Predicting the Outcome of ODI Cricket Matches using Decision Trees and MLP Networks

Rajeev Kumar*,Jalaz Kumar†
*Assistant Professor,†Student
Department of Computer Science and Engineering
National Institute of Technology Hamirpur, India
Email: *rajeev@nith.ac.in,†jalazkumar1208@gmail.com

***Abstract*—Applying Data Mining & Machine Learning in Sports Analytics is a booming sector in the field of Computer Science. After Football, Cricket is the second most popular sports with a fan base of around 2.5 billion and most popular in South Asia, Australia, The Caribbean and UK. The game has tremendous spectator support in more than 100 nations and the masses show great interest in predicting the game outcomes. There are lots of pre-game and in-game attributes which decides the outcome of a cricket match. Pre-game attributes like the venue, past track-records, Innings(First/Second), team strength etc. and in-game attributes like Toss, run rate, wickets in hand, strike rate etc. influence the result of a match in a predominant manner. In our study, we have used 2 different ML approaches, Decision Trees & MultiLayer Perceptron Network, to predict how these factors affect the outcome of an ODI cricket match. Based on the observed results, We have designed CricAI: Cricket Match Outcome Prediction System. Our designed tool takes into consideration the pre-game attributes like the ground, venue (home, away, neutral) & innings (first/second) to predict the outcome of given match.**

Keywords: Decision Tree Classifier, MLP Classifier, Features, Performance Measures

I. INTRODUCTION

Cricket which is the world's second most popular sport after soccer is basically a bat and ball game played between two teams of eleven players each. Each team comes to bat and has a single inning in which it seeks to score as many runs as possible, while the other team fields. The innings ends when the total quota of deliveries, which depends on game format has turned up, or the 10 batsmen have been dismissed, whichever comes first. The prime objective is to score more runs & thus Runs are the decisive factor.

Game of Cricket is highly unpredictable in nature. Until the very last moment, it is difficult to make accurate predictions about the game. Various natural factors affecting the game output, huge betting market and enormous media coverage have given strong incentives to model this game from the Machine Learning Perspective.

International Cricket Council (ICC) is the governing body which decides the rules of Cricket.

There are three internationally recognized formats of Cricket matches - Test match, ODI match(One Day International) and T20 match. The main difference between these three formats is the scheduled duration of the game which directly modifies the number of deliveries each team got to play in their respective innings.

Test cricket format is the longest one and is considered as the highest standard of the game. Match duration is five days in which each team get to play 2 innings each. A standard day of test cricket match comprises of 3 sessions each of 2 hours.

One Day International i.e. ODI format is in limited overs, where each team faces 300 deliveries(50 overs). ODI match is scheduled to complete in a Day or a Day/Night combination.

T20 is the shortest internationally recognized format of this game, where each team innings consist of 20 overs. This is more of an "explosive" and more "athletic" than the other two formats.

Our study is focused on the most popular format of Cricket, One Day Internationals or the ODIs. The outcome of ODI match is influenced by a large no. of factors and can be predicted like all other games. We need to find the best attributes or factors that influence the match outcome. For our study, we considered the factors analyzed by [1] and [2], which are proven to have a significant impact on the outcome of ODI match. The factors considered for analysis include:

- **Teams Past Performance:** This factor captures the historic outcomes of all the matches played between the teams.
- **Ground:** This plays a vital role as teams have great track records on particular grounds and carry psychological superiority over the other.
- **Innings:** This factor determines which team batted first & which batted second.
- **Home Game Advantage:** This is achieved by using Venue feature, which determines whether a particular ground is home/away/neutral for each of the playing teams.

Both of our classifiers are trained on the basis of these factors. For predicting the outcome of ODI matches we have used 2 supervised classification techniques - Decision Trees and Multi-Layer Perceptron Networks. We have conducted comparative studies between both the classifiers and summarized the results in this paper.

We then built a software tool called CricAI based on emerged results, which can be used to predict the outcome of any ODI match given the concerned factors as inputs. This software of ours can be of real value to the cricketers, support staff of teams and cricket analysts in terms of analyzing the future game in advance and working towards maximizing their chances of victory.

Clustering couldn't have made any contribution to our research as we dealt with multiple independent attributes, therefore placing them in clusters after finding similarity did not seem feasible.

The rest of this paper is organised as follows. Section 2 explains the approach we have used for conducting the analysis. Section 3 presents a comparative study of the classifiers used. Section 4 presents the related work in this domain area. Section 5 gives the conclusions and the future scope associated with this approach.

II. APPROACH FOR ANALYSIS

A. Data Collection

Data was extracted from [3] by running a scraping script in a justified manner, sending 1 request per second.

TABLE I: Scrapped Dataset Format

Match Id	Team 1	Team 2	Winner	Margin	Ground
ODI #1	Australia	England	Australia	5 wickets	Melbourne
ODI #2	England	Australia	England	6 wickets	Manchester
ODI #3	England	Australia	Australia	5 wickets	Lord's

Dataset comprises of all the ODI matches from Jan 5, 1971, to Oct 29, 2017. A total of 3933 ODI match results were scrapped. The collected dataset was subjected to cleaning process where some of the matches were deleted from the analysis. Since it's not possible to foresee the impact of nature on cricket, matches which either ended up in a tie/draw or interrupted by rain, were being removed from the dataset. Matches of Special teams like World XI, Asia XI & Africa XI were also removed.

We also further replicated our dataset two times by swapping the team positions i.e. A game between Team 1: India and Team 2: Sri Lanka was also replicated as Team 1: Sri Lanka and Team 2: India. For further making the dataset suitable for input to the various Machine Learning Classifier Models, we converted the continuous dataset into a categorical dataset, using dummy variables.

Innings Feature was determined by first translating Column: *Margin* into Column: *Winner Innings* using:

· Win by Wickets $\implies$ Winner Innings: 2

· Win by Runs $\implies$ Winner Innings: 1

Further, Using Column: *Winner* and the generated Column: *Winner Innings*, we aquired the innings of each team per match.

Venue Feature was determined by using Column: *Winner* and Scrapped dataframe from [3] which provided the names of cricket grounds in all countries. Combining both of these, Column: *Host Country* was generated, which was used to get venue of a match with respect to both the teams.

The dataset was saved in comma separated format. We used a total of 7494 match records for our analytical study which was further divided into the testing and training data.

⋆ Training Dataset Size: 5620

⋆ Testing Dataset Size: 1874

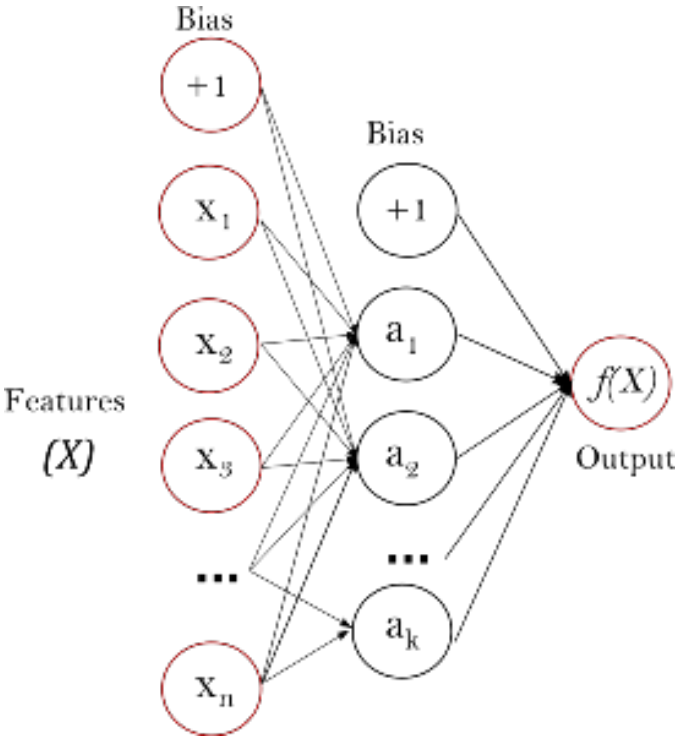


Fig. 1: Multilayer Perceptron Network

B. MultiLayer Perceptron Networks

MLP Network is a type of supervised learning algorithm which learns a function

$$f(.) : R^m \rightarrow R^o$$

by training on a dataset, where *o* is the total number of output units and *m* is the total number of input units. Given features set $X = x_1, x_2, ..., x_m$ and a target y, MLP Network can be trained to be a non-linear function approximator for classification as well as regression. The core difference between MLP Networks and Logistic regression is in the former one there can be one or more nonlinear layers, called hidden layers. Fig 1. shows a Multi-layer Perceptron Network with only 1 hidden layer.

The leftmost layer which represents the input features, known as the input layer, consists of a set of neurons.

$$x_i | x_1, x_2, \ldots, x_m$$

Each neuron in the hidden layer transforms the values from the previous layer with a weighted linear summation,

$$w_1 x_1 + w_2 x_2 + \ldots + w_m x_m$$

followed by a non-linear activation function acting on its output. The last hidden layer further transfers these values towards the output layer which transforms these intermediate values into the final output values.

MLPClassifier [4] is implemented using a multi-layer perceptron (MLP) algorithm in which Backpropagation is used for training. More precisely, the actual training of dataset is

done using some form of gradient descent whose gradients are calculated using Backpropagation.

MLP trains on two arrays: array **X** of size(n_samples, n_features) which holds the training samples represented as floating point feature vectors; and array **y** of size (n_samples) which holds the target values (class labels) for the training samples.

Currently, MLPClassifier[4] supports only the Cross-Entropy loss function, using which we can derive the estimated probabilities by running predict_proba function. The model also supports multi-label classification in which any input feature set can belong to more than one class which makes it quite suitable for our approach.

Advantages:

∗ MLP Networks are capable to run all types of non-linear models.

∗ MLPClassifier uses Backpropagation so, it continously learns and improvize itself.

∗ MLP Networks are capable to learn & train in realtime using partial fitting property.

Disadvantages:

∗ MLP Networks are highly sensitive for feature scaling.

∗ It uses a black box model, results may be more difficult to interpret.

∗ MLP Networks requires a large number of hyperparameters like the number of hidden neurons, layers and iterations to be properly tuned.

C. Decision Trees

Decision Trees are also a type of Supervised Machine Learning techniques where the input data while training is continuously split according to a certain parameter. Any decision tree can be explained using two of its entities, decision nodes and leaf nodes. The leaves denote the final outcomes or the overall decisions made and at the decision nodes, our data is split using some entropy calculation. Decision Trees (DTs) can be used for both classification as well as regression problems. Our goal is to create a supervised model which can be able to predict the value of any input target variable by making use of the simple decision rules inferred from the features of the dataset.

Given training vectors $x_i \epsilon\ R^n$, *i=1,..., l* and a label vector $y \epsilon\ R^l$, a decision tree recursively partitions the entire data space such that the data samples which have the same labels are grouped together. Let the data at node *m* be represented by *Q*. For each candidate split θ *(j,*t_m*)* consisting of a feature *j* and threshold t_m , partition the data into $Q_{left}(\theta)$ and $Q_{right}(\theta)$ subsets as,

$$Q_{left}(\theta) = (x, y)|x_j <= t_m$$

$$Q_{right}(\theta) = Q/Q_{left}(\theta)$$

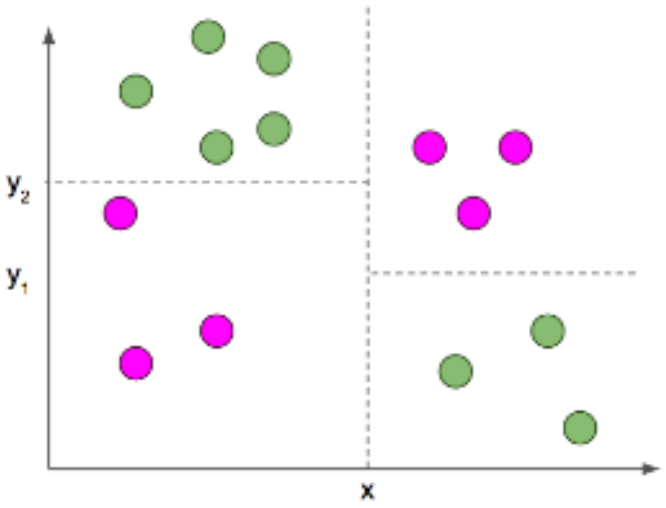


Fig. 2: Decision Tree

The impurity at *m* is computed using an impurity function *H()* , the choice of which depends on the task being solved (classification or regression).

$$G(Q, \theta) = \frac{n_{left}}{N_m} H(Q_{left}(\theta)) + \frac{n_{right}}{N_m} H(Q_{right}(\theta))$$

Select the parameters that minimises the impurity

$$\theta^* = argmin_\theta G(Q, \theta)$$

Recurse for subsets $Q_{left}(\theta^*)$ and $Q_{right}(\theta^*)$ until the maximum allowable depth is reached, $N_m < min_{samples}$ or N_m=1.

Advantages:

∗ Decision Trees are simple enough to understand, interpret its outcome and visualize the results.

∗ Able to handle both numeric as well as categorical data and also multi-output problems.

∗ The White box model is followed up. If some situation is observable in the model, then its explanation is easily explained using the logic of Boolean Algebra.

Disadvantages:

∗ Sometimes complex trees are created which are not able to generalize the data well. Decision Trees are prone to Over-fitting.

∗ Decision trees are usually very unstable and even small modifications in the data might result in the generation of a completely different tree.

∗ For the cases, where some classes dominate creation of biased Decision Tree takes place.

III. RESULTS AND OBSERVATIONS

A. Performance Measures

To evaluate classifier performance in a well effective manner, we need to define the performance measure. Efficiency

and goodness of any classifier is measured by the various defined performance measures which is itself a single index.

We have performed a comparative analysis of our classifiers considering the following performance measures:

Accuracy Score: This compares the actual outcomes with the predicted outcomes of our classifier for a given input dataset. For best Accuracy Score, the set of actual true labels in testing dataset must match the corresponding set of predicted labels.

In cases of presence of imbalanced classes, Precision-Recall is a useful index to measure the success of prediction. In information retrieval, result relevancy is measured by precision, while recall is a measure of the total number of truly relevant results which were returned.

Precision Score: This is defined as the number of true positives (T_p) over the number of true positives plus the number of false positives (F_p)

$$P = \frac{T_p}{T_p + F_p}$$

The precision is intuitively the ability of the classifier not to label as positive a sample that is negative. Best value: 1 and Worst value: 0.

Recall Score: This is defined as the number of true positives (T_p) over the number of true positives plus the number of false negatives (F_n)

$$P = \frac{T_p}{T_p + F_n}$$

The recall is intuitively the ability of the classifier to find all the positive samples. Best value: 1 and Worst value: 0.

F1 Score: This is defined as the interpretation of a weighted average of the precision and recall of a classifier. Numerically, it is the harmonic mean of precision and recall.

$$F1 = 2\frac{P * R}{P + R}$$

It is also known as the F-measure or balanced F-score. The relative contribution of precision and recall to the F1 score are equal.

Average Precision Score: The precision-recall curve is summarised as the weighted mean of precisions achieved at each threshold, with the increase in recall from the previous threshold used as the weight:

$$AP = \Sigma_k (R_k - R_{k-1}) P_k$$

where R_k and P_k are the recall and precision at the k^{th} threshold.

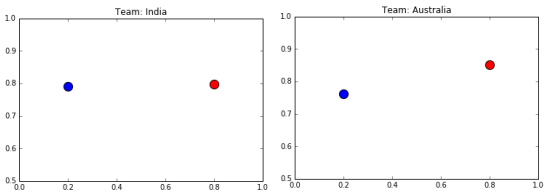
B. Comparative Analysis

Accuracy Score:

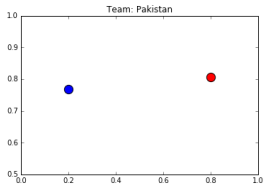
	MLP Classifier	Decision Tree Classifier
Accuracy Score	0.574	0.551

Observation: We selected 3 teams: India, Australia and Pakistan randomly and separated the match records of these 3 teams to obtain the performance measure for them separately.

Team Name	Training Dataset Size	Testing Dataset Size
India	1320	440
Australia	1288	430
Pakistan	1281	427

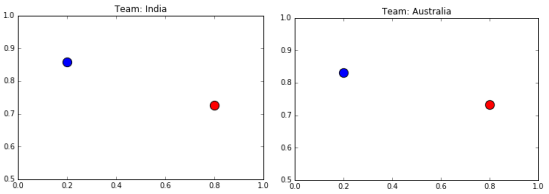


(a) P = 0.791, R = 0.797 (b) P = 0.760, R = 0.850

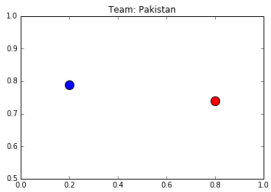


(c) P = 0.767, R = 0.806

Fig. 3: Precision-Recall Scatter Plot for MLP Classifier.



(a) P = 0.859, R = 0.726 (b) P = 0.830, R = 0.733



(c) P = 0.789, R = 0.739

Fig. 4: Precision-Recall Scatter Plot for DT Classifier.

Classifier	Performance Measure	India	Australia	Pakistan
MultiLayer Perceptron Classifier	Recall_Score	0.797	0.850	0.806
	Precision_Score	0.791	0.760	0.767
	F1_Score	0.794	0.803	0.786
	Average P_Score	0.744	0.749	0.724
Decision Tree Classifier	Recall_Score	0.726	0.733	0.739
	Precision_Score	0.859	0.830	0.789
	F1_Score	0.787	0.779	0.763
	Average P_Score	0.785	0.779	0.719

IV. RELATED WORK

From our literature survey, we observed that game of cricket has very few machine learning related work done on it. Though cricket shares some features with sports like baseball, it still remains unique in various ways and thus deserves to be analyzed in an independent manner.

Statistical Approach is the base of most of the analyzing studies done on cricket.

Bailey and Clarke conducted a study for predicting the outcome of an in-progress game in one-day international cricket [5]. WASP(Winning and Score Predictor), 2012 is a product grounded on the theory of Dynamic Programming, by Dr Scott Brooker and Dr Seamus Hogan at the University of Canterbury in New Zealand.

Neeraj Pathak & Hardik Wadhwa conducted a similar comparative analysis of match outcomes using the classification models: Support Vector Machines, Random Forests and Naive Bayes[6].Preeti Satao and Team predicted the score of cricket match using Clustering Techniques[7].

In Parag Shah, Mitesh Shah[8] and Amal Kaluarachchi, Aparna S. Varde[9], they explored the statistical significance of a range of factors & game-attributes which explain the outcome of a cricket match. In particular, home crowd advantage, match type (day-night/day), past performance of the team against each other & game plan (batting first or fielding first) were the key interests in their investigation.

Madan Gopal Jhanwar and Vikram Pudi used a supervised learning approach from some team composition perspective to predict the outcome of a One Day International (ODI) cricket match. Their work suggested that one of the distinctive features for predicting the winner is the relative team strength of both the competing teams. Swetha and Saravanan.KN analysed the factors that cricket game depends on and decides Winning[1].

V. CONCLUSION

In our study, we performed a comparative analysis of the predictions generated by 2 different supervised classification models for the same input dataset. We have been able to predict the match outcome using the features from the dataset.

The main contributions of our work are:

- Comparative analysis of performance measure of two different supervised learning techniques.
- Analysing all the factors which strive to affect the outcome of the cricket match.
- Development of the Prediction tool that can be used to predict the chances of winning, using input attributes.

As future work, we plan to expand this analysis more from the team composition perspective. Also, the relevance of considering 1980s match data equivalent to the 2017s match data also need to be analysed and worked upon. We can also apply our methodology and technique to predict the outcomes of games like hockey and football.

VI. ACKNOWLEDGMENT

This research was supported by the Department of Computer Science and Engineering, NIT Hamirpur, India.

We are grateful to all our colleagues who provided support and insight which assisted us a lot in carrying out this research.

We also thank all of them for their worthy comments & criticism on an earlier version, although any errors are our own and reputations of these esteemed persons should not be tarnished.

REFERENCES

[1] Swetha and Saravanan KN, "Analysis on Attributes Deciding Cricket Winning", International Research Journal of Engineering and Technology (IRJET), p-ISSN: 2395-0072, Volume: 04 Issue: 03 March-2017

[2] Mehvish Khan and Riddhi Shah. "Role of External Factors on Outcome of a One Day International Cricket (ODI) Match and Predictive Analysis", International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 6, June 2015.

[3] ESPN Cricinfo, http://www.stats.espncricinfo.com

[4] Scikit learn, http://scikit-learn.org/stable/index.html

[5] Bailey and Clarke, Journal of Sports Science and Medicine, 2006, Vol. 5, pp. 480487.

[6] Neeraj Pathak and Hardik Wadhwa,"Applications of modern classification techniques to predict the outcome of ODI Cricket". 2016 International Conference on Computational Science.

[7] "CRICKET SCORE PREDICTION SYSTEM (CSPS) USING CLUSTERING ALGORITHM", Preeti Satao, Ashutosh Tripathi, Jayesh Vankar, Bhavesh Vaje, Vinay Varekar. International Journal Of Current Engineering and Scientic Research (IJCESR), 23940697, Volume-3, Issue-4, 2016.

[8] Parag Shah and Mitesh Shah, "Predicting ODI Cricket Result". Journal of Tourism, Hospitality and Sports, 2312-5179, Vol.5, 2015.

[9] Kaluarachchi, Amal, and S. Varde Aparna. "CricAI: A classication based tool to predict the outcome in ODI cricket." 2010 Fifth International Conference on Information and Automation for Sustainability. IEEE, 2010.

[10] Madan Gopal Jhanwar and Vikram Pudi, "Predicting the Outcome of ODI Cricket Matches: A Team Composition Based Approach", European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, Report No: IIIT/TR/2016/-1, Conference Center, Riva del Garda.