ODI Cricket Match Winning Prediction Using Data Mining Techniques

M.Sc. in Information Technology

2018

ODI Cricket Match Winning Prediction Using Data Mining Techniques

Subasingha SADP

158777V

Faculty of Information Technology

University of Moratuwa

June 2018

ODI Cricket Match Winning Prediction Using Data Mining Techniques

Subasingha SADP

158777V

Dissertation submitted to the Faculty of Information Technology, University of Moratuwa, Sri Lanka for the fulfillment of the requirements of Degree of Master of Science in Information Technology.

June 2018

Declaration

We declare that is our own work and has not been submitted in any form for another degree or diploma at any university or other institution of tertiary education. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

Name of Student	Signature of Student
S.A.D.P Subasingha	……
	Date: ……..

Supervised by

Name of Supervisor	Signature of Supervisor
S.C. Premarathne	…….
	Date: ……...

Acknowledgements

First and foremost, I would like to express my sincere gratitude towards my supervisor, Mr. Saminda Premarathne, Senior Lecturer, Faculty of Information Technology, University of Moratuwa for his guidance, supervision, advices and sparing valuable time thorough the research project.

Furthermore, big thank should goes to Prof. Asoka Karunananda who taught Research Methodology and Literature Review and thesis writing subjects which were the basis for this research work.

I would also like to thank all the batch mates of the M.Sc. in IT degree program who gave their valuable feedbacks to improve the results of the research and my family for the support they provided me through my entire life and in particular. I must acknowledge my wife encouragement and editing assistance, I would not have finished this thesis.

Abstract

Cricket is one of the most popular games in many countries. More than 19 countries playing cricket as main game & population for cricket will increasing gradually. However there are no proper tool for analyzing pre outcome of the match stating to end, and available tools are not support to simulate match using batting partnerships. The ultimate goal of predicting pre outcome of cricket match is to identify key players and their batting performances also prevent wrong players selecting and toss decision by making statistical predictions.

This project is focusing on One Day International (ODI) cricket match and predicts the outcome of the particular match. Our proposed solution consists of three major modules namely; Web UI Module, CRIC-Win Analytic Engine and Backend Data Module. CRIC-Win Analytic Engine has two sub data models, one for predict overall match outcome based on given pre match data and next for predicting match outcome based on batting partnership both home team and opponent team. All sub models in the CRIC-Win Analytic Engine are developed base on Naïve Bayes algorithm and use for generating the classifier model which can be used to predict the outcome of the cricket match.

Mainly prediction result divided into two segments. First predict overall pre outcome of the match based on given details and next predict how each partnership will affect to win the match. This work suggests that the relative team strength between the competing teams forms a distinctive feature for predicting the winner. Modeling the team strength boils down to modeling individual players batting partnerships and bowling performances forming the basis of our approach. We use partnership statistics as well as the recent performances of home team & opponent team, partnership records, wicket falling pattern, remaining overs, required runs, toss decision, ground and day/night effects have also been considered in order to predict the outcome of a match. The solution can be simulated match outcome before the match start and while playing. Also this research describes in detail the different attribute selection techniques as well as the data mining algorithms used to solve this problem of outcome prediction in cricket. We have also used accuracy as the evaluation criteria to evaluate how well the prediction performs.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Prolegomena

This chapter gives an introduction to overall project on Data Mining solution for the cricket match wining prediction. Cricket is a bat and ball team sport first documented as being played in southern England in the 16th century. By the end of the 18th century, cricket had developed to the point where it had become the national sport of England. The expansion of the British Empire led to cricket being played overseas and by the mid-19th century the first international matches were being held. Today, the sport is played more than 100 countries and become very popular game every knock and corner of the world.[1]. Currently, in one Day International (ODI) cricket matches first innings score is predict on the basis of Current Run Rate which can be calculated as the amount of runs scored per the number of overs bowled. Furthermore, in second innings used required run rate but there is no proper method to predict overall outcome of the match.[2]

There have number of score prediction tools and most of them are considered only current run rate and required run rate but these factors are not enough to simulate overall outcome in cricket match for instance partnership between wickets, number of wickets fallen, the venue and the batting team itself and toss decision should be considered when predict outcome. This project is focusing on sports sector and try give much more beneficial to sports lovers and people who are involving selecting best players etc. Because so far no one focused on how to use data mining technology to predict pre outcome of the cricket match by simulation match based on given situation.

1.2 Problem Statement

Despite the popularity of sports, prediction in cricket has not been addressed in great detail as in other sports like baseball, basketball, football and soccer etc. Since we have toss decisions, batting records, bowling records, individual player records, scorecard of different matches played and so on, no one considered how these data put into data mining model to predict the results of games and it can be used to predict some strategic decisions to win the games.

1.3 Aim and Objectives

This research aims to predict the outcome of the ODI Cricket game using data mining techniques with comprehensive statistical data and other game related data and finally implement web based tool to visualized predicted outcome.

With the view to develop a CRIC-Win solution for ODI Cricket match domain, we identify the following five objectives.

1. Critical review of data mining technology in cricket sport.
2. In depth study of classify data mining technology.

3. Develop web based tool for predicting pre outcome of the match by simulate game based on given situation.
4. Evaluate the proposed system
5. Develop a document (thesis/ dissertation/ report) of the project

1.4 Background and Motivation

Today's world, sports are intensely competitive propositions. Motivated by huge financial rewards, sports professionals are engaged in huge competition always try to take advantages over their opponents. Currently, sports professionals include not only the sportsmen actively participating in the game, but also their coaches, trainers, physiotherapists, and in many cases, strategists. Coaches, captains and team managers leverage their expertise and make decision using their intuition. Such decisions can be biased by the human impressions and judgments of players and hence might overlook players' weakness. Moreover, interesting patterns in the game may hide the eye of the best tactician [15].

Effective formulation of strategies requires carrying out extensive analysis of past games, current performance in the game in progress, and numerous other factors affecting a game. Players and team management (collectively often referred to as the team think-tank in sports) perform as a "human expert system", relying on experiences, expertise and analytic ability to arrive at the best-possible course of action before as well as during a game. Vast amount of raw data and statistics are available to aid in the decision-making process, but determining what it takes to win a game is extremely challenging [15]. The trend noticeable both in individual sport such as tennis and in team sports such as baseball and basketball is that this knowledge is used to determine pre-game strategy. Successful application of this pre-determined strategy often becomes the important factor towards the victory [16].

Cricket is the second most popular sports in the world with billions of fans across India, Sri Lanka, England, Pakistan, Africa, Australia, etc. It is an outdoor game played on a cricket field at 22-yard rectangular long pitch, between two teams consisting each of 11 players. It is played in three formats namely Test, One Day International (ODI) and Twenty over International (T20). In ODI each team takes its chance to bat, trying to score as many amount of runs which can be scored in 50 overs while the other team fields for that much amount of overs. Each chance is termed as an innings [1].

Unlike other sports, cricket stadium's size and shape is not fixed except the dimensions of the pitch and inner circle which are 22 yards and 30 yards respectively. The cricket rules do not mention the size and the shape of the field of the stadium [3]. Pitch and outfield variations can have a substantiate effect on batting and bowling. The bounce, seam movement and spin of the ball depends on the nature of the pitch. The game is also affected by the atmospheric conditions such as altitude and weather. A unique set of playing conditions are created due to these physical differences at each venue. Depending on these set of variations a particular venue may be a batsman friendly or a bowler friendly [2].

Currently, in cricket match the projected scores can be seen displayed at the score card during the first innings, which is basically the final score of the batting team at the end of that innings if it scores according to the current run rate or a particular rate. Run rate is defined as the amount of runs scored per the number of overs bowled. However, run rate is considered as the only criteria for calculating the final score. But there are other factors too which may affect the final score like number of wickets fallen, the venue and the batting team itself [3].

Match data since the beginning of the ODI game is available. However our literature search found no proper tool for predicting pre match outcome and simulating game in given situation for make decisions. Some related work could be found on topic of "A Classification Based Tool to Predict the Outcome in ODI Cricket" by Amal Kaluarachchi and Aparna S. Varde [1]. They used Naïve Bayes algorithm to predict outcome but they have considered only basic factors and failed to simulate game for making strategic decisions. So, this research aims to study the problem of predicting the game results before the game start, based on the statistics and data available from the data set and simulate pre outcome based on given situation to make decisions for win the game.

1.5 Problem in Brief

Since we have toss decisions, batting records, bowling records, individual player records, and scorecard of different matches played and so on, no proper involvement in cricket how those historical matches' data used for predicting outcome of the match.

1.6 Proposed Solution

We proposed web based system that takes in historical match data to predict win or loss and provide facility to user simulate the game for making some strategic decisions to win the game. We model the game using a subset of match parameters, using a Naïve Bayes classification algorithms. It also suggests partnership performances along with their preferred roles in the match.

1.7 Structure of the Thesis

The overall thesis is structured as follows, First chapter gives an introduction to fill project with the objectives, background, problem, and solution and second chapter critically reviews the literature in the data mining technology in cricket sports with a special reference to classification technique. Third chapter is about details of data mining technology by showing it's relevant to ODI cricket match and forth chapter present our approach with users, inputs, outputs, process and features. Fifth chapter is the design of the CRIC-Win solution, while sixth chapter implementation of the solution. Seventh chapter reports on the evaluation of the solution. Finally, chapter 8 concludes the solution with a note on further work.

Chapter 2

Review of Using Data Mining Techniques for Wining Prediction in Cricket

2.1 Introduction

This chapter critically reviews the use of data mining technique for score and wining prediction in cricket. In this sense, first we discuss concepts and general usage of data mining techniques in most popular sports such as football, cricket [16],[1] and soccer [3], subsequently, specifications of data mining sports field. Then we identified unsolved issues and concerns in the data mining techniques in wining prediction in cricket. Finally we define our research problem to be addressed in the thesis. This chapter also identifies the possible data mining techniques to be used for solving the problem. The chapter is organized under the heading so use of data mining in various sports, Summary of challenges and problem definition.

2.2 Data Mining Techniques

Data mining involves several disciplines and approaches, based on various tasks; data mining can be classified into Association, Classification, Clustering, Predictions, Sequential Patterns, and Similar Time Sequences. Depend on different explore methods; data mining can be generally divided into machine learning, statistics, neutral network and database. In machine learning, it can be divided more detail, such as inductive learning, case-based learning and genetic algorithm, etc. In statics, it can be divided more detail into regression analysis, clustering, and discriminant analysis and so on; for the neural network methods, it can be divided into self-organizing neural networks and feed-forward Neural Networks. The main method in database is Multidimensional data analysis and On Line Analytical Processing. Following Figure shows distribution of data mining techniques.

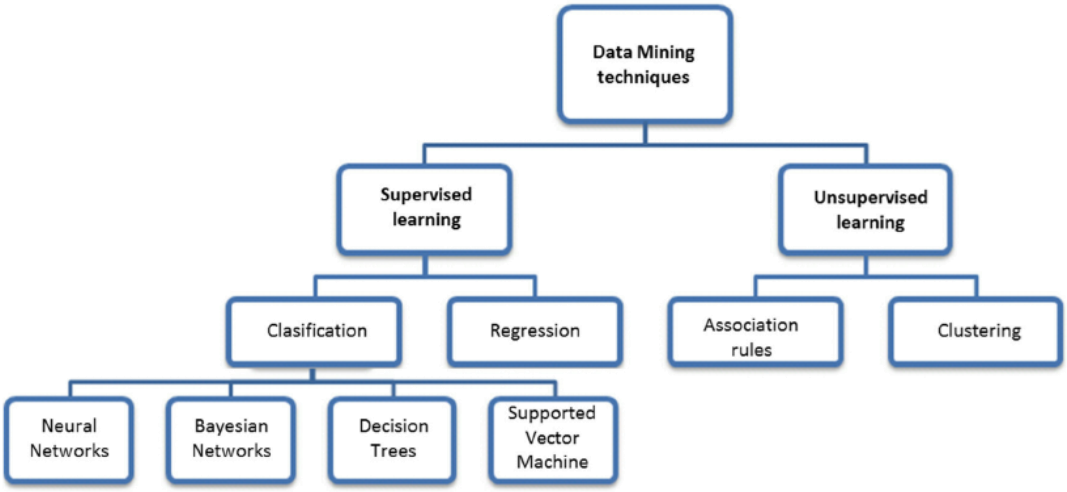


Figure 2.1 Data Mining Techniques

There is no data mining method can cope with all the requirements. For a particular problem, the characteristics of the data itself will affect the choice of tools. The following section will generally introduce several basic methods which involves in classification techniques.

- **Decision Tree**

Decision tree is a method for classification by modeling a tree structure model with leaves representing class labels and branches representing conjunctions of features. Then method is known as "divide and conquer". The output of the learning process is a classification tree where the split at each node of the tree represents one if -then decision rule and each leaf correspond to one value of the target variable. Given an example its target could be predicted by starting from the root and going down to a leaf of the decision tree by matching the variables (features) of the example with the splitting conditions at each node. The training algorithm chooses at each step the best variable to split the set of training examples. The criterion to compare between variables is how well the variable split the set of training examples into homogeneous subsets of examples with respect to the values of the target variables. Examples of the splitting criterions used to choose a variable are the Gini Impurity and Information Gain. The popular application of decision tree in CRM domain is customer classification. Decision tree is a very easy model and can be understand by non-professional people. It is such a simple model that it may not perform well on complex classification problems.

- **Logistic Regression**

Logistic regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes).

In logistic regression, the dependent variable is binary or dichotomous, i.e. it only contains data coded as 1 (TRUE, success, pregnant, etc.) or 0 (FALSE, failure, non-pregnant, etc.).

The goal of logistic regression is to find the best fitting (yet biologically reasonable) model to describe the relationship between the dichotomous characteristic of interest (dependent variable = response or outcome variable) and a set of independent (predictor or explanatory) variables. Logistic regression generates the coefficients (and its standard errors and significance levels) of a formula to predict a *logic transformation* of the probability of presence of the characteristic of interest:

$$logit(p) = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + \ldots + b_kX_k$$

Where p is the probability of presence of the characteristic of interest. The logic transformation is defined as the logged odds:

$$odds = \frac{p}{1-p} = \frac{probability\ of\ presence\ of\ characteristic}{probability\ of\ absence\ of\ characteristic}$$

And

$$logit(p) = \ln\left(\frac{p}{1-p}\right)$$

Rather than choosing parameters that minimize the sum of squared errors (like in ordinary regression), estimation in logistic regression chooses parameters that maximize the likelihood of observing the sample values [22].

- **Naïve Bayes**

The Naive Bayesian classifier is based on Bayes' theorem with the independence assumptions between predictors. A Naive Bayesian model is easy to build, with no complicated iterative parameter estimation which makes it particularly useful for very large datasets. Despite its simplicity, the Naive Bayesian classifier often does surprisingly well and is widely used because it often outperforms more sophisticated classification methods.

If the NB conditional independence assumption actually holds, a Naive Bayes classifier will converge quicker than discriminative models like logistic regression, so we need less training data. And even if the NB assumption doesn't hold, a NB classifier still often does a great job in practice. A good bet if want something fast and easy that performs pretty well. The main disadvantage is that it can't learn interactions between features. As an example it can't learn that although you love movies with Brad Pitt and Tom Cruise, you hate movies where they're together.

There are lots of data mining techniques and for the key model such as Logistic Regression, Artificial Neural Networks and SVM. Navïe Bayes algorithm is used for this project will be discuss more detail in the Implementation chapter.

2.3 Data Mining Tools

The following section presents and introduces some popular data mining tool, and WEKA as the main tool for the experiment will be explain more detail.

- **RapidMiner**

RapidMiner builds a software platform for data science teams that unites data prep, machine learning, and predictive model deployment. Organizations can build machine learning models and put them into production faster than ever, using RapidMiner's lightning fast visual workflow designer and automated modeling capabilities. RapidMiner eliminates the complexities of cutting edge data science by making it easy to use the latest machine learning algorithms and technologies like Hadoop, and Spark [23].

- **Azure Machine Learning Studio**

Azure Machine Learning Studio gives an interactive, visual workspace to easily build, test, and iterate on a predictive analysis model. We can drag-and-drop datasets and analysis modules onto an interactive canvas, connecting them together to form an experiment, which we run in Machine Learning Studio. To iterate on our data model design, we edit the experiment, save a copy if desired, and run it again. When we are ready, we can convert our training experiment to a predictive experiment, and then

publish it as a web service so that our model can be accessed by others. The main advantage of the Azure Machine Learning Studio is that, there is no programming required, just visually connecting datasets and modules to construct our predictive analysis model. This tool will suit for large business organizations because it may charge huge cost [24].

- **WEKA**

Weka is a data mining tool which integrates several machine-learning tools within a common framework and a uniform GUI. Classification and summarization are the main data-mining tasks supported by the Weka system. Users can use GUI or their own Java consuming Weka‟s API to perform machine learning tasks directly. Weka has the function for data pre-processing, classification, regression, clustering, association rules, and visualization. Weka with GUI is chosen as the tool for the model fitting process of this project. Because Weka provides all functions required by this project, including data preprocessing, all classification models, and result analysis tool. Weka with GUI also provide Knowledge Flow tool, which can help user to manage their model fitting workflow [17]. Weka's powerful functionalities, open source and intuitive user interface are the major factor that we choose this tool.

2.4 Data Mining Process

The data mining process can be generally divided into the following phases and the figure below shows the whole process: The data mining process can be generally divided into the following phases and the figure 2.2 below shows the whole process.

- **Problem Definition**

A data mining project starts with a correct understanding of the business problem. Here the understanding can be explained into the project objectives and the requirements from a business perspective. The project objective is then translated into a data mining problem definition and will give a direction for the following work. In the problem definition phase, data mining tools are not yet required.

- **Data collection pre-processing**

Data collection is to acquire the data; it can be either extremely simple or very complicated. Obtain data can be either automatically or manually. These processes include: data selection, data pre-processing and data conversion. The purpose of data selection is to determine the related objects involved in data mining tasks, according to the specific requirements of the data mining task, extracted from the relevant data sources and mining related data sets. The data pre-processing usually consists of the elimination of noisy data; handling missing data; eliminate duplicate data and data type conversion processing. The main purpose of the data conversion is to reduce the data set and the feature dimension (referred to as dimensionality reduction), Preparing the data for the modelling tool by selecting tables, records, and attributes, are typical tasks in this phase. The meaning of the data is not changed. Filtering the real feature which related to the data mining tasks in order to improve the efficiency of data mining.

- **Modelling**

There are many data mining functions can be used to solve different type of problems. In this phase, through applying and selected various modelling techniques several times to calibrate parameters into an optimal state until best values are achieved. When the final modelling phase is completed, a model of high quality has been built.

- **Evaluation**

Evaluating the model mean to estimate the model whether satisfy the expectations or not. If the model does not fit the original expectations, they go back to the modelling phase and rebuild the model by changing its parameters until optimal values are achieved. When the models are finally satisfied with the targets, they can extract business explanations and evaluate the questions like: Whether the model fit the business objective or not? Did all business factors be considered? Then, how to take advantage of the data mining results?

- **Deployment**

In this step will involve deploy plan, monitor and maintenance plan, finally express the results and understand the results. Results can be exported into database tables or into other applications, for example, spreadsheets and also can be display by visualization technique.

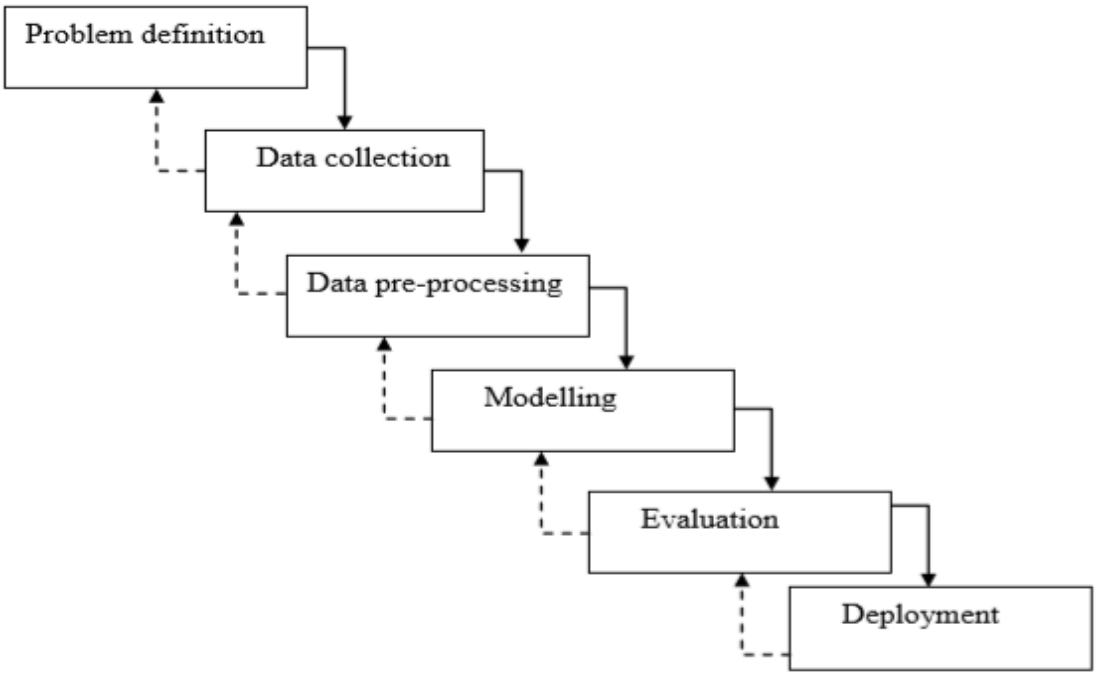


Figure 2.2 Data Mining Process

2.5 Use of Data mining in sports

The problem of match outcome prediction has been studied extensively in the context of baseball, basketball and soccer. Bhandari et al. [7] developed the Advanced Scout system for discovering interesting patterns from basketball games, which has is now used by the NBA teams. More recently, Schultz [8] studies how to determine types and combination of players most relevant to winning matches. In soccer, Luckner et al. [10] predict the outcome of FIFA World Cup 2006 matches using live Prediction Markets. In baseball, Gartheepan et al. [9] built a data driven model that helps in deciding when to 'pull a starting pitcher'

In 2015 Khabir Uddin Mughal and P. Bhatia [11] has been proposed that two methods, first predicts the score of first innings on the current run rate, considering number of wickets fallen, venue of the match and batting team. The second innings considering the same attributes as of the first method along with the target given to the batting team. Almost previous works[4] are worked in statistically predicting such as (average predicting techniques)[5] the scores or the outcome of the match. But this research use two separate models one for the first innings and other for the second innings. Also use linear regression classifier predictions for current data and Naïve Bayes classifier on the past records. They observed that the error in linear regression classifier is less than the current run rates other method in predicting the final score. Also accuracy of the Naïve Bayes is goes from higher percentage. However further research would be conducted to improve the accuracy of both models.

Also in 2016 Anik Shah, Dhaval Jha and Jaladhi Vyas introduced tool that Wining and score predictor (WASP)[7] is a calculation tool used in cricket to predict the score and possible results of a limited over match format. According to analytics provided throughout this report, it can be derived that, there are loopholes in the existing method and it needs to be resolved for better judgment. Also, different parameters analyzed during the new method suggest that it is an improved method than before and can be implemented at a higher platform like international cricket with some changes being done.

Furthermore in 2014 V. V. Sankaranarayanan, J. Sattar, and L. V. Lakshmanan developed separate models[4] for finding home runs and none-home runs using historical features as well as instantaneous match features from past games. Also it demonstrated the quality and accuracy of their predictions with an extensive set of experiments on real ODI cricket data[2]. In addition to that predicting runs for future segments. They proved that winner prediction accuracy is the highest reported in ODI cricket mining literature. However future research should be conducted to predict fall of wickets and improve the prediction accuracy even further.

Rizwan khlaiq khan , Irian Manarvr and Mohay-ud-dhr has been established that data mining[6] could be successfully used to evaluate the past performance of the any cricket team such as data is continuously being collected all around the world. It analysis could be used successfully or making future strategies by team managers and players against other teams. It may also be conclude from the performance of New Zealand's team could perform much better at home grounds even against best teams of the world because of the support it gets from the crowd as well as having more practice on same ground. However this research not clearly mentioned how they

analyzed data only represents previous data. So future research should be conducted how to construct good modal to analyze cricket match records.

In the recent past (in the last three decades/Since 1980s) Data mining has shown a multifaceted developments (growing interest) in many disciplines. Undisputedly R.P Singh & Rizwan khlaiq have produced the first reported web tool to predict score and wining prediction called WASP [5] (wining and score predictor) in 2016. Subsequently numerous applications in sports wining prediction were developed by many researchers. For instance (example), Brown and others [6] develop some model for predicting wining prediction in cricket match but considering only few affecting factors. Such as run rate, pitch condition etc. However there are much more affecting factors to be considered to prediction about cricket match. Because new rules and regulations are introduced by international cricket council called ICC. It should be noted that these are rather industry-based and complex in nature.

In contrast data mining techniques have also been used to predict cricket score and wining by many researchers. These researches are primarily targeted only predict projected score any how some research predict winning percentage also. But accuracy of prediction is very much low.

There is always a sport in everyone's life that takes over their lives and we don't mean that in a bad way. After being dwelled in in work all day the human mind really needs to rest and that is where it needs a sport. This sport shouldn't be just any other boring sport, it should be something that thrills you and excites us. Something that makes our heartbeat faster on every move. Such sports provide us an adrenaline rush which makes us forget about everything and focus on the game.

The one sport that can get us the adrenaline rush after football is none other than cricket! Cricket has been in our lives for quite some time now. Ranked as the second most watched sport in the world, cricket has its own fan following that goes lengths. With millions of followers from all over the world, the sport goes back to ages. From the rivalry between India Pakistan to the legendary matches between Australia and England, this is something that goes back ages and has been a part of everyone's life for a long time.

Match data since the beginning of the ODI game is available. Two team members are announced before match. Some works could be found on cricket match scoring rates by Clarke [12] and Preston and Thomas [13]. They used dynamic programming methods to predict scoring rates.

Also some studies, such as those conducted by De Silva [14] analyze the magnitude of the victory. It is found that most of these studies describe the factors with the goal of predicting the probability of victory. In the real world scenario, however there are cases where the magnitude of the victory is important especially when betting is involved.

There are different ways to do the prediction. The prediction can be done taking into consideration the player's performance as well as the team performance. There are many unpredictable things that happen in a cricket game like matches being washed out due to rain, a key player getting injured before the game, players changing their

teams, etc. Sometimes a key player also gets inured during the game and hence is not able to take further part in the game. All these factors do affect the prediction to some extent. The report discusses a methodology that I followed for the game result prediction. The methodology consists of first the attribute selection algorithms which trim down the list of attributes to only important ones and then the data mining algorithms which can be applied on those attributes. The game prediction problem that our study does not take into consideration the player's performance but it does take into consideration the team's past performance at a high level extent along with the other factors like toss winner, toss decision, home support, etc. The attribute selection techniques consist of the wrapper method and the ranker method. The data mining algorithms that are used are Decision Tree (J48), Random Forest, Naïve Bayes, K-Nearest Neighbor. The data mining tool used in the project is WEKA and it is a freely available data mining tool which has good support for a number of different data mining algorithms.

2.6 Sports Data Mining Applications

Currently, sports data mining tools as the derivatives of the data mining techniques have been emerged in a large number, players, coaches and rivals can get a better understanding of their competitive level by using sports data mining tools. So a new industry is rising which takes applying data mining to sports for commercial as purpose. The following session will introduce some popular sports data mining tools:

- **Advanced Scout**

IBM developed Advanced Scout in the mid-1990s as a data mining tool used for National Basketball Association (NBA) data analysis. The application is specifically tailored for NBA coaches and statisticians to discover the hidden patterns or features in basketball data, which provides a new insight by using the business intelligence and data mining technique. There are two data sources for this tool, one came from a courtside collection system include the time stamped events data such as shots, rebounds, three goal, etc. The other source is the game tape includes game footage. This source can be kept by coaches to prepare for upcoming opponents as well as to check mistakes and improve effective [20].

- **Digital Scout**

Digital Scout is a software used for collecting and analyzing game-based statistic and tools for baseball, basketball, and football, etc. It also supports the function of producing reports. For instance, baseball hit charts, basketball shots charts and football formation strengths [21].

- **Synergy Online**

This product has the similar function with Advanced Scout that dedicates to basketball-based multimedia and contains an index of live video broadcasts as searchable media. Coaches, players and fans can query plays in real-time and receive constantly updating player statistics by using this software [20].

2.7 Summary of Challenges

Our discussion of previous section has identified large number of unsolved issues in using data mining technology and lack of proper predictions in cricket match rather than other type of games. Among others, the challenges in using data mining technology for ODI cricket strategy recommendations considered to further analyzed. It appears that all the applications are used run rate for wining prediction. The table of 2.1 shows the summary of challenges.

Product	Technology Used	Positive points	Negative points
A Data Mining Approach to ODI Cricket Simulation and Prediction	Linear regression and Nearest neighbor clustering algorithms.	Developed separate models for home runs and non-home runs using historical features.	Did not considered partnerships and fall of wickets just predict only win or not.
A Classification Based Tool to Predict the Outcome in ODI Cricket	Naïve Bayes classifier formulas with different combinations of attributes.	Considered different combination attribute toss, home, home away and develop java based standalone tool to predict winning team.	Lack of analysis previous match results and consider only basic factors for making prediction.
Evaluating Performance of Blackcaps of New Zealand vs. Global cricket teams	Use statistical year wise cricket data analyze using Weka.	Graphically represent country wise result and statics data for easy understanding.	Considered only country wise matches winning or not all other attributes was discarded.
Score and Winning Prediction in Cricket through Data Mining	Gaussian Naïve Bayes classifier algorithms using Weka.	Clearly show the accuracy of the Naïve Bayes Classifier at different range of overs.	Consider only very few data set. Predict winning percentage based current run rate and required run rate.

Table 2.1 Summary of Challenges

2.8 Problem Definition

As per literature all detailed information about match available on the internet but there is no any comprehensive analyzing tool to match wining prediction, predict what happened when particular wickets falling down this moment who are the key bowlers, key batsmen under pitch conditions etc.

So we try to develop web based software tool using data mining techniques to overcome above circumstances.

2.9 Summary

This chapter presented a comprehensive critical review use of data mining techniques with a specific reference to web data mining. We reported developments in data mining model in broad spectrum of disciplines including sports and some other industry in general and web data mining in particular. We defined the research problem and also identified the possible technology addressing the research problem. We also identified the possible technologies that can be used to address the research problem. Next chapter will discuss the technologies adapted for solving our problem.

Chapter 3

Technology Adapted in Winning Prediction Analysis

3.1 Introduction

In the previous chapter we discussed different findings in the area of winning predictions in sports, its developments, issues and future challenges and we define our research problem and also identified classification data mining as the technology to address the problem. This chapter highlights the effectiveness of selected technology that distinguishes it from the technologies applied in existing literature.

3.2 What is Data Mining?

Nowadays we come across with large, complex set of data which generated by computers, networks and humans. Government and private agencies, scientific institutes and business dedicated huge amount of resources to collect and keep data. However, only small amount of data is used because, in many cases volume is too large to manage, or the data structures themselves are too complicated to analyse effectively. Ability to extract useful knowledge hidden in these data and to act on that knowledge is becoming important in this competitive world. Therefore the data mining process is emerged. The entire process of applying a computer based methodologies including new techniques for discovering knowledge can be treated as "Data Mining".

Data mining is a process of analyzing big data from different perspectives and summarizing them to useful information by the means of different techniques. Formally this is defined as non-trivial process of identifying valid novel, potentially useful and ultimately understandable patterns in data. The overall process of finding useful knowledge in raw data involves the sequential line up of steps such as developing an understanding of the application domain, creating a target data set based on an intelligent way of selecting data by focusing on a subset of variable or data samples, data cleaning and preprocessing, data reduction and projection, choosing the data mining task, choosing the data mining algorithm, interpreting minded patterns and consolidating discovered knowledge. Hence, process of data mining can be consider as total solution which consist of phases business understanding and preparation, modeling, evaluation and deployments in interactive manner as in figure 3.1.

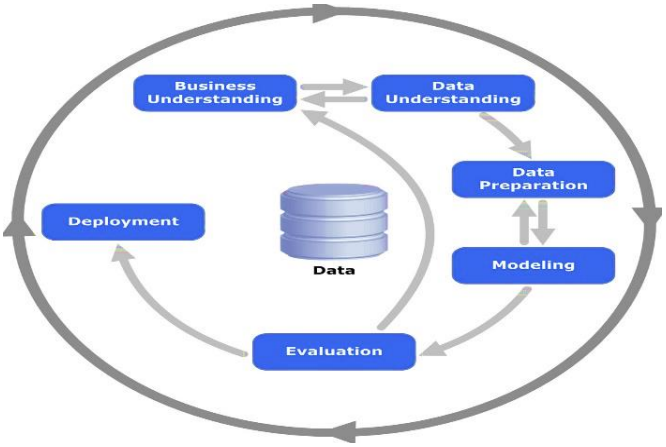


Figure 3.1 Standard Process for Data Mining

Data mining is having two primary goals of being predictive or descriptive. According to the task, different techniques are available in data mining. For predictive task, techniques such as classification, regression and deviation detection are used. Meanwhile techniques such as association rules, cluster analysis are used for descriptive tasks as in figure 3.2. Predictive algorithms determine models or rules to predict the values of variables when given input data. On the other hand descriptive algorithms determine models to summarize the data in some manner. Therefore selecting the most appropriate mining technique depends on the goal which users going to achieve.

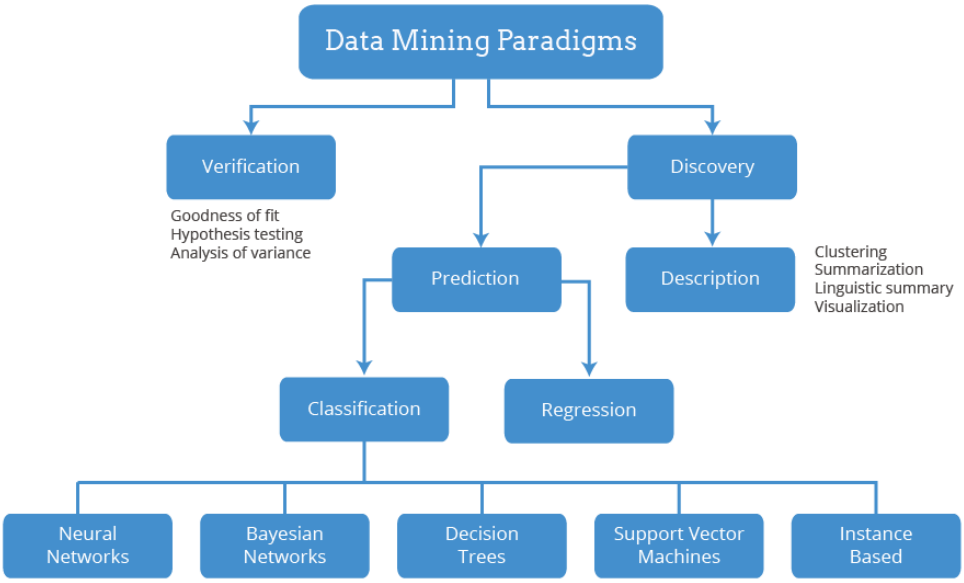


Figure 3.2 Data Mining Techniques

Researchers have identified data mining as interesting, beneficial subject area, because of successful knowledge discovery applications. When considering business applications of data mining, market basket analysis in marketing, fraud detection in finance, defect findings in manufacturing and governance can be identified as effective applications. In science, data mining achieve remarkable success in fields

such as telecommunication, astronomy, pattern discovery in biology. There are so many other application areas of data mining such as web content mining, web usage mining and stream data mining.

3.3 Reasons for Classification Data Mining Algorithms for ODI-Cricket Match Winning Prediction

We tried to use three machine learning techniques association rules mining, clustering and classification. Some selected algorithms of each machine learning techniques were trained by using WEKA tool. Captured data set was organized in Attribute Relation File Format (ARFF) as required for WEKA.

However, clustering and association algorithm did not make any contribution to our research. We try to predict win or loss probability and try to simulate game by given scenario. As we expected, classification algorithms presented significant results in our research. Because of our data set represent independent attributes so binary classification provide expected result.

In machine learning, Naïve Bayes and Decision Trees are two popular classification approaches. According to literature survey we found that other classification algorithms named Naïve Bayes, Decision Tree, AdaBoost and Bagging could be used to make prediction in various type of sports for instance Cricket, football, soccer and basketball. The results obtained were interesting in terms of improving the performance. So classification algorithms are the most suitable method to analyze the ODI Cricket Match Wining prediction.

3.4 Java

One of the most significant advantages of java is its ability to move easily from one computer system to another. The ability to run the same program on many different systems is crucial to World Wide Web software and Java succeeds at this by being platform-independent at both the source and binary levels. In addition to that, WEKA API is developed by using Java language. So Java is the most compatible language for accessing the WEKA API.

3.5 WEKA / WEKA API

Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes [17] and also WEKA can be used in any java code using WEKA API.

3.6 .NET Framework

.Net Framework is the software framework developed by Microsoft that runs on primarily on windows and recently announced that .Net core can runs on Mac and Linux other than Windows [19].The .NET Framework is language independent. This means that, as a developer, you can develop in one of the many languages that target the .NET Framework, such as C#, C++/CLI, Eiffel, F#, IronPython, IronRuby, PowerBuilder, Visual Basic, Visual COBOL, and Windows PowerShell [18]. So Front-End (Web UI) development can be used ASP.Net web technology.

3.7 MS SQL

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications which may run either on the same computer or on another computer across a network including the Internet.

Our Back-end module we have used SQL Server Express is the free version of Microsoft's primary relational database management system (RDBMS).

3.8 Summary

This chapter presented data mining as the technology proposed to analyze ODI-Cricket Match wining prediction and simulate the match according to given scenario for making strategic decisions to win the game. In this sense, Java and WEKA API can be used to data modeling then .Net Framework can be used for front end development also it is pointed out how the classification data mining algorithm offers and efficient and accurate solution for ODI Cricket wining prediction analysis. The next chapter shows a novel approach to predict outcome of the ODI Cricket match through technology presented here.

Chapter 4

A Novel Approach to Predict Outcome of the ODI Cricket Match

4.1 Introduction

Chapter three presented the technology to be used to solve the research problem. This chapter described our approach to address the problem of inadequate in evaluation outcome of the cricket match. CRIC-Win Analytic Engine is the key module of our solution. We present our approach by highlighting hypothesis, input, output, process, users and features of the CRIC-Win Predictor solution.

4.2 Hypothesis

We hypothesis that the issue of unavailability proper mechanism for cricket match wining prediction can be solved by introducing CRIC-Win Predictor tool. This hypothesis was influenced by the fact that freely available cricket data can be liable to make competitive advantages by simulating game in given situation.

The hypothesis of this research is that the wining prediction of a cricket match including simulation facility of the pre-outcome of the match in given situation can be achieved by using classifier analysis. We are going to use various classify technologies such as Naïve Bayes, Decision Tree, AdaBoost and Bagging on then finally picked up most accuracy classifying techniques based on data model.

4.3 Input

For conducting research, we collect last two year ODI cricket match data for selected teams including Sri Lanka, India, Bangladesh and Zimbabwe from the website www.crickinfo.com. The attributes selected were Team, Opponent team, Home/Away, Day/Night, Toss, Bat 1st and the result. Each team was analyzed individually against every other team. The reason for doing so is the following. When we have one data set for all matches, one match forms two records in the data set. For instance Sri Lanka plays against India. The data set is as shown in table below.

Sample Data Set of Two Teams which explains the over-fitting situation

Team	Toss	Bat 1st	Result
Sri Lanka	Win	Yes	Win
India	Lost	No	Lost

Table 4.1 Data Set Over-fitting Situation

This created over- fitting results with the given data set. The best way to avoid that was to select one team against other at a time.

There are lots of data mining techniques but we try to use three techniques: association rule mining, clustering and classification. Selected algorithms from each

technique were trained using WEKA. Data set was organized in ARFF file format as required by WEKA as represented in figure 4.1.

```
@relation CrickData
@attribute TeamID {10, 11, 12, 13….}
@attribute GroundID {1, 10, 12, 13……}
@attribute IsWonToss {True,False}
@attribute IsDayMatch {True,False}
@attribute InningNo {1, 2}
@attribute ParnershipPlayers {120-121,120-14,120-17,120-18…….}
@attribute WicketNo {1,10,2,3,4,5,6,7,8,9}
@attribute PartnershipRuns numeric
@attribute TotalRuns numeric
@attribute RemaningWickets {1,10,2,3,4,5,6,7,8,9}
@attribute RemainingOver numeric
@attribute CurrentRunRate numeric
@attribute Remainingrun numeric
@attribute RequiredRunRate numeric
@attribute Result {Won,Lost
```

Figure 4.1 ARFF Format Data Set

4.4 Output

As main output of this process will be given probability of match winning condition. Web based software tool that accept cricket match data then CRIC-Win Analytical Engine responsible to generate prediction result and next predicted result would be sent back to the UI. This tool is called CRIC-Win Predictor support to simulate the game in given situation to make strategic decisions and future development of cricket game. In summary, output of CRIC-Win Predictor tool includes following components.

- CRIC-Win Analytic Engine
- Web UI
- Back End Database

4.5 Process

Even the inputs our project generate the CRIC-Win Predictor tool. In order to generate predictions from the inputs we use Java with WEKA API and ASP.Net technology as the main technologies. The process of building the environment for generating cricket match wining probability we have special software called WEKA. They will provide good mechanism to make model for predicting match result and then our CRIC-Win analytic engine can act as a mediator between backend and the Web UI. It can route predicting result generated by WEKA to the appropriate modules.

4.6 Users

Several categories of users can be identified for CRIC-ESS. They are sport lovers, match analyzers, cricketers and researchers in sports. However all features of the CRIC-ESS can be accessed any one because this tool act as information hub so there are no any authentication restrictions rather than other users based system. CRIC-ESS will freely available for everyone who interested to cricket.

4.7 Features

The features or a nonfunctional requirement of CRIC-Win Predictor includes the following in a broader sense. Among other features, low coupling is the significant features of CRIC-Win Predictor. According to our design CRIC-Win Predictor tool is developed as separated modules likewise UI Module, CRIC Win Analytical Engine and Backend Module. So CRIC-Win Analytical Engine is separated java API Module as well as it can be act as individual component in future development. Rest of features can be listed as follows

- Light weight :
- Low cost
- Operator independence
- Higher performance
- Easy to learn
- Expandability

4.8 Summary

This chapter presented the machine learning approach for conducting our thesis. We discussed learning techniques including hypothesis, input, output, process, features and the users of CRIC-Win Predictor tool. We defined the research process and also identified the possible approach for addressing the research problem. Next chapter will present the design of our CRIC-Win Predictor solution.

Chapter 5

Design of the CRIC-Win Predictor Tool

5.1 Introduction

Chapter four presented the approach to develop a CRIC-WIN Predictor tool by using data mining technology. This chapter elaborates the overall picture of the proposed solution and the design of the solution with its module, their roles and interconnections. Our CRIC-Win Predictor tool mainly designs with three modules namely, User Interface, Data mining analytical Engine and Backend Database module. So this chapter described the role of each module, connections among each module within the top level architecture of CRIC-Win Predictor solution.

5.2 Top Level Architecture (Design) of CRIC-Win Predictor

In our design we have decided maintain a separate interface module so enable access from different sources of inputs. We have also included a database for data manipulation. The core of proposed solution is defined as the CRIC-Win Analytics Engine. The top level architecture of the CRIC-Win Predictor is shown in Figure 5.1.

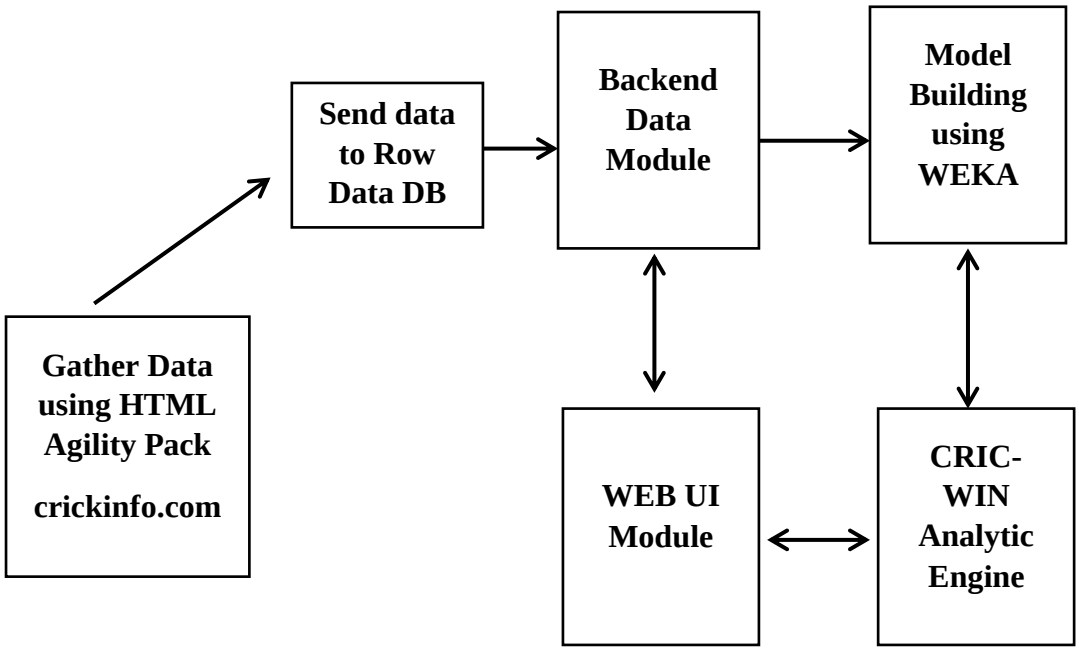


Figure 5.1 Top Level Architecture of CRIC-WIN Predictor solution

According to figure 5.1 we collect cricket match data using HTML Agility library. As we expected www.crickinfo.com sports web site have displayed all previous matches' data in html format so we have to scrap data and push all raw data to the separate database. We do this purposely to reduce the weight on main database. Then collected row data clean and send to the main backend data module for easy query manipulation. Then we build classifier models by using WEKA tool, and also we include testing and evaluation for model building. The core module of this solution is

that CRIC-Win Analytical Engine. Mainly it acts as a mediator when request come, it analyze the data coming from Web UI and picked up appropriate models for making prediction. Then visualized predicted result on Web UI.

5.3 Data Collection and Preprocessing

We have use Html Agility Pack for data collection process through web data sources. Html agility pack is the library that helps to scrap the matches' data from www.cricinfo.com. It is speedup data collection process because we don't have freely available APIs for collection ODI cricket matches data. Initially we stored scraped data into separate database to minimize the weight of primary database. Then cleaned data send to primary database for future query manipulation.

5.4 User Interface Module

This module enables user interaction with the system in terms of input and output requirements. More importantly, this module has direct access to the main back-end database and Analytical Engine. This module is not limited to handle input/output, but execute certain pre-processing on input data and also processing of results to appear in different forms.

5.5 CRIC-Win Analytic Engine

This engine primarily works as data mining engine, and performs the roles of data preprocessing, attribute selection, classification and finally analyzed results push to the main database. Figure 5.2 shows phases of data mining that are covered by CRIC-Win Analytic Engine. Also this engine directly connected with WEKA tool. The output of the WEKA tool directly captured by CRIC-ESS Analytic Engine and then parallel sends to both back end and the user interface module.

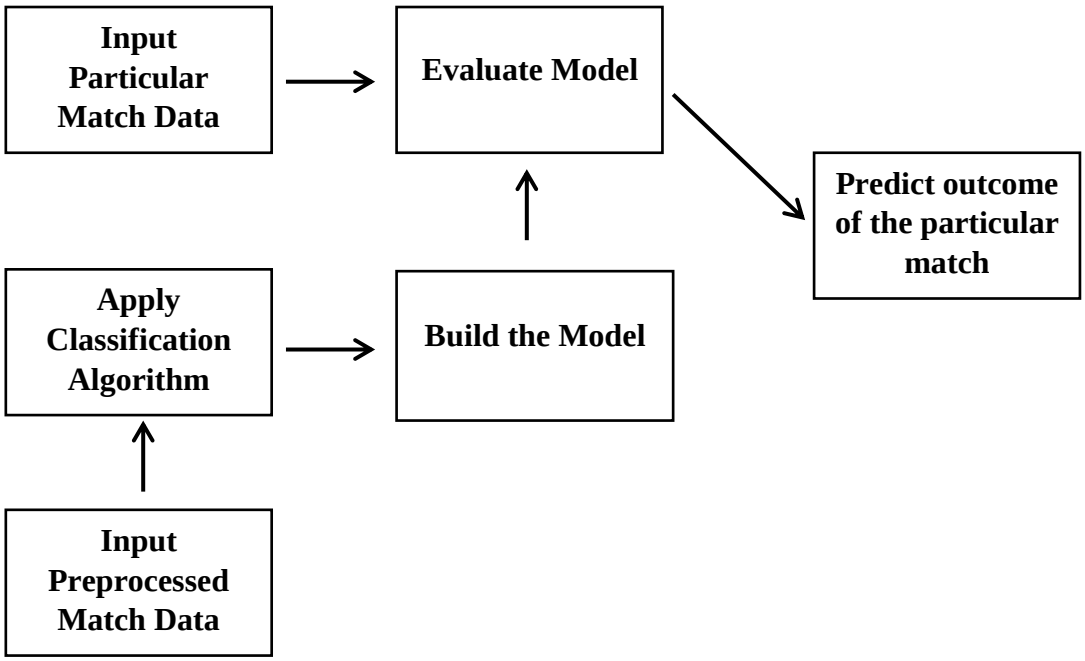


Figure 5.2 Design of CRIC-Win Analytical Engine

Based on the request coming from the user interface this module can generate output through database. This module cannot be accessed directly by the users without going through Web UI. Because of user have to choose two teams, players details, ground details and other necessary stuffs from Web UI. When user request coming from Web UI, will send to the backend data module then backend data module send back json formatted data to the Web UI. Then only user can connect with CRIC-Win Analytic Engine and it responsible to evaluate the model based on given match data and predicted result send back to the Web UI. However, limited information related to the current session can be returned to user interface module from the CRIC-Win Analytic Engine. In general, this module can be seen as an extension to and ordinary data mining tool except for input/output mechanism.

5.6 Back-End Database

The back-end database is the key to handle ongoing data mining with CRIC-Win Analytical Engine. This is also an essential component to work with the User interface. The design of this database is rather crucial activity in the solution. This affects the analytical engine as well as user interface.

The database consist all analyzed summary data and also players' details, Match details, Ground Details and other supported tables to minimize data redundancy. Details of the design of back-end database is shown in Appendix B.

According to this design without going through CRIC Win Analytical Engine the system can facilitate sessions to query on the database, related to the past data. More importantly, interaction between user interface and back-end database create opportunity for the CRIC-Win Analytical Engine work efficiently by knowing what exactly should be done for the specific user request. Thus work load on the analytic Engine will be reduced.

5.7 Summary

This chapter discussed the design of CRIC-ESS solution. We described three modules, namely, User Interface, CRIC-ESS Analytical Engine and Back-End Database together with their roles. Next chapter describes the implementation of these modules to exhibits the intended roles.

Chapter 6

Overall Implementation of CRIC-WIN Predictor Tool

6.1 Introduction

In chapter 5, we described overall design of the proposed solution. This chapter provides implementation details of each of the modules mentioned in the previous chapter. Nevertheless, this presents software and algorithms used in each module with sample outputs.

6.2 Overall Implementation

Overall solution is a web based one and has been implemented as .Net based application running on any web browser. The implementation is concerned about ASP.net MVC architecture and java based rest services for connecting WEKA tool with Web UI. Main module of our product CRIC-Win Analytical Engine implement as java based soap service. Because data mining tool called WEKA can be easily integrated with java platform. Web UI module develops based on .Net technology and Backend data module is develop by using Entity Framework 6.0 with MS SQL RDBMS technology for creating databases. Figure 6.1 shows snapshot of CRIC-Win Predictor tool. Next we discussed about module wise implementation with reference to the top level architecture.

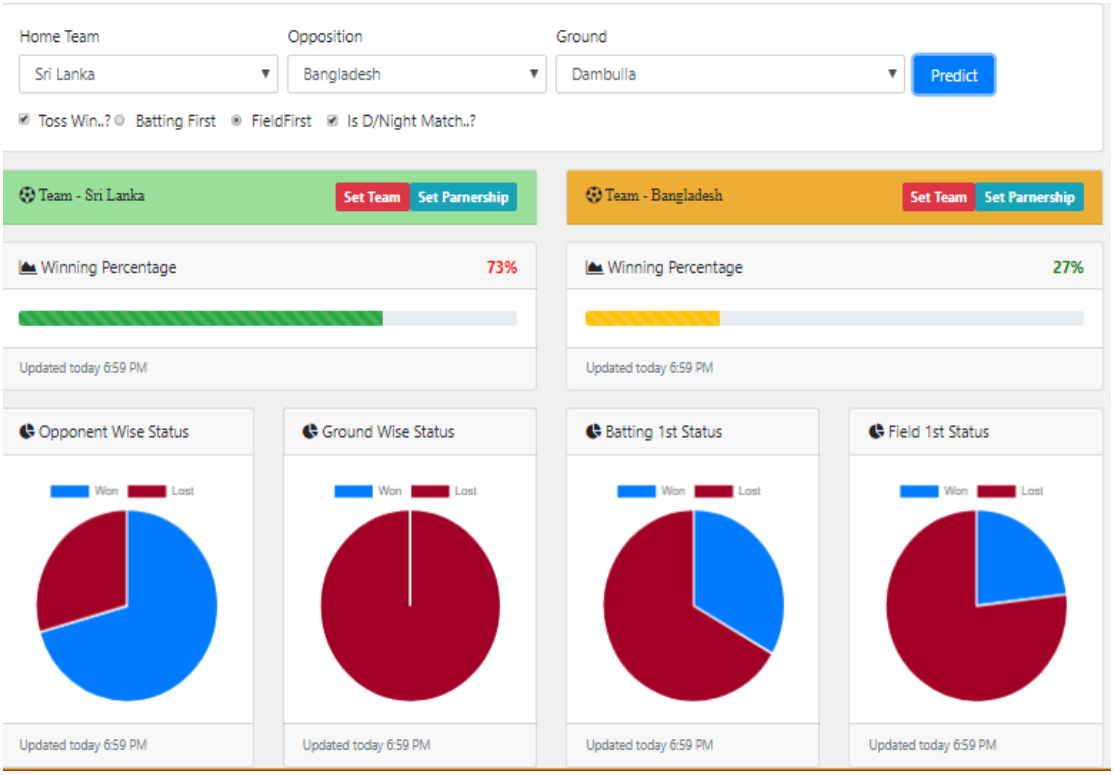


Figure 6.1 CRIC-Win Predictor Tool

6.3. Data Collection and Preprocessing

Data for the analysis is collected from www.cricinfo.com. However, we have to use special .Net library called Html Agility Pack. We have created web data scrapping tool to gather cricket match related data. Figure [] in Appendix A, shows data scrapping tool with its interfaces. Then scrapped data push into separated database because scrapped data had lot of missing values and noisy values in addition to that redundancy data is the major problem when scrapping data from web. After eliminating redundant data and cleaning noisy data we stored them in separated database which is used to create main backend module. We do this purposely to reduce the weight of main database module. Next we use MS SQL Server Management studio for query manipulation and fetching up necessary information to create the main data set. Figure 6.1 shows sample data set and MS SQL queries use to create main data set in .ARFF format which required for WEKA. Finally we use WEKA tool to apply Attribute Selection filter to select only necessary attribute with the evaluator InfoGainAttributeEval which evaluated the worth of an attribute by measuring the information gain with respect to the class which shows in Appendix A.

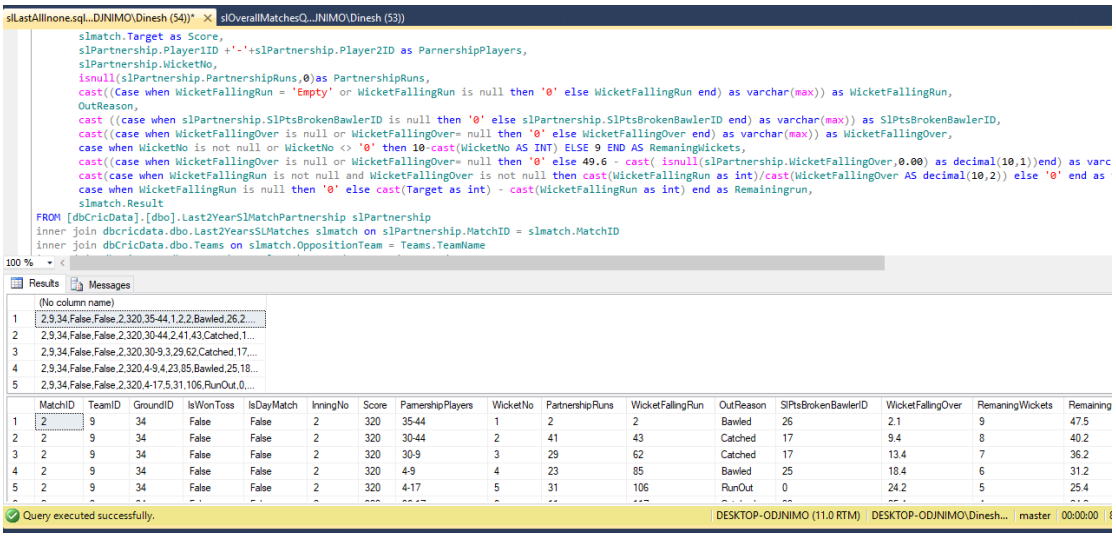


Figure 6.2 Data Preprocessing

6.3 Implementation of User Interface module

Mainly user interface module is developed by using ASP.Net MVC technology which is latest web technology introduced by Microsoft. All frontend developments are done by using latest technology and used JQuery Ajax for all client side scripting, in addition to that all data is formatted to .json for improving the user experience with web UI, because all client side requests are running asynchronously and user does not feel it. In application development coupling and cohesion is the two major concern. So our development is ensure that keeping independence between two other modules namely CRIC-Win Analytic Engine and Backend Data Module with Web UI module. This feature will help for future expansion of CRIC-Win Predictor tool.

6.4 Implementation of CRIC-Win Analytical Engine

CRIC-Win Analytical Engine is the major module of our solution. This module is developed by using java technology with WEKA API. WEKA tool can be easily integrated with java but our UI module was developed by using .Net technology so sort out this language gap we introduced separate java based SOAP service module. After creating complete data set we have to construct data models to predict outcome of the selected match in given situation. In model creation we have try different machine learning techniques namely clustering, classification and association rule mining. Selected algorithms from each technique were trained using the WEKA tool. Clustering and association rule mining did not make any contribution to predict outcome of the match, because we have used multiple independent attributes therefore placing them in groups based on their similarity did not seem feasible.

However, classification techniques produced significant results. In machine learning, Naïve Bayes and Decision Trees are two popular classification approaches. We also looked into two other algorithms, AdaBoost and Bagging. The results obtained were interesting in terms of improving the performance. Likewise we create separate data models and stored them in backend data module in binary format. Then our CRIC-Win Analytical Engine is responsible to fetch appropriate model from the Backend data module and evaluate the model based on user requests. Finally predicted outcome send back to Web UI. This CRIC-Win analytical engine can be easily plugged any web application without any modification as shown in figure 6.2.

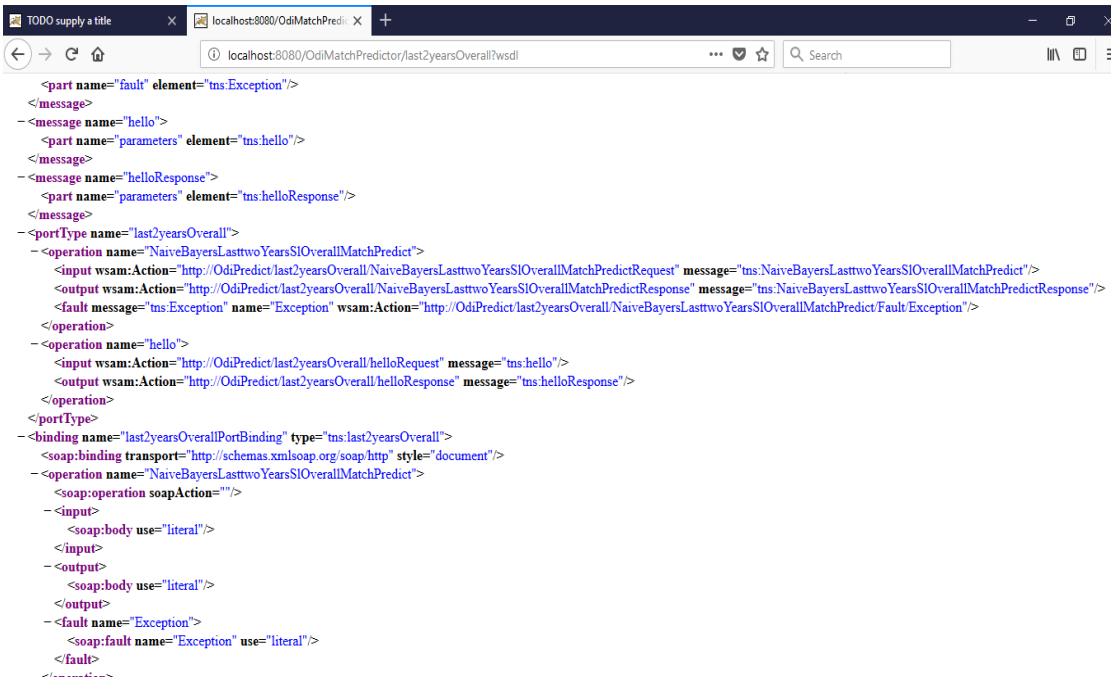


Figure 6.3 CRIC-Win Analytical Engine WSDL

6.5 Implementation of Back-End Data Module

Back end data module is the separate module that handle all data request coming from the Web UI. The reason for using separate module for data requesting is that reduce the dependency of main database with Web UI and CRIC-Win Analytical Engine. This module was implemented by using Entity Framework 6.0 which is very easy to use with MS SQL database and easy to maintain. A snapshot of Entity framework model diagram and main database diagram appears in Appendix A.

6.6 Summary

This chapter provided overall implementation details of each module of the proposed solution. Moreover, it mentioned software and data mining techniques for models development with align to design. Next chapter evaluates all the modules implemented in the solution.

Chapter 7

Evaluation

7.1 Introduction

The chapter 6 discussed the details on implementation of all the modules mentioned in the proposed solution. This chapter justifies and evaluates the overall solution, data mining techniques and data models used in CRIC-Win Analytic Engine.

7.2 Evaluation of Classification Techniques

We have trained collected data set using different classification techniques namely Naïve Bayes, J48, Bagging and RandomForest by the help of WEKA tool. ODI Cricket match outcome prediction model is the main input for the CRIC-Win Analytical Engine. For evaluating a classifier quality we can use confusion matrix which can be evaluated various measurements such as accuracy, recall and precision. These measurements and their definition are given below Table

Measurement	Formula	Description
Precision	TP/(TP + FP)	The percentage of positive predictions those are correct.
Recall / Sensitivity	TP / (TP + FN)	The percentage of positive labeled instances that were predicted as positive.
Specificity	TN / (TN + FP)	The percentage of negative labeled instances that were predicted as negative.
Accuracy	(TP + TN) / (TP + TN + FP + FN)	The percentage of predictions those are correct.

Table 7.1 Classification Evaluation Measurements

Using previously mentioned measurements we can derived TP rate, FP rate, F-measure and ROC area. TP rate is equal to sensitivity, while FP rate is equal to 1-specificity F-measure calculated by precision and recall. The area under a ROC curve quantities the overall ability of the test to discriminate between usefulness and uselessness a truly useless test as an area of 0.5. A perfect test has an area of 1.00. Usually best models have higher TP rate, lower FP rate and ROC space close to 1.00. The classification results of each techniques are given Table 7.2.

Technique	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC
Naïve Bayes	0.982	0.082	0.982	0.982	0.981	0.978
Decision Tree	0.975	0.092	0.975	0.975	0.974	0.998
AdaBoost	0.772	0.746	0.691	0.772	0.704	0.867

Bagging	0.696	0.809	0.6	0.696	0.644	0.402
RandomForest	0.722	0.759	0.641	0.722	0.674	0.829

Table 7.2 Comparison of Classification Techniques to Determine Overall Match Outcome

Table 7.2 shows that Naïve Bayes and Decision Tree represent almost equal accuracy, but Naïve Bayes having higher accuracy value than Decision Tree. It was found that Naïve Bayes produced the best results in ODI Cricket match winning prediction. Other classification techniques as shown in above table did not perform significantly well in our research. WEKA model creation summary attached into Apendix C.

7.3 Evaluation of CRIC-Win Predictor Tool

We tried some match wining factors including toss, day/night effect, and ground advantages for predicting match outcome before starting match and next we consider batsman combination with their partnership runs to predict match outcome.

Teams	Win Toss	Batting 1st	Day Match?	Ground	Win Probability
Sri Lanka	Yes	Yes	No	Dambulla	52%
Bangladesh	No	No	No		48%
Sri Lanka	Yes	No	No	Dambulla	73%
Bangladesh	No	No	No		27%
Sri Lanka	No	Yes	Yes	Mohali	25%
India	Yes	No	Yes		75%
Sri Lanka	Yes	No	Yes	Melbourne	60%
India	No	Yes	Yes		40%
Sri Lanka	Yes	No	No	Hambanthota	79%
Zimbabwe	No	Yes	No		21%

Table 7.3 Analysis Wining Factors Before Start the Match

Table 7.4 shows CRIC-Win Predictor tool generated result of Sri Lanka vs. India match held in Dharmasala on 10th Dec 2017. Snapshot of UI outcome shows in Appendix D.

Teams	Win Toss	Batting 1st	Day Match?	Ground	SL Win Probability
Sri Lanka	Yes	No	No	Dharmasala	51%
India	No	Yes	No		49%

Table 7.4 Sri Lanka vs. India Match Pre-Outcome

Wicket No	Partnership combination	Runs	Total Run	Overs	SL Win Probability
04	MK Pandey SS Lyer	8	16	12.5	75%
06	MS Dhoni HH Pandya	12	28	15.2	73%
10	MS Dhoni YS Chahal	35	112	38.2	85%

Table 7.5 Analysis of 1st Inning Partnerships

Wicket No	Partnership combination	Runs	Total Run	Overs	Target	Win Probability
03	AD Mathews WU Tharanga	53	65	12.3	112	87%

Table 7.6 Analysis of 2nd Inning Partnerships

As shown in Tables 7.3-7.6 our CRIC-Win Predictor tool produced some interesting results, so winning predictor tool can be used to predict outcome of future matches using the knowledge discovered from our research. Also cricketers, coaches, team selectors and other interesting parties who interested to gain competitive advantages to win the game, can be used our predictor tool for decision making.

7.4 Summary of Evaluation

This chapter evaluated the methodologies and the results discussed in the implementation chapter. Next chapter discuss limitations and future improvements of CRIC-Win Predictor tool.

Chapter 8

Conclusion and Further Work

8.1 Introduction

In this research, we have addressed the problem of predicting the chances of victory in a One Day International cricket match. By analyzing different attributes related to the ODI game, we have been able to predict the winning criteria formulated using attributes from the dataset. We have developed a web based software tool called CRIC-Win Predictor based on our study.

The main goal of this research is to learn a model for predicting game progression and outcome in ODI cricket match. We separate our prediction into two segments. First one is for predicting wining possibility by considering toss effect, ground condition, day night effect and opponent. Next we consider batting performances by taking into players combination, wicket no, partnership runs, total runs or target and overs to predict the progression of the game. This way help us to simulate the game properly. Naïve Bayes algorithm is used to predict match outcome any given situation. We demonstrated the quality and accuracy of our predictions with an extensive set of experiments on real ODI cricket data. In addition to predicting wining chances for future segments, our winner prediction accuracy is indicate higher value because of calculate wining possibility before start the match and while playing the match.

Currently, team strategists rely on a combination of personal experience and team constitution. Inherently, the methodology employed by human experts is to extract and leverage important information from both past and current game statistics. So, match outcome prediction can also be helped to make strategic decisions to increase wining chances. The main contributions of our work can be listed as follow.

- Comparison of machine learning techniques which revealed that classification is the best approach to solve the problem.
- Evaluation of various classifiers over real data which proved that Naïve Bayes works best over the concerned datasets.
- Partnership combinations, partnership runs and wicket falling pattern taken into predict progression of the game.
- Development of the web based CRIC-Win Predictor tool that can be used in real-world scenarios to predict the chances of victory in a given match.

8.2 Limitations

By default cricket is unpredictable game. Some tidy situation cannot imagine winner until last moment, so making prediction about unpredictable game is real challenge. Sometime historical match data mapping into current match situation is not an easy task. In addition to that new player may be turning point of the match but in prediction generated using historical match data simply discarded it.

8.3 Future Developments

As future work, we are planning to expand our analysis using more attributes such as the previous match result of the selected team and the opponent team, the number of known batsmen in the selected team and the opponent team and more. It is also possible to apply association techniques to predict partnership broken bawler.

We are currently working to further reduce the prediction error. Furthermore, to make the prediction engine functionally complete, we intend to predict fall of wickets, overcoming the challenges presented in current data set and expanding match prediction other countries as well. Finally, we aim to leverage bowler's features (in addition to the batsmen's) to improve the prediction accuracy even further.

8.4 Summary

This chapter concludes the thesis by describing the solution given with data mining to analyze the ODI Cricket match wining prediction and how match outcome prediction and simulation of match progression help to make decisions to gain competitive advantages for improving the wining chances.

References

[1] A. Kaluarachchi and S. V. Aparna, “CricAI: A classification based tool to predict the outcome in ODI cricket,” in Information and Automation for Sustainability (ICIAFs), 2010 5th International Conference on, 2010, pp. 250–255.

[2] T. Singh, V. Singla, and P. Bhatia, “Score and winning prediction in cricket through data mining,” in Soft Computing Techniques and Implementations (ICSCTI), 2015 International Conference on, 2015, pp. 60–66.

[3] Y. Saito, M. Kimura, and S. Ishizaki, “Real-time prediction to support decision-making in soccer,” in Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K), 2015 7th International Joint Conference on, 2015, vol. 1, pp. 218–225.

[4] V. V. Sankaranarayanan, J. Sattar, and L. V. Lakshmanan, “Auto-play: A data mining approach to ODI cricket simulation and prediction,” in Proceedings of the 2014 SIAM International Conference on Data Mining, 2014, pp. 1064–1072.

[5] A. Shah, D. Jha, and J. Vyas, “WINNING AND SCORE PREDICTOR (WASP) TOOL.”

[6] R. K. Khan, I. Manarvi, and others, “Evaluating performance of Blackcaps of New Zealand vs. global cricket teams,” in Computers & Industrial Engineering, 2009. CIE 2009. International Conference on, 2009, pp. 1500–1504.

[7] I. Bhandari, E. Colet, and J. Parker. Advanced Scout:Data mining and knowledge discovery in NBA data.Data Mining and Knowledge Discovery, 1(1):121{125,1997.

[8] D. Lutz. A cluster analysis of NBA players. In MITSloan Sports Analytics Conference, 2012.

[9] G. Gartheeban and J. Guttag. A data-driven method for in-game decision making in mlb: when to pull a starting pitcher. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '13, pages 973{979, New York, NY, USA, 2013. ACM.

[10] S. Luckner, J. Schroder, and C. Slamka. On the forecast accuracy of sports prediction markets. In Negotiation, Auctions, and Market Engineering, International Seminar, Dagstuhl Castle, volume 2, pages 227{234,2008.

[11] Khabir Uddin Mughal. Top 10 Most Popular Sports In The World. http://sporteology.com/top-10-popular-sports-world/Accessed 2 February 2015.

[12] Allsopp, P., & Clarke, S. R., Rating teams and analyzing outcomes in one-day and test cricket. Journal ofthe Royal Statistical Society A, (2004).

[13] S.R. Clarke, “Dynamic programming in one-day cricket— optimal scoring rates”, Journal of the Operational Research Society, 1988, Vol. 39, No. pp. 331–337.

[14] B.M De Silva, and T.B. Swartz, Estimation of the magnitude of the victory in one-day cricket. Australia and New Zealand Journal of Statistics, 2001, Vol. 43, pp. 1369-1373.

[15] V. Veppur Sankaranarayanan, "Towards a time-lapse prediction system for cricket matches," PhD Thesis, University of British Columbia, 2014.

[16] D. Prasetio and others, "Predicting football match results with logistic regression," in *Advanced Informatics: Concepts, Theory And Application (ICAICTA), 2016 International Conference On*, 2016, pp. 1–5.

[17] "Weka 3 - Data Mining with Open Source Machine Learning Software in Java." [Online]. Available: https://www.cs.waikato.ac.nz/ml/weka/. [Accessed: 07-May-2018].

[18] rpetrusha, "Language Independence and Language-Independent Components." [Online]. Available: https://docs.microsoft.com/en-us/dotnet/standard/language-independence-and-language-independent-components. [Accessed: 07-May-2018].

[19] "Microsoft Launches Its .NET Distribution For Linux And Mac," *TechCrunch*, 29-Apr-2015. .

[20] R. P. Schumaker, O. K. Solieman, and H. Chen, "Open Source Data Mining Tools for Sports," in *Sports Data Mining*, vol. 26, Boston, MA: Springer US, 2010, pp. 89–92.

[21] "Digital Scout." [Online]. Available: https://www.digitalscout.com/. [Accessed: 18-May-2018].

[22] F. Schoonjans, "Logistic regression," *MedCalc*. [Online]. Available: https://www.medcalc.org/manual/logistic_regression.php. [Accessed: 18-May-2018].

[23] "Lightning Fast Data Science Platform | RapidMiner." [Online]. Available: https://rapidminer.com/. [Accessed: 18-May-2018].

[24] heatherbshapiro, "What is Azure Machine Learning Studio?" [Online]. Available: https://docs.microsoft.com/en-us/azure/machine-learning/studio/what-is-ml-studio. [Accessed: 18-May-2018].

Appendix – A

Attribute Selection

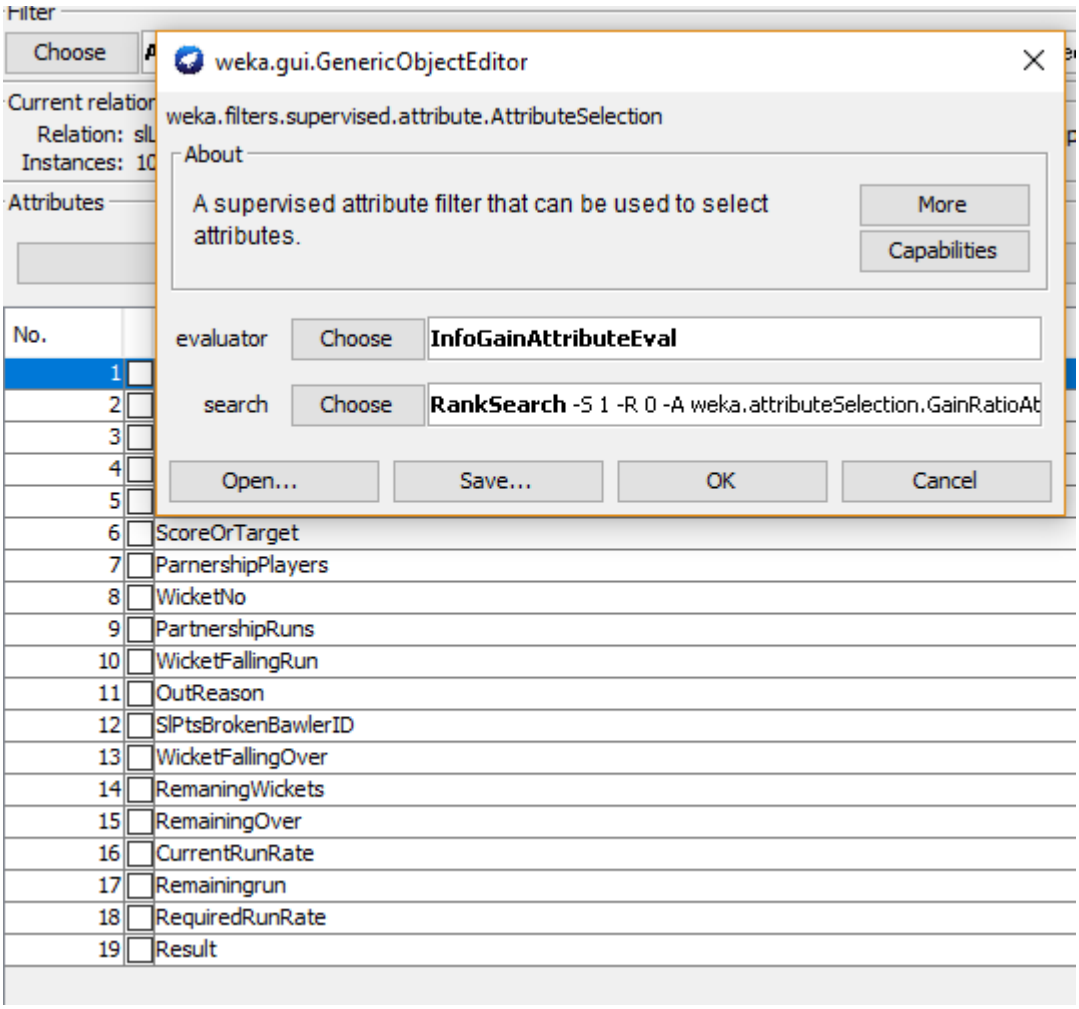


Figure A: 1 Information Gain Attribute Evaluation

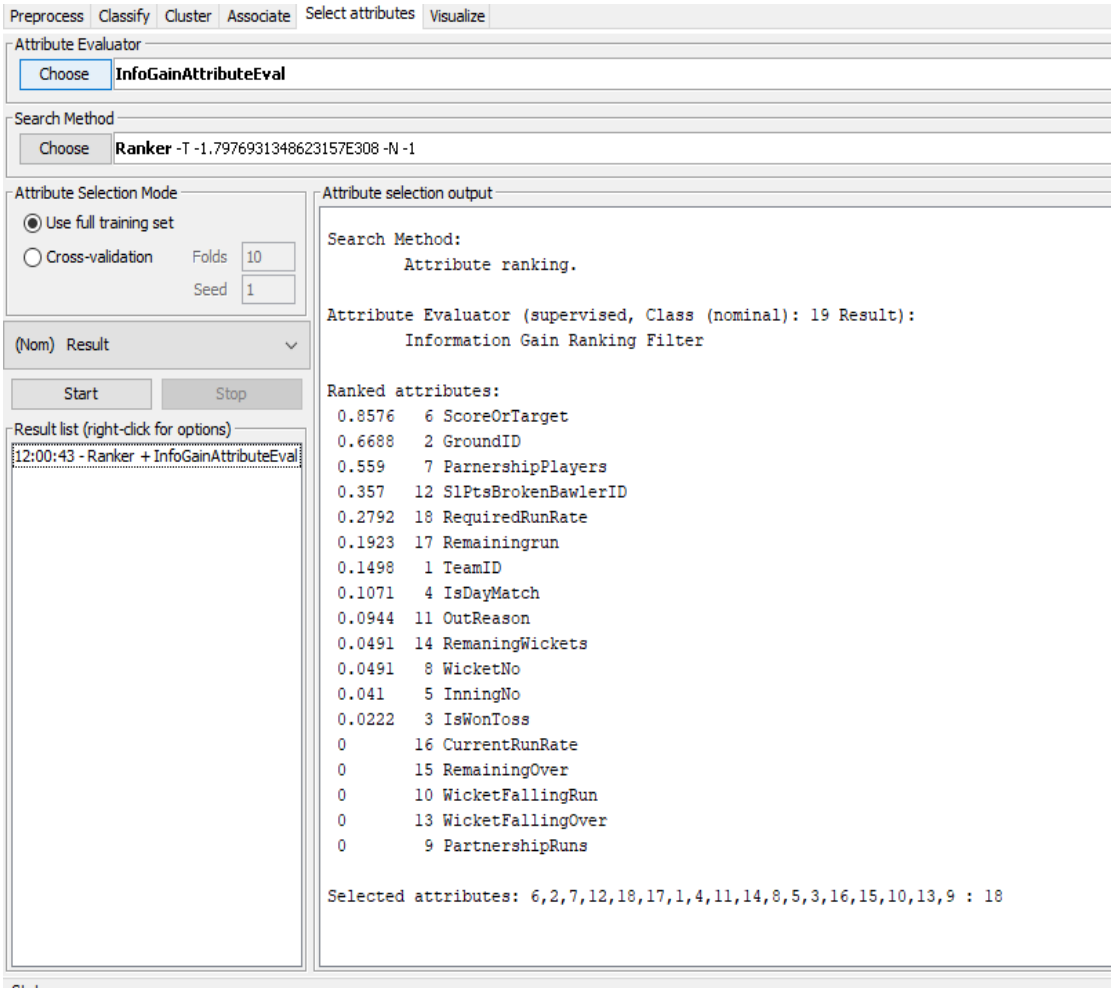
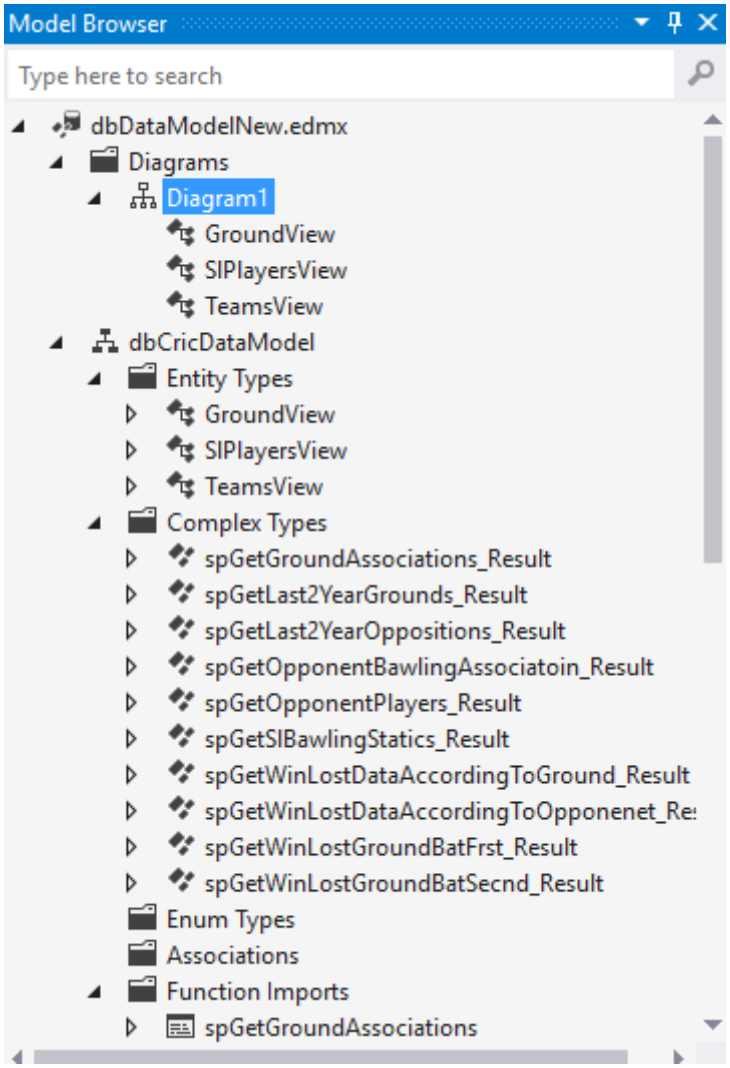


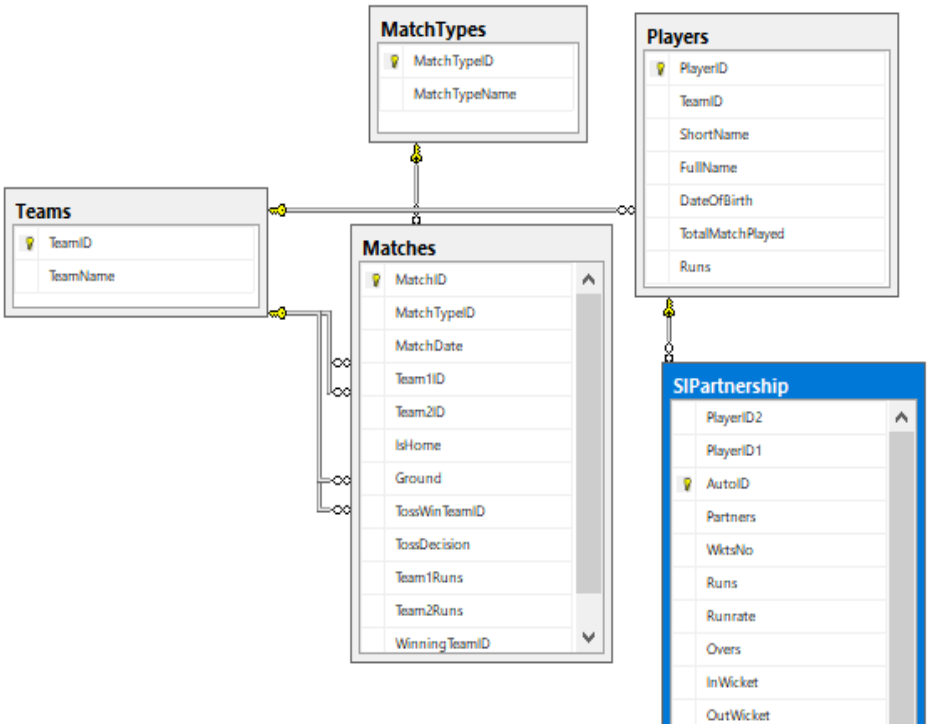
Figure A: 2 Attribute Selection

Appendix - B

Backend Data Module Implementation



Appendix B: 1 Entity Framework Model Browser



Appendix B: 2 Back-End Database Diagram

Appendix - C

Model Evaluation Summary

```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          54               98.1818 %
Incorrectly Classified Instances         1                1.8182 %
Kappa statistic                          0.9364
Mean absolute error                      0.0216
Root mean squared error                  0.1371
Relative absolute error                  7.0812 %
Root relative squared error             35.5068 %
Total Number of Instances               55

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.9       0          1         0.9       0.947      0.978    Won
                 1         0.1        0.978     1         0.989      0.978    Lost
Weighted Avg.    0.982     0.082      0.982     0.982     0.981      0.978
```

Appendix C: 1 Naive Bayes Model Creation Summary

Classifier output

```
Size of the tree :         83


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          77               97.4684 %
Incorrectly Classified Instances         2                2.5316 %
Kappa statistic                          0.9217
Mean absolute error                      0.0253
Root mean squared error                  0.143
Relative absolute error                  7.3896 %
Root relative squared error             34.7324 %
Total Number of Instances               79

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.882     0          1         0.882     0.938      0.998    Won
                 1         0.118      0.969     1         0.984      0.998    Lost
Weighted Avg.    0.975     0.092      0.975     0.975     0.974      0.998
```

Appendix C: 2 Decision Tree Model Creation Summary

```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          61               77.2152 %
Incorrectly Classified Instances        18               22.7848 %
Kappa statistic                          0.0379
Mean absolute error                      0.225
Root mean squared error                  0.3489
Relative absolute error                 65.666  %
Root relative squared error             84.7424 %
Total Number of Instances               79

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.059     0.032      0.333     0.059      0.1        0.867    Won
                 0.968     0.941      0.789     0.968      0.87       0.867    Lost
Weighted Avg.    0.772     0.746      0.691     0.772      0.704      0.867
```

Appendix C: 3 AdaBoost Model Creation Summary

```
Time taken to build model: 0.02 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          57               72.1519 %
Incorrectly Classified Instances        22               27.8481 %
Kappa statistic                         -0.0483
Mean absolute error                      0.2969
Root mean squared error                  0.3797
Relative absolute error                 86.6652 %
Root relative squared error             92.221  %
Total Number of Instances               79

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.059     0.097      0.143     0.059      0.083      0.829    Won
                 0.903     0.941      0.778     0.903      0.836      0.829    Lost
Weighted Avg.    0.722     0.759      0.641     0.722      0.674      0.829
```

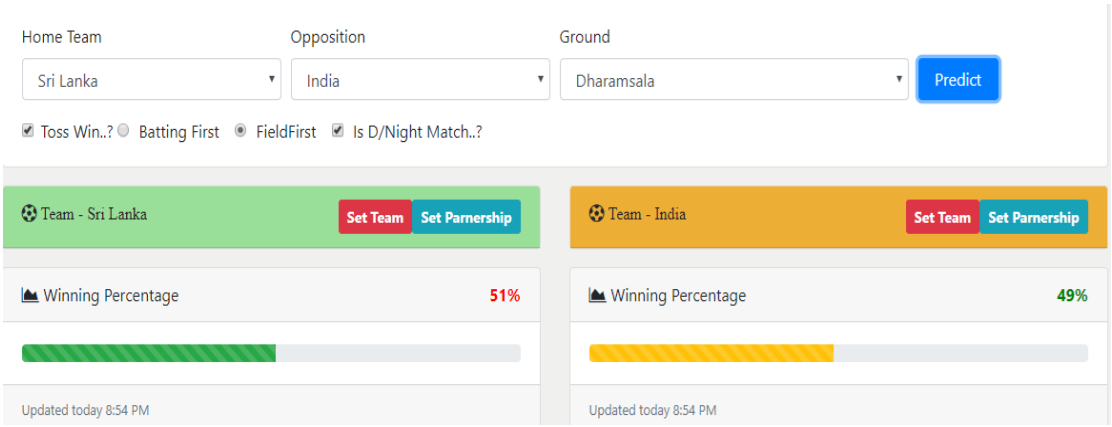
Appendix C: 4 RandomForest Model Creation Summary

```
Time taken to build model: 0.02 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          55               69.6203 %
Incorrectly Classified Instances        24               30.3797 %
Kappa statistic                         -0.1435
Mean absolute error                      0.3692
Root mean squared error                  0.4611
Relative absolute error                107.7608 %
Root relative squared error            111.9925 %
Total Number of Instances               79

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0         0.113      0          0         0          0.4      Won
                 0.887     1          0.764      0.887     0.821      0.402    Lost
Weighted Avg.    0.696     0.809      0.6        0.696     0.644      0.402
```

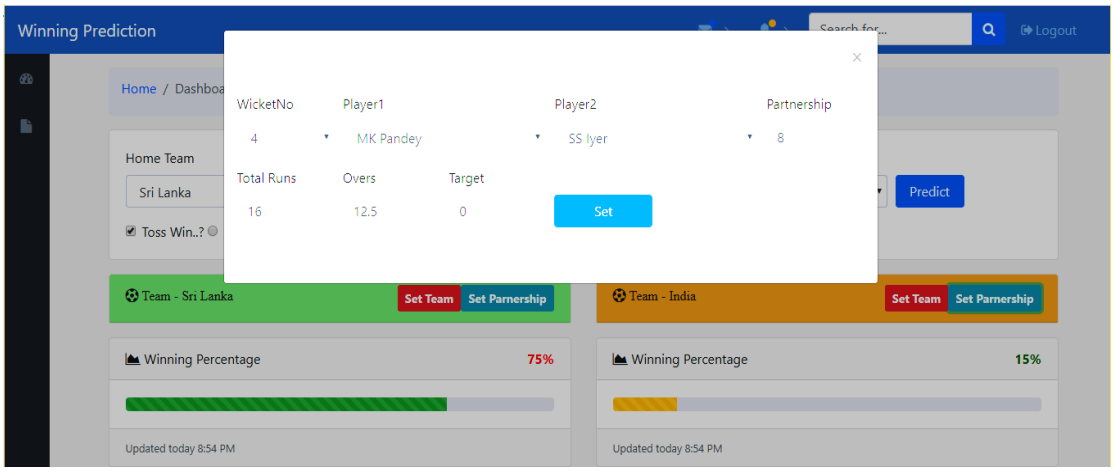
Appendix C: 5 Bagging Model Creation Summary

Appendix - D

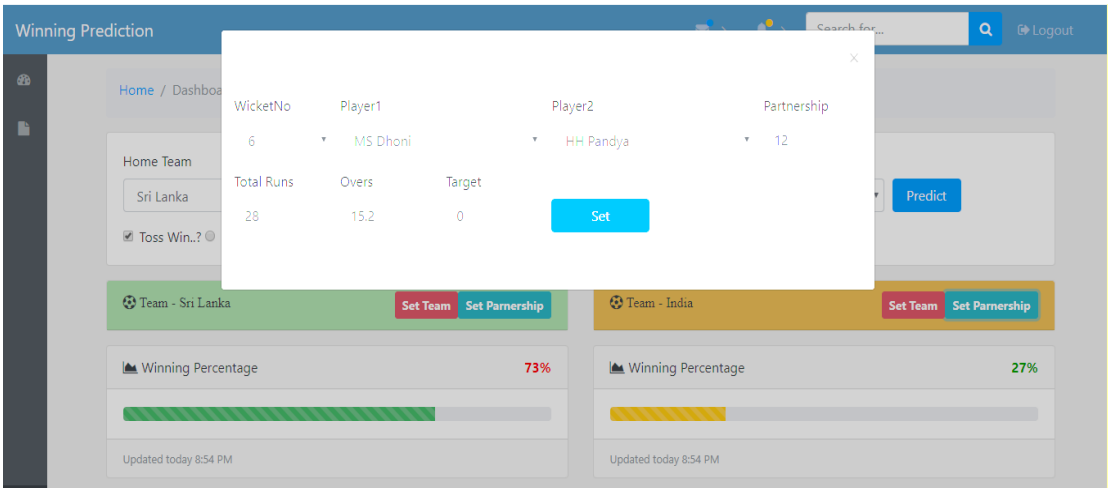
CRIC-Win Predictor Tool Test Result



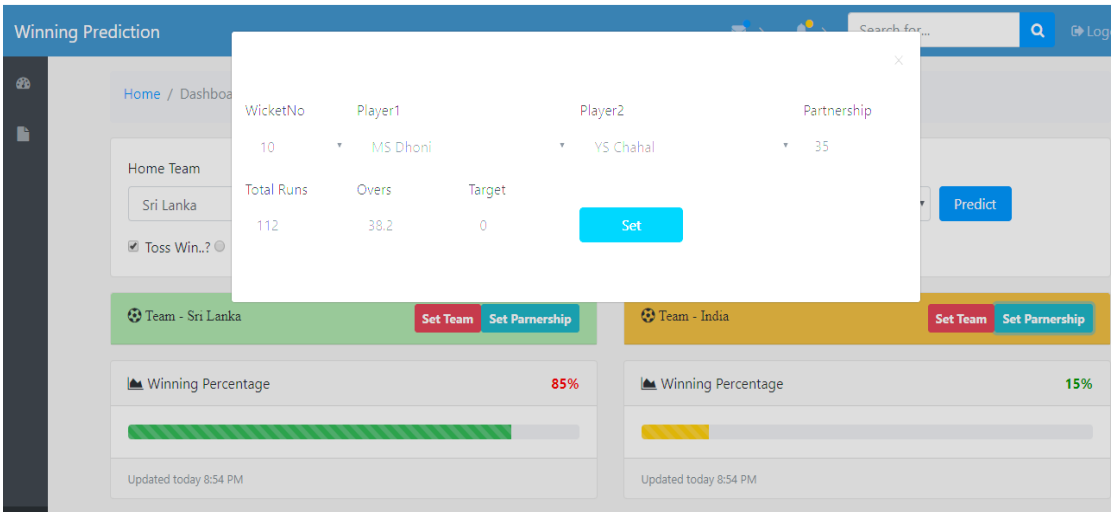
Appendix D: 1 SL vs India Match Outcome



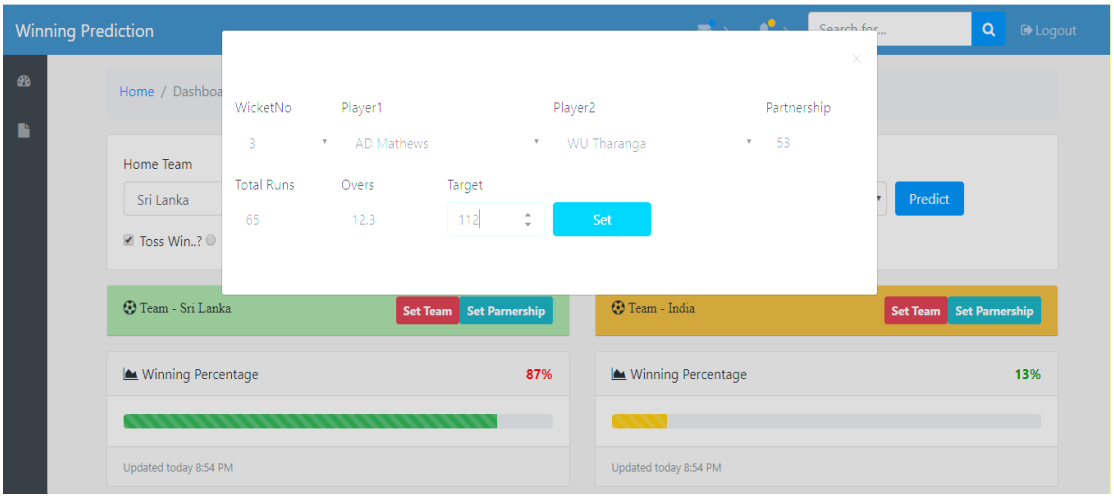
Appendix D: 2 1st Inning 4th Wicket Partnership



Appendix D: 3 1st Inning 6th Wicket Partnership



Appendix D: 4 1st Inning Last Wicket Partnership



Appendix D: 5 2nd Inning 3rd Wicket Partnership

Appendix - E

Code Snippet

```
static void Main(string[] args)
      {
         Entities db = new Entities();
         List<SlPartnership> lstSlMatch2017 = new List<SlPartnership>();
         HtmlWeb hw = new HtmlWeb();
         HtmlDocument hDoc =
hw.Load(@"http://stats.espncricinfo.com/ci/engine/stats/index.html?class=2;home_or_away=
1;host=8;result=1;result=2;spanmin1=14+Jan+2008;spanval1=span;team=8;template=results;
type=allround;view=results");
         HtmlNodeCollection rows =
hDoc.DocumentNode.SelectNodes("//*[@id=\"ciHomeContentlhs\"]/div[3]/table[3]/tbody/tr"
);
         for (int i = 1; i <= rows.Count; i++)
         {
            HtmlNodeCollection columns =
hDoc.DocumentNode.SelectNodes(string.Format("//*[@id=\"ciHomeContentlhs\"]/div[3]/tab
le[3]/tbody/tr" + "[" + "{0}" + "]" + " /td", i));

            SlPartnership obj = new SlPartnership
            {
               Partners = columns[0].InnerText,
               WktsNo = columns[1].InnerText,
               Runs = columns[2].InnerText,
               Overs = columns[3].InnerText,
               Runrate = columns[4].InnerText,
               InWicket = columns[3].InnerText,
               OutWicket = columns[4].InnerText,
               FirstOrSecndInning = columns[5].InnerText,
               Opposition = columns[7].InnerText,
               Ground = columns[8].InnerText,
               MatchStart = columns[9].InnerText,
               IsDayMatch = false,
               IsWinMatch = false,
               IsWonToss = false

            };
            //lstSlMatch2017.Add(obj);


         }

        db.SlPartnership.AddRange(lstSlMatch2017);
        db.SaveChanges();

      }
```

Appendix E: 1 Web Cricket Data Scraping

```
HomeServices objHomeServices;
        public ActionResult Home()
        {
            ViewBag.slPlayerGrid = BGrid.CreateBGrid<SlPlayersView>("slPlayerGrid",
GetFirstApprovalPendingGridModel(), new List<SlPlayersView>(), new GridOptions() {
EditOption = false, EditFunctionName = "edit" });
            ViewBag.OppPlayerGrid = BGrid.CreateBGrid<SlPlayersView>("OppPlayerGrid",
GetFirstApprovalPendingGridModel(), new List<SlPlayersView>(), new GridOptions() {
EditOption = false, EditFunctionName = "edit" });
            return View();
        }

        public JsonResult GetTeams(bool isOverall)
        {
            objHomeServices = new HomeServices();
            var reslt = objHomeServices.GetTeamData(isOverall);
            return Json(reslt, JsonRequestBehavior.AllowGet);
        }

        public JsonResult getOpponentPlayers(bool isOverall,int TeamID)
        {
            objHomeServices = new HomeServices();
            var reslt = objHomeServices.GetOpponentAllPlayers(isOverall,TeamID);
            return Json(reslt, JsonRequestBehavior.AllowGet);
        }


        public JsonResult GetGronds(bool isOverall)
        {
            objHomeServices = new HomeServices();
            var reslt = objHomeServices.GetGrounds(isOverall);
            return Json(reslt, JsonRequestBehavior.AllowGet);
        }

        public JsonResult GetGroundAssociations(string GroundID)
        {
            objHomeServices = new HomeServices();
            var reslt = objHomeServices.GetGroundAssocations(GroundID);
            return Json(reslt, JsonRequestBehavior.AllowGet);
        }

        public JsonResult GetOppositionBawlingAssociations(string TeamID)
        {
            objHomeServices = new HomeServices();
            var reslt = objHomeServices.GetOppositionBawlingAssociations(TeamID);
            return Json(reslt, JsonRequestBehavior.AllowGet);
        }
```

Appendix E: 2 Fetch Data from Backend Module

```
public WinLostCountView GetGroundWiseBattingSecns(int groundID)
      {
         try
         {
            db = new dbCricDataEntities();
            List<spGetWinLostDataAccordingToGround_Result> lst =
db.spGetWinLostDataAccordingToGround(groundID).ToList();
            WinLostCountView objWinlossView = new WinLostCountView();
            foreach (var item in lst)
            {
               if (item.Result == "Won")
               {
                  if (item.Count == 0)
                  {
                     objWinlossView.WinCount = 0;
                  }
                  else
                  {
                     objWinlossView.WinCount = Convert.ToInt16(item.Count);
                  }

               }
               else if (item.Result == "Lost")
               {
                  if (item.Count == 0)
                  {
                     objWinlossView.LossCount = 0;
                  }
                  else
                  {
                     objWinlossView.LossCount = Convert.ToInt16(item.Count);
                  }

               }

            }
            if (lst.Count == 0)
            {
               objWinlossView.LossCount = 0;
               objWinlossView.LossCount = 0;
            }
            return objWinlossView;
         }
         catch (Exception)
         {

            throw new Exception("Run Time Error Occured");
         }
      }
```

Appendix E: 3 Backend Service Module code Snippet

```
package OdiPredict;

import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;
import weka.classifiers.Classifier;
import weka.core.Instances;
import weka.core.converters.ConverterUtils;

/**
 *
 * @author Dinesh
 */
@WebService(serviceName = "last2yearsOverall")
public class last2yearsOverall {

    @WebMethod(operationName = "NaiveBayersLasttwoYearsSlOverallMatchPredict")
    public String NaiveBayersLasttwoYearsSlOverallMatchPredict(String modelFileSerialized,
String testFileARFF)
    throws Exception
    {
    // Deserialize the classifier.
    Classifier classifier =
        (Classifier) weka.core.SerializationHelper.read(
            modelFileSerialized);

    // Load the test instances.
    Instances testInstances = ConverterUtils.DataSource.read(testFileARFF);

    // Mark the last attribute in each instance as the true class.
    testInstances.setClassIndex(testInstances.numAttributes()-1);

    int numTestInstances = testInstances.numInstances();
//    System.out.printf("There are %d test instances\n", numTestInstances);

    //Loop over each test instance.
    String reslt = "";
    for (int i = 0; i < numTestInstances; i++)
    {
        // Get the true class label from the instance's own classIndex.
        String trueClassLabel =
            testInstances.instance(i).toString(testInstances.classIndex());

        // Make the prediction here.
        double predictionIndex =
            classifier.classifyInstance(testInstances.instance(i));

        // Get the predicted class label from the predictionIndex.
        String predictedClassLabel =
            testInstances.classAttribute().value((int) predictionIndex);

        // Get the prediction probability distribution.
        double[] predictionDistribution =
            classifier.distributionForInstance(testInstances.instance(i));
```

```
      reslt +=  "ID:"+Integer.toString(i) +","+"Prediction:"+predictedClassLabel.toString()+",";
//System.out.println(i+ trueClassLabel.toString()+ predictedClassLabel.toString());
       // Print out the true label, predicted label, and the distribution.
//        System.out.printf("%5d: true=%-10s, predicted=%-10s, distribution=",
//                          i, trueClassLabel, predictedClassLabel);


       // Loop over all the prediction labels in the distribution.
       for (int predictionDistributionIndex = 0;
            predictionDistributionIndex < predictionDistribution.length;
            predictionDistributionIndex++)
       {
          // Get this distribution index's class label.
          String predictionDistributionIndexAsClassLabel =
             testInstances.classAttribute().value(
                predictionDistributionIndex);

          // Get the probability.
          double predictionProbability =
             predictionDistribution[predictionDistributionIndex];

          reslt +="Probability
"+predictionDistributionIndexAsClassLabel.toString()+":"+Double.toString(Math.round(pred
ictionProbability * 100D) / 100D)+" ,";
       }
       reslt += "\n";
    }
    return reslt;
}
}
```

Appendix E: 4 Weka API Web Service Code Snippet