CRICAI: CRICKET MATCH OUTCOME PREDICTION SYSTEM

A PROJECT

Submitted in partial fulfillment of the
requirements for the award of the degree
of

BACHELOR OF TECHNOLOGY

By

Amandeep Prasad	**14MI507**	**Aditya Thakur**	**14MI521**
Rajat Patiyal	**14MI512**	**Jalaz Kumar**	**14MI528**
Pretesh Chauhan	**14MI516**	**Aarti Ramoul**	**14MI547**

Under the guidance

of

Er. Rajeev Kumar

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY

HAMIRPUR-177005, HP (INDIA)

Dec, 2017

NATIONAL INSTITUTE OF TECHNOLOGY

HAMIRPUR (H.P.)

CANDIDATES' DECLARATION

We hereby certify that the work which is being presented in the project report titled "**CricAI: Cricket Match Outcome Prediction System**" in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology and submitted to the Department of Computer Science and Engineering, National Institute of Technology Hamirpur, is an authentic record of our own work carried out during a period from August 2017 to December 2017, under the supervision of **Er. Rajeev Kumar**, Assistant Professor Department of Computer Science and Engineering, National Institute of Technology Hamirpur.

The matter presented in this project report has not been submitted by us for the award of any other degree of this or any other Institute/University.

Amandeep Prasad **Pretesh Chauhan** **Jalaz Kumar**

Aditya Thakur **Rajat Patiyal** **Aarti Ramoul**

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Date: 13 Dec, 2017

Er. Rajeev Kumar
Assistant Professor
CSED

The project Viva-Voce examination of name of the candidates has been held on 13 Dec, 2017

Er. Rajeev Kumar
Signature of Supervisor

ACKNOWLEDGEMENT

The project report **CricAI: Cricket Match Outcome Prediction System** is outcome of guidance, moral support and devotion bestowed on us throughout our work. For this we acknowledge and express our profound sense of gratitude and thanks to everybody who have been a source of inspiration during the seminar presentation.

We would sincerely thank and acknowledge our deep gratitude to our Mentor **Er. Rajeev Kumar** for his invaluable guidance, constant assistance, support, endurance and constructive suggestions for the betterment of the technical project. We would like to thank all the staff members of the department of computer science and engineering for helping us directly or indirectly in completing this work successfully.

We have mentioned all the various information and sources we have used during our report completion, at the end of report under bibliography section

ABSTRACT

Applying Data Mining & Machine Learning in Sports Analytics is a blooming 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 mostly popular in South Asia, Australia, The Caribbean's and United Kingdom.

It has tremendous spectator support and the masses show great interest in predicting the outcome of games. The result of a cricket match depends on lots of in-game and pre-game attributes. Pre-game attributes like 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 a match result predominantly.

In our study, we have used 2 different Supervised Machine Learning approach for Classification, Decision Trees & MultiLayer Perceptron Network, to predict how these factors affect the outcome of an ODI cricket match.

Based on the emerged results, we have designed CricAI: Cricket Match Outcome Prediction System. Our designed tool takes into consideration the pre-game attributes like ground, venue (home, away, neutral) & innings (first / second) to predict the outcome of given match.

TABLE OF CONTENTS

FIGURE-INDEX

TABLE-INDEX

CHAPTER 1: Introduction

1.1. Motivation:

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 batsman has been dismissed, whichever comes first. The prime objective is to score more runs & thus Runs are the decisive factor.

Game of Cricket is a highly unpredictable in nature. Till 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.

Rules of Cricket are determined by the International Cricket Council (ICC). 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 game. Match duration is five days in which each team get to play 2 innings each. A standard test cricket day consists of 3 sessions of 2 hours each.

One Day International i.e. ODI format is of 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.

We focus our research on One Day Internationals, the most popular format of the game. 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 factor 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 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 them.
- **Ground:** This plays a vital role as teams have great track records on 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 classification models are built using these factors. To predict the outcome of ODI matches we have applied two classification techniques - Decision Trees and Multi Layer Perceptron Networks. We have conducted comparative studies among various 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.

1.2. Problem Definition:

To predict the outcome of an One Day Internationals Cricket match being held between two teams at a particular venue, by using pre-game attributes like teams past records, innings (first/second) & venue (home/away/neutral) with respect to both the teams as the features of our models. We strive to develop a smart outcome prediction tool which is a desktop-app which provides us future predictions. We make use of 3 different classification models:

- Multilevel Perceptron Classifier
- Decision Tree Classifier
- SVM classifier

We also make observations regarding the performance measures of all the 3 classifiers & try to comprehend that information in tabular format, for the ease of doing comparative analysis.

CHAPTER 2: Methodology and Approach

For getting solution of the problem defined, we have taken the Machine Learning Approach using supervised learning methods. The whole approach we took has a base assumption that all the future matches are somehow relatable to the historic data & no setbacks are possible.

The first and foremost phase was the data collection, in which data was collected & made suitable for the input to the models we chose, by applying different data preprocessing techniques. One dataset is available, then training of data was done for the 3 models we have chosen, and accuracy along with other performance feature was evaluated.

Later on, observations were made for each of the model by selecting 3 teams at random & relevant plots were drawn. The last & final phase was the implementation of UI Platform for the tool, which in our case we have chosen Terminal-based App as well as Desktop-App.

2.1. Data Collection:

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

Table 1. Scrapped Data 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 ODIs match results were scrapped. The collected data was subjected to cleaning process where some of the matches were deleted from the analysis. Since the impact of the nature on the game of cricket cannot be foreseen, matches which were either interrupted by rain or ended up in a draw/tie 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 =⇒ Winner Innings: 2

Win by Runs =⇒ Winner Innings: 1

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

Venue Feature was determined by using Column: *Winner* and Scrapped data frame from [1] which provided the names of cricket grounds in all countries. Combining both, a new Column: *Host Country* was generated, which was used to get venue of a match with respect to both the teams.

The data set was saved in comma separated format. A total of 7494 match records were used in analysis. Finally, we divided the dataset into two parts, namely, the test data and the training data.

- Training Dataset Size: 5620
- Testing Dataset Size: 1874

2.2. Multi-Layer Perceptron Network:

A multilayer Perceptron (MLP) is a class of feed forward artificial neural network. An MLP consists of at least three layers of nodes. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called back propagation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable.

Multilayer perceptron are sometimes colloquially referred to as "vanilla" neural networks, especially when they have a single hidden layer.

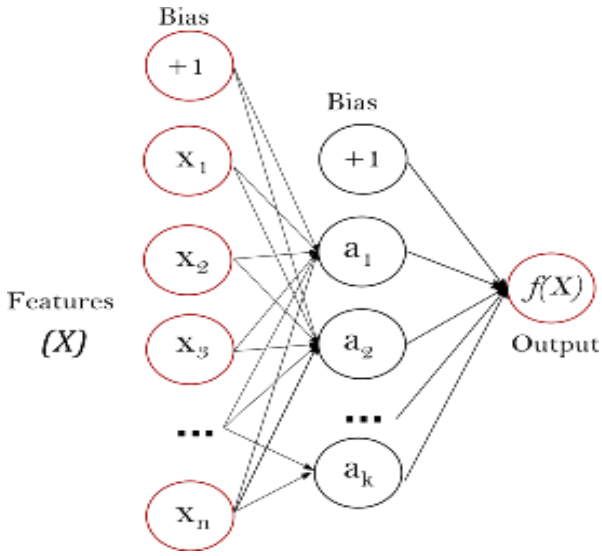

Fig 1. Multilayer Perceptron with 1 hidden layer

2.2.1 Activation Function

If a multilayer perceptron has a linear activation function in all neurons, that is, a linear function that maps the weighted inputs to the output of each neuron, then linear algebra shows that any number of layers can be reduced to a two-layer input-output model. In MLPs some neurons use a nonlinear activation function that was developed to model the frequency of action potentials, or firing, of biological neurons.

Alternative activation functions have been proposed, including the rectifier and softplus functions. More specialized activation functions include radial basis functions (used in radial basis networks, another class of supervised neural network models).

2.2.2 Layers

The MLP consists of three or more layers (an input and an output layer with one or more hidden layers) of nonlinearly-activating nodes making it a deep neural network. Since MLPs are fully connected, each node in one layer connects with a certain weight to every node in the following layer.

2.2.3 Terminology

The term "multilayer perceptron" does not refer to a single perceptron that has multiple layers. Rather, it contains many perceptrons that are organized into layers. An alternative is "multilayer perceptron network". Moreover, MLP "perceptrons" are not perceptrons in the strictest possible sense. True perceptrons are formally a special case of artificial neurons that use a threshold activation function such as the Heaviside step function. MLP perceptrons can employ arbitrary activation functions. A true perceptron performs binary classification (either this or that), an MLP neuron is free to either perform classification or regression, depending upon its activation function.

The term "multilayer perceptron" later was applied without respect to nature of the nodes/layers, which can be composed of arbitrarily defined artificial neurons, and not perceptrons specifically. This interpretation avoids the loosening of the definition of "perceptron" to mean an artificial neuron in general.

2.2.4 Applications

MLPs are useful in research for their ability to solve problems stochastically, which often allows approximate solutions for extremely complex problems like fitness approximation.

MLPs are universal function approximations as showed by Cybenko's theorem, so they can be used to create mathematical models by regression analysis. As classification is a particular case of regression when the response variable is categorical, MLPs make good classifier algorithms.

MLPs were a popular machine learning solution in the 1980s, finding applications in diverse fields such as speech recognition, image recognition, and machine translation software, but thereafter faced strong competition from much simpler and support vector machines. Interest in back propagation networks returned due to the successes of deep learning.

2.2.5 Advantages

- It is capable to run non-linear models.
- MLPClassifier uses Back propagation so, it continuously learns and improvise itself.
- Capability to learn models in real-time using partial fitting.

2.2.6 Disadvantages

- Highly sensitive to feature scaling.
- It uses a black box model, results may be more difficult to interpret.
- MLP requires tuning a number of hyper parameters such as the number of hidden neurons, layers, and iterations

2.3 Decision Trees

The classification technique is a systematic approach to build classification models from an input dataset. For example, decision tree classifiers, rule-based classifiers, neural networks, support vector machines, and naive Bayes classifiers are different technique to solve a classification problem. Each technique adopts a learning algorithm to identify a model that best fits the relationship between the attribute set and class label of the input data. Therefore, a key objective of the learning algorithm is to build predictive model that accurately predict the class labels of previously unknown records.

Decision Tree Classifier is a simple and widely used classification technique. It applies a straightforward idea to solve the classification problem. Decision Tree Classifier poses a series of carefully crafted questions about the attributes of the test record. Each time it receives an answer, a follow-up question is asked until a conclusion about the class label of the record is reached.

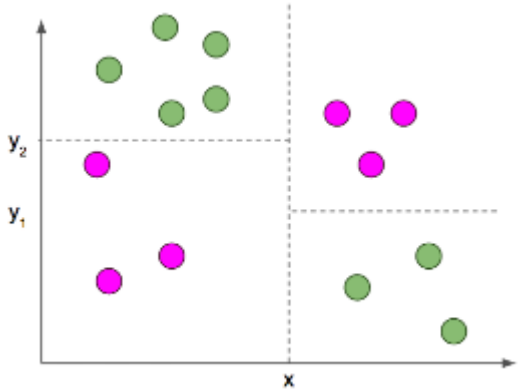

Fig 2. Decision Tree

2.3.1 Build A Decision Tree

Build a optimal decision tree is key problem in decision tree classifier. In general, may decision trees can be constructed from a given set of attributes. While some of the trees are more accurate than others, finding the optimal tree is computationally infeasible because of the exponential size of the search space.

However, various efficient algorithms have been developed to construct a reasonably accurate, albeit suboptimal, decision tree in a reasonable amount of time. These algorithms usually employ a greedy strategy that grows a decision tree by making a series of locally optimum decisions about which attribute to use for partitioning the data. For example, Hunt's algorithm, ID3, C4.5, CART, SPRINT are greedy decision tree induction algorithms.

2.3.2 Determine The Best Attribute Test Condition

The decision tree inducing algorithm must provide a method for specifying the test condition for different attribute types as well as an objective measure for evaluating the goodness of each test condition.

First, the specification of an attribute test condition and its corresponding outcomes depends on the attribute types. We can do two-way split or multi-way split, discretize or group attribute values as needed. The binary attributes leads to two-way split test condition. For nominal attributes which have many values, the test condition can be expressed into multi way split on each distinct values, or two-way split by grouping the attribute values into two subsets. Similarly, the ordinal attributes can also produce binary or multi way splits as long as the grouping does not violate the order property of the attribute values. For continuous attributes, the test condition can be expressed as a comparison test with two outcomes, or a range query. Or we can discretize the continuous value into nominal attribute and then perform two-way or multi-way split.

Since there are may choices to specify the test conditions from the given training set, we need use a measurement to determine the best way to split the records. The goal of best test conditions is whether it leads a homogenous class distribution in the nodes, which is the purity of the child nodes before and after splitting. The larger the degree of purity, the better is the class distribution.

To determine how well a test condition performs, we need to compare the degree of impurity of the parent before splitting with degree of the impurity of the child nodes after splitting. The larger their difference, the better is the test condition. The measurements of node impurity/purity are:

- Gini Index
- Entropy
- Misclassification Error

2.3.4 Stop the Split Procedure

A stop condition is also needed to terminate the tree-growing process. A possible strategy is to continue expanding a node until either all the records belong to the same class or all the records have identical attribute values. Although they are sufficient conditions to stop decision tree induction algorithm, some algorithm also applies other criteria to terminate the tree-growing procedure earlier.

2.2.4 Advantages

- Quite Simple to understand, interpret and visualize.
- Able to handle both numeric as well as categorical data and also multi-output problems.
- Uses a white box model. If a given situation is observable in a model, the explanation for the condition is easily explained by boolean logic.

2.2.4 Disadvantages

- Creation of over-complex trees that do not generalize the data well. Overfitting is a problem in Decision Tree.
- Decision trees can be unstable because small variations in the data might result in a completely different tree being generated.
- For the trivial cases, where some classes dominate over all others creation of biased Decision Tree usually takes place.

2.4 Support Vector machine Classifier:

"Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification and regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well.

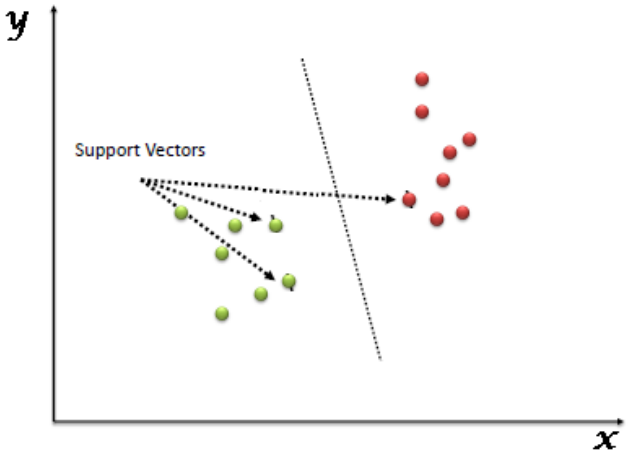

Fig 3. Support Vector Machine Classification

In machine learning, support vector machines (SVMs), are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis.

Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting).

An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-

dimensional feature spaces.

When data are not labeled, supervised learning is not possible, and an unsupervised learning approach is required, which attempts to find natural clustering of the data to groups, and then map new data to these formed groups. The clustering algorithm which provides an improvement to the support vector machines is called support vector clustering[2] and is often[citation needed] used in industrial applications either when data are not labeled or when only some data are labeled as a preprocessing for a classification pass.

3.1 Data Preparation For Svm

This section lists some suggestions for how to best prepare your training data when learning an SVM model.

- **Numerical Inputs:** SVM assumes that your inputs are numeric. If you have categorical inputs you may need to covert them to binary dummy variables (one variable for each category).
- **Binary Classification:** Basic SVM as described in this post is intended for binary (two-class) classification problems. Although, extensions have been developed for regression and multi-class classification.

CHAPTER 3: Results & Observations

3.1. Performance Measures

To evaluate Classifier performance in a well effective manner, we need to define the performance measure. A Classifier performance measure is a single index that measures the Goodness of the classifiers considered.

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 labels predicted for a sample must match the corresponding set of labels in y_true.

Precision-Recall is a useful measure of success of prediction when the classes are very imbalanced. In information retrieval, precision is a measure of result relevancy, while recall is a measure of how many truly relevant results are returned.

Precision Score: This is defined as the number of true positives (Tp) over the number of true positives plus the number of false positives (Fp). The precision is intuitively the ability of the Classifier not to label as positive a sample that is negative. The best value is 1 and the worst value is 0.

Recall Score: This is defined as the number of true positives (Tp) over the number of true positives plus the number of false negatives (Fn). The recall is intuitively the ability of the Classifier to find all the positive samples. The best value is 1 and the worst value is 0.

F1 Score: This is defined as the interpreted as a weighted average of the precision and recall. It is the harmonic mean of precision and recall. It is also known as the balanced F-score or F-measure. The relative contribution of precision and recall to the F1 score are equal.

Average Precision Score: This summarizes a precision-recall curve as the weighted mean of precisions achieved at each threshold, with the increase in recall from the previous threshold used as the weight.

3.2. Comparative Analysis

Table 2 Accuracy Score of all 3 models

DT Classifier	MLP Classifier	SVM Classifier
0.551	0.574	0.612

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.

Table 3 Splitted Data for Observations

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

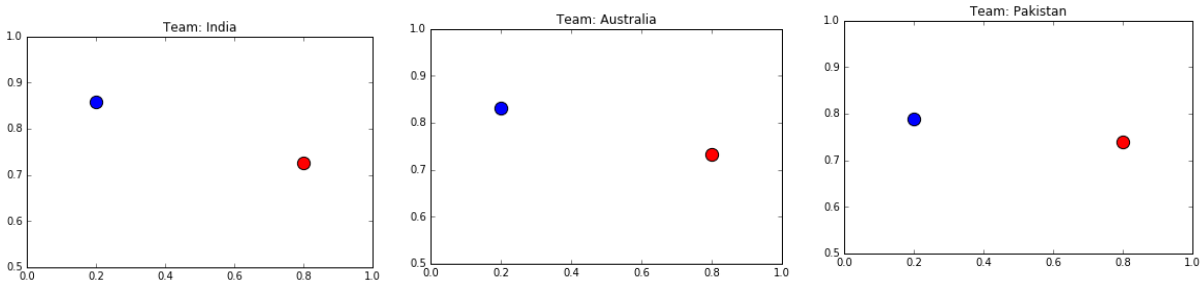

Fig 4 Recall-Precision Score Value for DT Classifier

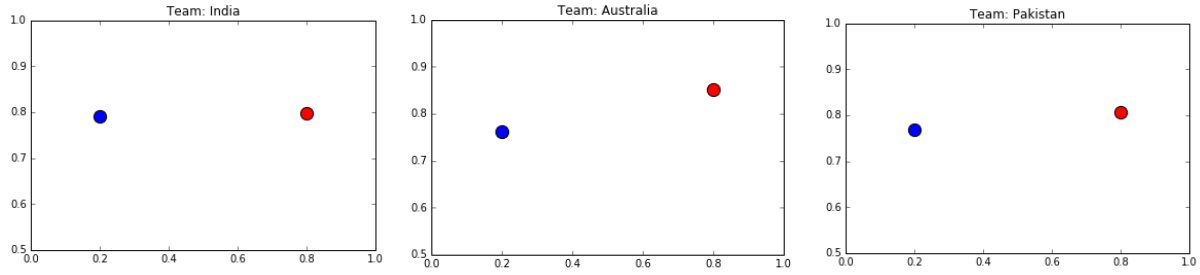

Fig 5 Recall-Precision Score Value for MLP Classifier

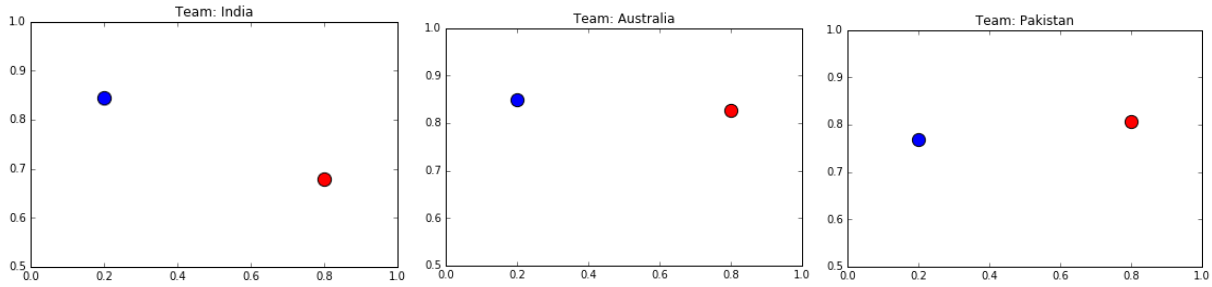

Fig 6 Recall-Precision Score Value for SVM Classifier

Table 4 Observed Value of Performance Measures

		India	Australia	Pakistan
DT 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
MLP Classifier	Recall Score	0.797	0.850	0.906
	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
SVM Classifier	Recall Score	0.797	0.850	0.806
	Precision Score	0.843	0.849	0.812
	F1 Score	0.752	0.837	0.789
	Average P Score	0.744	0.749	0.724

CHAPTER 4: Project Snapshots

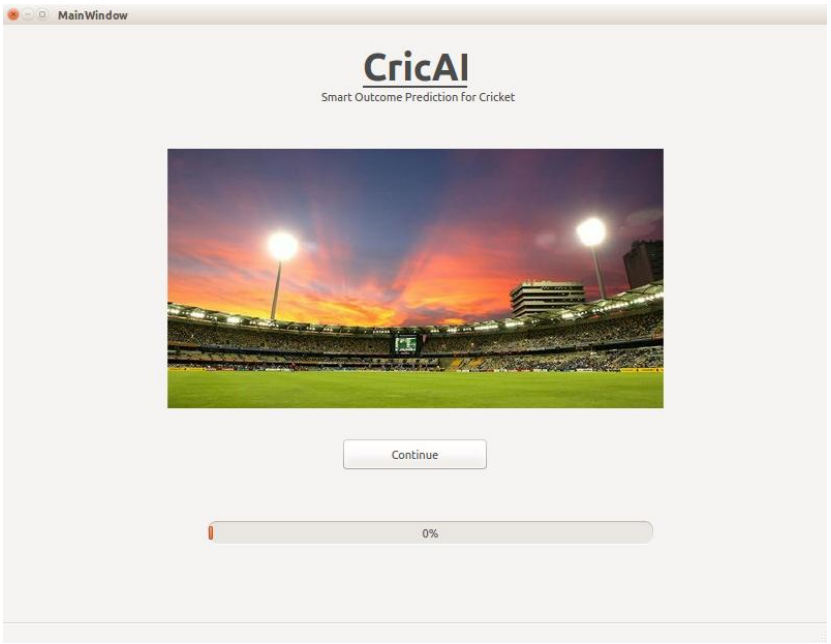

Fig 7 Welcome Window

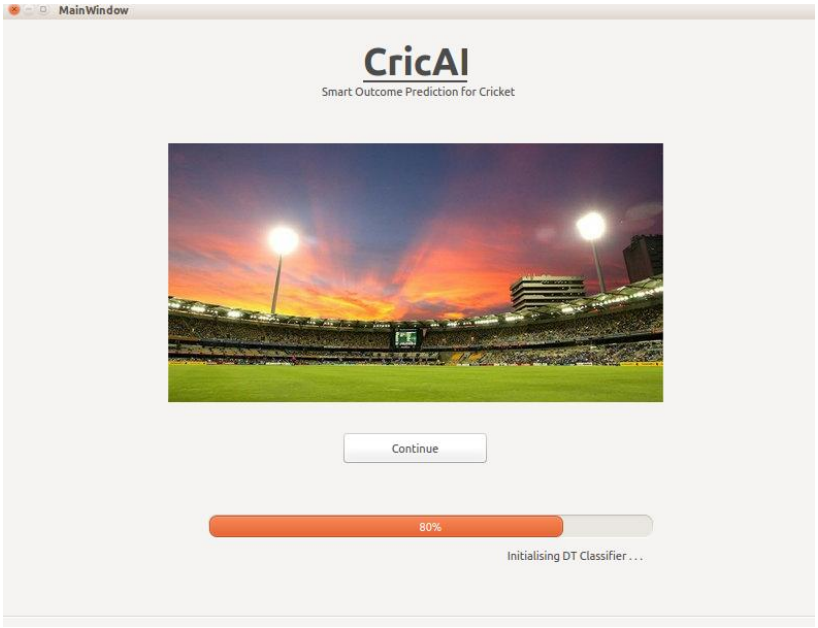

Fig. 8: Loading Window

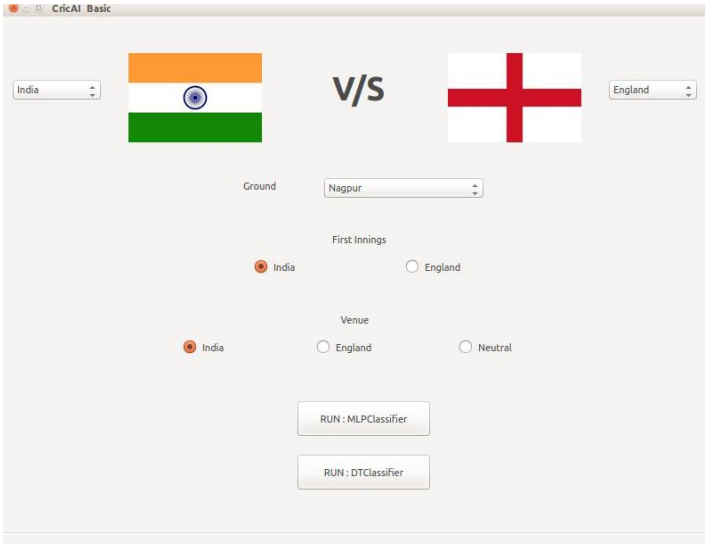

Fig 9(a) Game 1

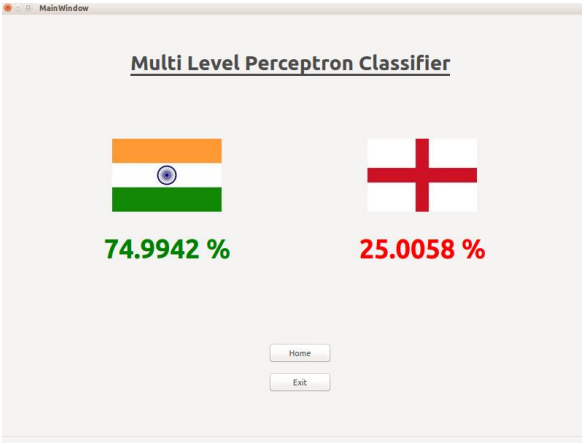

Fig 9(b) Game 1: MLP Result

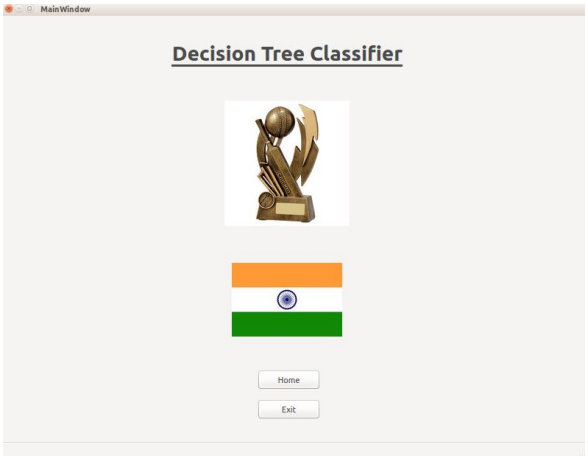

Fig 9(c) Game 1: DT Result

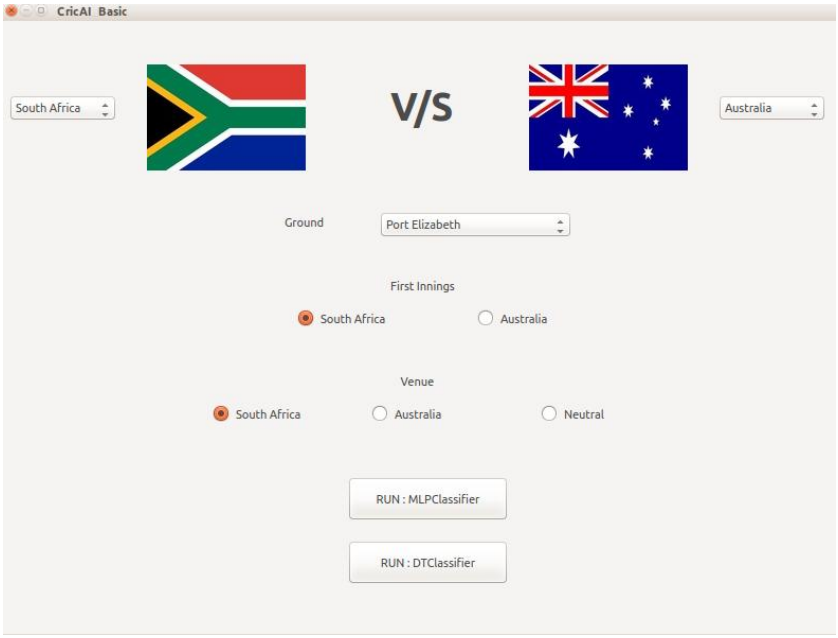

Fig 10(a) Game 2

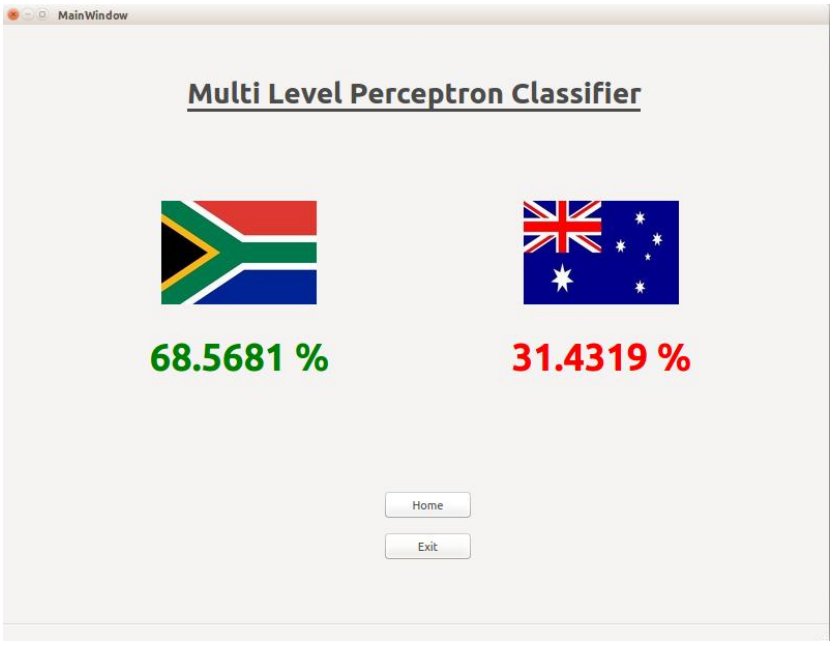

Fig 10(b) MLP Prediction

Prediction for Associate Member Teams

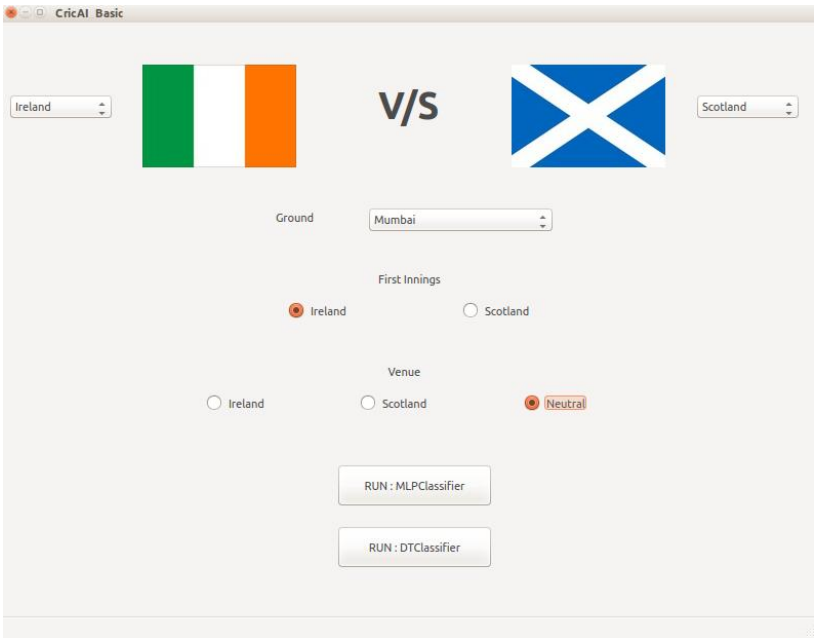

Fig 11(a) Game 3

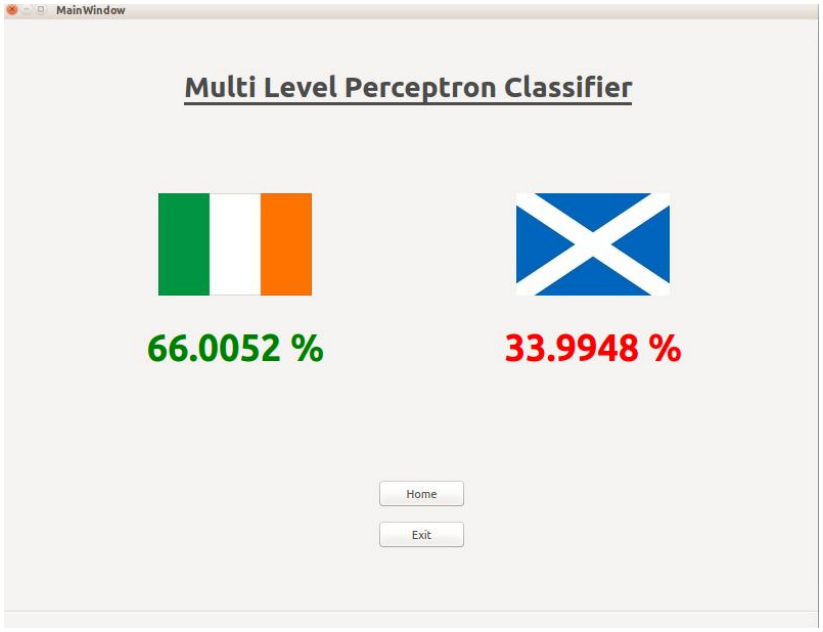

Fig 11(b) MLP Prediction

Capturing Home Advantage

Bangladesh V/S India, Home (Bangladesh)

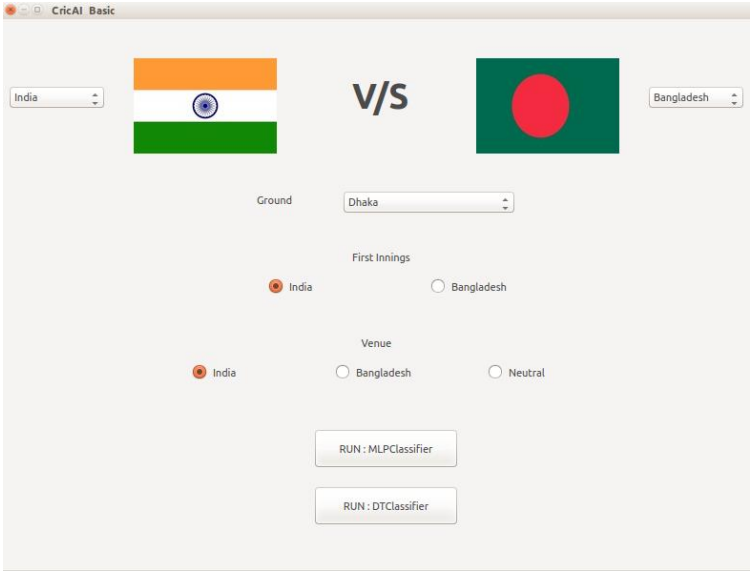

Fig 12(a) Game 4

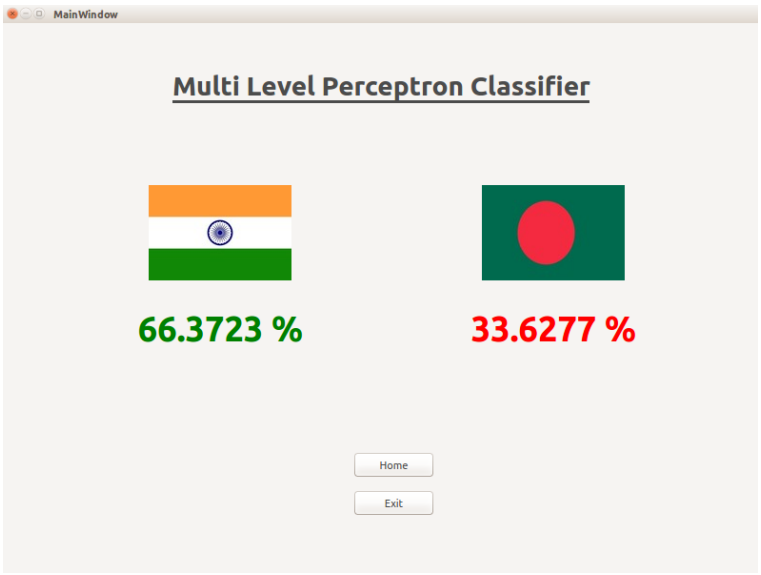

Fig 12(b) Game 4: MLP Prediction

Bangladesh V/S India, Home (India)

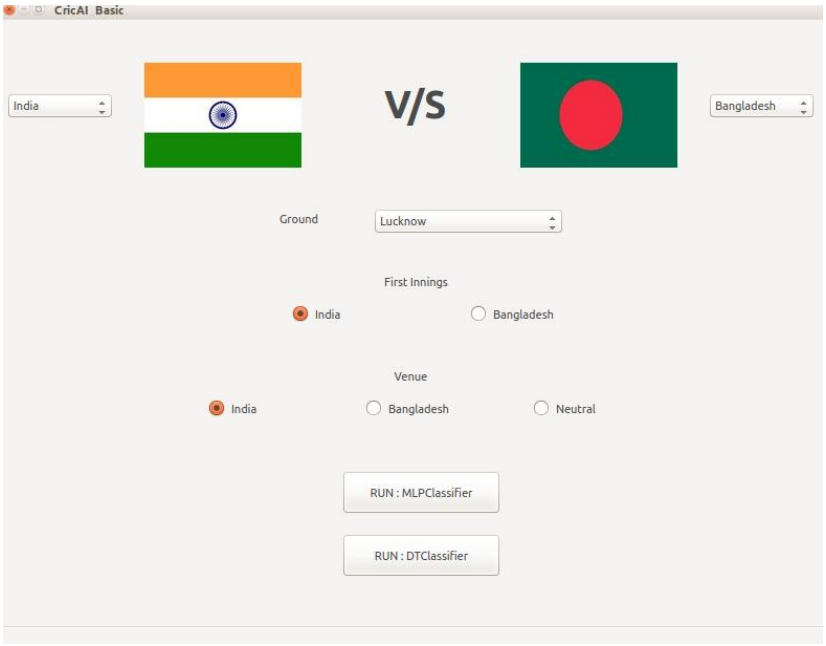

Fig 12(c) Game 4

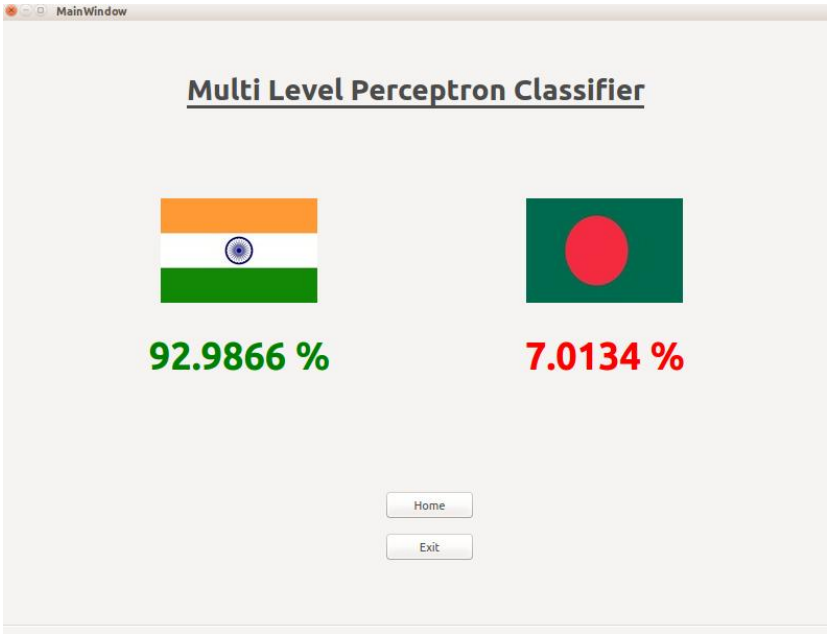

Fig 12(d) Game 4: MLP Result

Capturing Historic Track Records

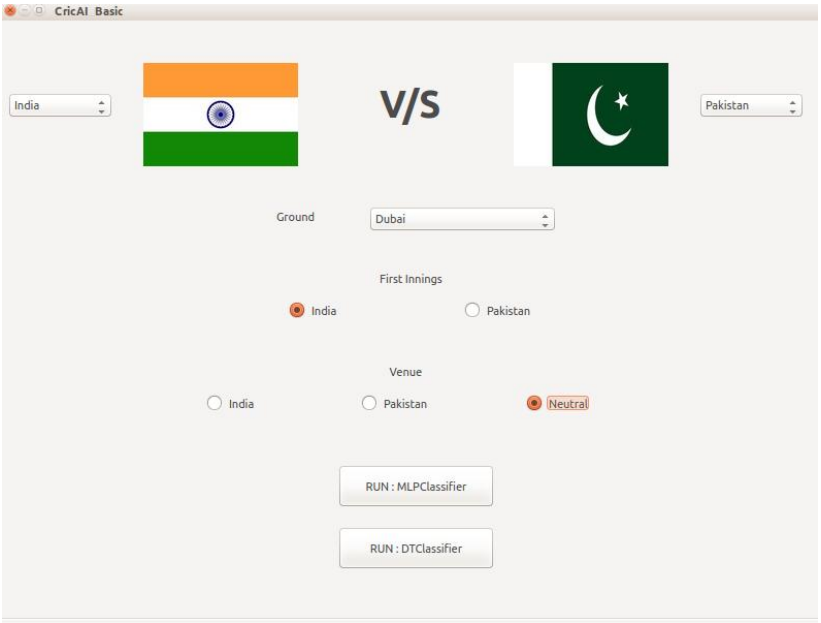

Fig 13(a) Game 5

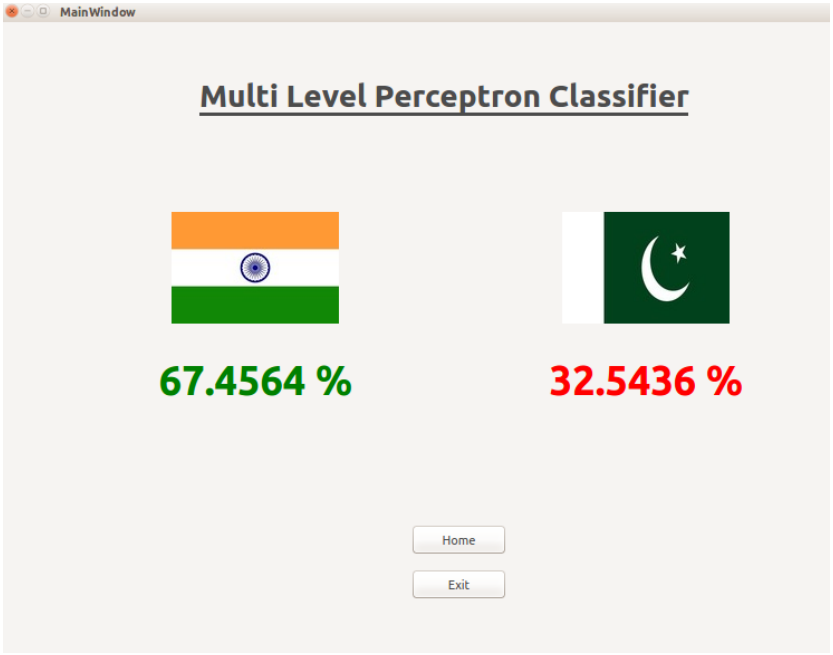

Fig 13(b) Game 5: MLP Prediction

CHAPTER 5: Limitation of Project

The prediction model is completely based on Historic data comprising of past track records between teams and at certain venue. We have used 2 pre-game attributes viz. Venue & Innings, to model our classifiers in more justifies manner. But what lacks in our model approach is to team combination perspective.

Also, we need to check the relevance of giving equal priority to the 1970s data as given to the 2017's match data. Since, many teams have evolved later in time so its bit unjustified on the way to assume such a case.

We also need to incorporate the team formation, player's form and the role of support staff in getting more better results. Last, but not the least Everything is on Web & Mobile. So, we need to build a web-app or a mobile-app instead of the native desktop-app to make the tool more accessible & approachable.

CHAPTER 6: Tech Stack

1. Data Set:

1.1. Pandas:

1.2. Beautiful Soup:

1.3. UrlLib

2. Project Development:

2.1. Scikit-learn

2.2. Jupyter IPython Notebook

2.3. Git & GitHub

3. UI Development:

3.1. PyQT5

Python was used as the main scripting language & all the classifier models along with project directory were built using Object Oriented Paradigm of python. All the scripts for dataset preprocessing were written on python only.

Python 3: Classifier Models & Main Module

Python 2: Dataset preprocessing scripts

CHAPTER 7: Related Works

From our literature survey, we found that very limited machine learning work has been done on game of cricket. Though cricket shares some attributes with other sports such as baseball, it still remains unique in certain respects and deserves to be analyzed independently.

Most of analyzing studies on cricket so far have been conducted using statistical methods. Bailey and Clarke conducted a study to predict the outcome in one day international cricket while the game is in progress [5].

WASP (Winning and Score Predictor), 2012 is product of some extensive research of Dr. Scott Brooker and Dr. Seamus Hogan at University of Canterbury in New Zealand. The WASP System is grounded on the theory of Dynamic Programming.

Neeraj Pathak & Hardik Wadhwa conducted a similar comparative analysis of a 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].

Parag Shah, Mitesh Shah [8] and Amal Kaluarachchi, Aparna S. Varde [9] explored the statistical significance for a range of variables that could explain the outcome of an ODI cricket match. In particular, home field advantage, game plan (batting first or fielding first), match type (day or day & night), past performance of team were the key interests in their investigation.

Madan Gopal Jhanwar and Vikram Pudi embarked on predicting the outcome of a One Day International (ODI) cricket match using a supervised learning approach from a team composition perspective [10]. Their work suggests that the relative team strength between the competing teams forms a distinctive feature for predicting the winner. Swetha and Saravanan.KN analysed the factors that cricket game depends on and decides Winning [1].

CHAPTER 8: Conclusion and Future Scopes

In our study, we performed a comparative analysis of the predictions generated by 3 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.
- Analysis of factors that affect the outcome of the game.
- 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 relevancy of considering 1980s match data equivalent to the 2017s match data also needs to be analyzed and worked upon. It is also possible to apply the machine learning techniques we used in our study for predicting the outcomes of other games such as hockey and soccer.

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.

[11] Wikipedia Foundation https://en.wikipedia.org

[12] Medium.com https://medium.com/machine-learning-101/