.: Xtreme gradient boosting algorithm (XGBoost) (2018). http://www.ashukumar27.io/xgboost/. Accessed 26 Feb 2020
12. Munir, F., Hasan, M., Ahmed, S., et al.: Predicting a T20 cricket match result while the match is in progress. Ph.D. thesis, Brac University (2015)
13. Mustafa, R.U., Nawaz, M.S., Lali, M.I.U., Zia, T., Mehmood, W.: Predicting the cricket match outcome using crowd opinions on social networks: a comparative study of machine learning methods. Malays. J. Comput. Sci. **30**(1), 63–76 (2017)
14. Muthuswamy, S., Lam, S.S.: Bowler performance prediction for one-day international cricket using neural networks. In: IIE Annual Conference. Proceedings, p. 1391. Institute of Industrial and Systems Engineers (IISE) (2008)
15. Passi, K., Pandey, N.: Increased prediction accuracy in the game of cricket using machine learning. arXiv preprint arXiv:1804.04226 (2018)
16. Rushe, S.: Cricsheet. https://cricsheet.org/. Accessed 22 Feb 2020
17. Sankaranarayanan, V.V., Sattar, J., Lakshmanan, L.V.: Auto-play: a data mining approach to ODI cricket simulation and prediction. In: Proceedings of the 2014 SIAM International Conference on Data Mining, pp. 1064–1072. SIAM (2014)
18. Singh, S., Kaur, P.: IPL visualization and prediction using HBase. Procedia Comput. Sci. **122**, 910–915 (2017)
19. Singh, T., Singla, V., Bhatia, P.: Score and winning prediction in cricket through data mining. In: 2015 International Conference on Soft Computing Techniques and Implementations (ICSCTI), pp. 60–66. IEEE (2015)
20. Wikipedia contributors: International cricket in 2004—Wikipedia, the free encyclopedia (2020). https://en.wikipedia.org/w/index.php?title=International_cricket_in_2004&oldid=937461535. Accessed 10 Feb 2020