National College of Ireland

Predicting outcome of cricket matches using classification learners

MSc Research Project
Data Analytics

Vivek Purkayastha
x16103572

School of Computing
National College of Ireland

Supervisor: Dr. Niall Moran

National College of Ireland
Project Submission Sheet – 2017/2018
School of Computing

Student Name:	Vivek Purkayastha
Student ID:	x16103572
Programme:	Data Analytics
Year:	2016
Module:	MSc Research Project
Lecturer:	Dr. Niall Moran
Submission Due Date:	11/12/2017
Project Title:	Predicting outcome of cricket matches using classification learners
Word Count:	5526

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature:	
Date:	11th December 2017

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
3. Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Predicting outcome of cricket matches using classification learners

Vivek Purkayastha
x16103572
MSc Research Project in Data Analytics

11th December 2017

Abstract

Cricket is an outdoor sport played between two teams of eleven players each. It is one of the most followed game, with a world of riches on offer. The goal is to predict the outcome of a cricket match and improve upon the currently reported accuracy of the learners. The paper found feature engineering and attribute selection plays a significant part in improving the accuracy of the classifiers. Following learners: k-NN, Naive Bayes and Random forest classifiers were used. Each of the models recorded major performance improvement, with Random forest performing best with an accuracy of 84 %, closely followed by Naive Bayes at 83% and k-NN at 82%.

1 Introduction

Cricket is an outdoor team sport. A cricket match is played involving two teams of eleven players each. The game is mainly played at domestic and international levels and in limited overs (one-day internationals and twenty-twenty) and test formats. This research concentrates on the limited overs internationals.

Game play is divided into two innings; with one team batting in the first innings and the other team bowling. Roles are reversed at the end of the innings. Which team bats first is decided by coin toss. A one day international is played with fifty overs bowled in each innings, and a twenty-twenty game is played with twenty overs for each innings. That apart, rest of the rules are the same. The team scoring the highest runs wins the game.

The sport is played mainly in the following ten countries, who are full time members of the International cricket council and are also referred to as the test nations; Australia, Bangladesh, England, India, New Zealand, Pakistan, South Africa, Sri Lanka, West Indies and Zimbabwe. Cricket is followed by a billion plus audience worldwide and growing in popularity rapidly and humongous amount of money being spent on the game, be it in telecast rights or sports betting.

In comparison to other team sports such as football and baseball, the amount of work that has been done on cricket in the field of analytics is less. The economic and social perspectives such as fan following, media coverage, etc. provides strong incentive to analyse the game.

Data used in the project is unstructured in nature. It was collected from `www.howstas.com` through web scrapping. Wunderground weather API was used to collect historical weather data, to have more detailed data available on weather than just whether it rained or not. Unfortunately, due to lack of ability to establish veracity about the data, it could not be used in the analysis. However, it must be noted that the disagreement between the sources may be due to first source reporting whether it rained during the play and wunderground API data being reported as an overall condition for the day.

The main goal of the analysis is to be build accurate and robust classification models to predict the outcome of a limited over international game. This research uses multinomial Naive Bayes, k-NN and Random forest classifiers. Effort is being put to improve the performance of these learners with the help careful feature engineering and attribute selection.

Current reported accuracies for the models are : 61% for naive bayes, 71% for random forest(Murdeshwar; 2016) and 70% for k-NN(Jhanwar and Pudi; 2016). Technological tools used in the project are CRISP-DM, Python, R, SPSS and Rapid Miner. All the tools were used as needed, however the project was implemented mainly with the help of python technologies.(Dwivedi et al.; 2016)

This report is divided into following sections: section 2. Related work, section 3. Methodology, section 4. Implementation, section 5. Evaluation and section 6. Conclusion

2 Related Work

This section discusses the literary work of other authors in the game of cricket in the field of data analytics. This section is divided into two main sub sections: 2.1 Attribute selection and 2.2 Learning models.

2.1 Attribute selection

Attribute selection forms the foundation for any classification task. The performance of the future learners depends heavily on attribute selection and feature engineering. In this section we will discuss about the feature selection techniques based on the works of other scholars.

Buursma (2010) in his work discusses about extent of historical data that should be considered for the analysis. It is advised to select a basic set of features and then change the amount of data considered for analysis. Look for the best performance achieved and decide on the historical point. Once this has been done then the next step is to proceed with the feature selection. Feature selection is performed by conducting an elimination process on the basic set of features. When the basic features are exhausted, then introduce new features and conduct the elimination process all over again until the final attribute set is arrived at. The approach to attribute selection is simple, elegant and based on hard facts, but the process of selecting a historical point is questionable. Because it works on the premise that the learner will behave similarly for the historical points for all the feature combinations. Rather a better approach would be to use statistical techniques such as independent sample t-tests to find a significance difference on a continuous scale over a period to indicate change in nature of the game.

Bandulasiri (2006) used logistic regression to explore the signicance of the features that could explain the outcome of a match. The research focussed on home eld advantage,

winning the toss, game plan (batting rst or elding rst), match type (day or day and night), and the eect of the Duckworth-Lewis method for matches shortened due to weather interruptions. The research found, home advantage to be an important factor. He also observed toss can be important but only in case of a day/night match, but proves to be disadvantageous for day games. This agrees with the work done by De Silva and Swartz (1998), that states home ground improves winning chance and toss is not important. However, De Silva and Swartz (1998) does not account for day/night matches.

Trawinski (2010) proposes a two-step approach; rst an intuitive approach and the next a more advanced approach. The rst step considers the results of the last three games for the two teams as the variables. The argument behind it is to back the team with a winning streak. In the second step consider more sophisticated variables and then select the best feature set from them. To do so the seven-dierent set of algorithms were used on the features to get robust results. The algorithms are CfsSubsetEval, ChiSquaredAttributeEval, ConsistencySubsetEval, GainRatioAttributeEval, OneRAttributeEval, ReliefFAttributeEval and SVMAttributeEval. However, this approach can over complicate the model. Also, since the attributes are largely independent, hence chances are that such a complicated models will incur unnecessary overheads.

Among the above discussed approaches, Buursma (2010)s approach is the most simple and elegant approach to the problem. It provides for a simple and evidence based model. Trawinski (2010)s approach of using winning streaks also inspire the use of dynamically generated attributes in the model. Anjali et al. (2015) inspires the use of tweets about matches to be used as an attribute, with the thought that a quantitative analysis on the tweets may be used as a supplement for human intuition. But the problem in this approach is the time limit placed by public version of the twitter API on historical tweets fetching.

2.2 Models

Murdeshwar (2016)s models use percentage split with 70:30 split and K-Fold cross validation with k=10. The algorithms considered were Decision Tree, Random Forest, Naive Bayes, K-NN classier. All of the models showed improvement when used with k-fold cross validation, except for k-NN. Random forest performed best with 71% accuracy and Decision tree and Naive Bayes models produced similar accuracy's of 63% and 61% respectively. Author makes no mention of the sampling strategy used.

Pathak and Wadhwa (2016) used Naive Bayes, Support Vector Machines, and Random Forest. Building on the work of Bandulasiri (2006) the model chooses toss outcome, home game advantage, day/night eect and rst batting as the attributes. The model uses percentage split with 80:20 split. SVM out performed Naive Bayes and Random forest slightly. However, in case of imbalanced data set Naive Bayes was the only algorithm that still produced results without any noticeable anomaly. The model achieved a highest accuracy approximately 62% with SVM.

Sankaranarayanan et al. (2014) attempts to predict the match outcome by predicting match gameplay. The model considers the historical as well as instantaneous match features. Given the instantaneous match data, model attempts to predict the remainder of the game. Maximum runs scored of the two innings gives the winner. The model breaks an entire innings into ten segments of ve over each. Given a match state in a segment n, the model predicts the runs at the end of an innings and then predicts the score of the next innings. The literature mentions prediction of home and nonhome runs

to predict match state. However, there is no such terms in cricket literature nor does the author explains them clearly. Model uses both historical and gameplay attributes. In that the model becomes unusable in the current research, because object of the research is to be able to correctly predict the winner before start of play.

Jhanwar and Pudi (2016) uses player modelling to predict the winner. Player potential is used to establish relative strength of one team against the other. Paper uses different methods to model batsmen, bowler and team. Using this methodology, k-NN achieved the highest performance with 70% accuracy and SVM performed worst.

From the above we can see that Random forest and k-NN seem to be performing well, with naive bayes being most consistent. Rish (2001) suggest that Naive Bayes along with working well for independent features also works well with functionally dependent features, such as one team batting first automatically implies the other team batting second.

3 Methodology

This section discusses the methodology followed for the research and the valuation of the outcome of the research conducted.

3.1 CRISP-DM

CRISP-DM stands for Cross Industry Standard Process for Data Mining. It is a comprehensive process model for data mining projects. This project uses CRISP-DM due to its independence from technology and industry sector(Wirth and Hipp; 2000). The

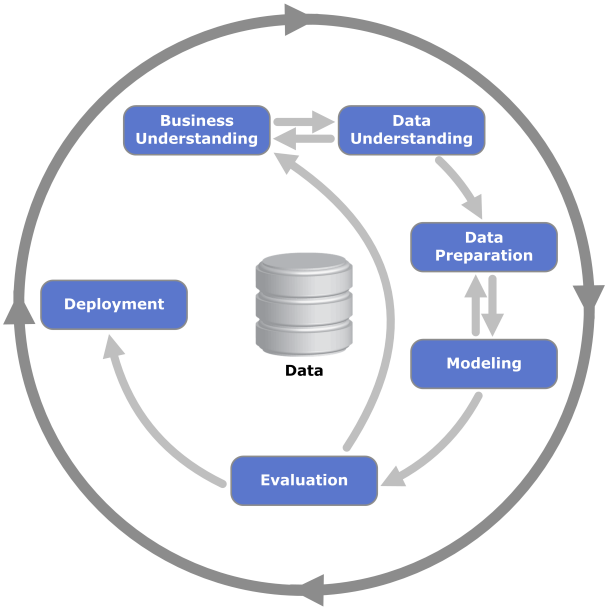

Figure 1: CRISP-DM model phases.

phases from Figure 1[1] in context our project are: -

[1] https://en.wikipedia.org/wiki/Cross-industry_standard_process_for_data_mining

CRISP-DM phases	Corresponding steps in project
Business understanding	Understanding business question(s), models and technologies to be used. See section 1
Data understanding	data collection, data pre-processing and exploratory data analysis. See sub sections 3.2, 3.3 and 3.4
Data preparation	Prepare data to be fed to model. See section 3.5
Modelling	application of different learners and calibrating their parameters to optimal values. See section 3.7
Evaluation	Model evaluation using different performance metrics; see section 5
Deployment	Models can easily be integrated to any enterprise level software system with some tuning.

Table 1: CRISP-DM phase mapping

3.2 Data collection

Data was web scrapped from www.howstat.com. This source contained match information such as date, teams (countries), ground, result, day and night match or not, toss winner, team batting first, etc. From this, only the factors that are reported before the play starts were considered for model building keeping in mind the objective of the research, rest of them were used for exploratory data analysis as applicable. However, it must be noted that a common problem faced while scrapping large amount of data from web, is with Timeouts. The same problem was faced with and overcome by making the system sleep for few seconds after collecting data for each year. To establish the veracity of the data, data instances were picked at random and information was compared by performing a manual search on the web.

3.3 Data pre-processing

This section discusses the steps taken to tidy the data and convert it to a format suitable for analysis. All the column names were checked for inconsistency and suitably handled to maintain consistency. All the observations with unwanted results, such as abandoned matches, conceded matches, cancelled matches, walkovers etc. were dropped. Unnecessary columns such as the one indicating row number and the one providing link text were dropped. The links were used during data collection to fetch game play details and were not required any further. Data types of the columns were converted to the required data types, since the information was retrieved by web scrapping hence all the columns were stored as object types. There were outliers in the data set, but were left untransformed because they are natural outliers.

Further it was made sure data was represented in a format that is suitable for analysis. Each row represents unique individual observations. Each of the columns in the data represents separate variables. It was made sure data table was stored in melted format rather than a pivoted format.(Wickham et al.; 2014)

3.4 Exploratory Data Analysis

The approach to exploratory data analysis was to first generate hypothesis from the data and then try to either prove or disprove the hypotheses generated. The thought was that in doing so it will help in the later stages with feature engineering without being influenced by the data available in the corpus.

3.4.1 What is the effect of location on match outcome?

The below figure visually analyses the difference in performance of teams according to the location.

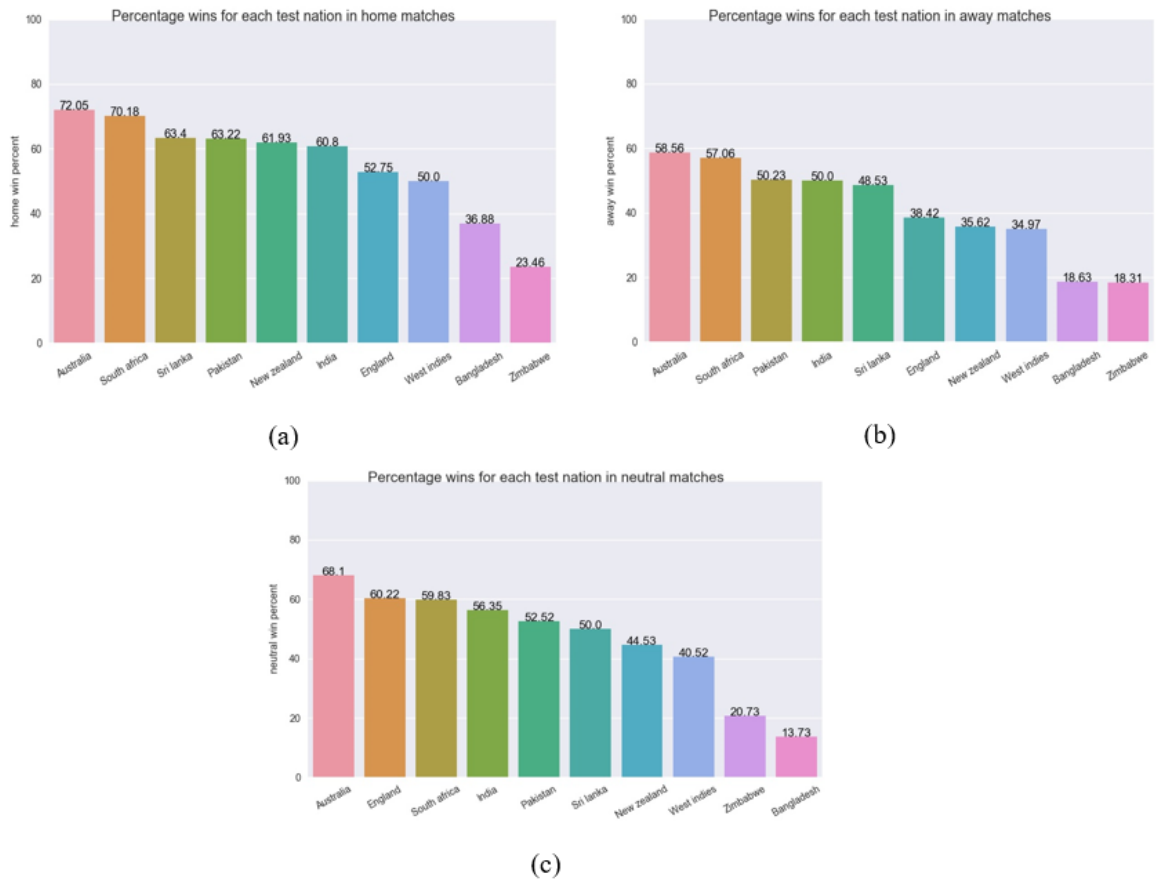

Figure 2: Location wise win percentage of each team; (a) home, (b) away and (c) neutral

We can observe from Figure 2 that, the percentage of wins for each team varies according to the location. Not only does the win percentage differ for the teams, but also the positional order of the teams changes according to location. Thus, it can be assumed that *location* is *important*.

3.4.2 Does innings order and rain effect match outcomes?

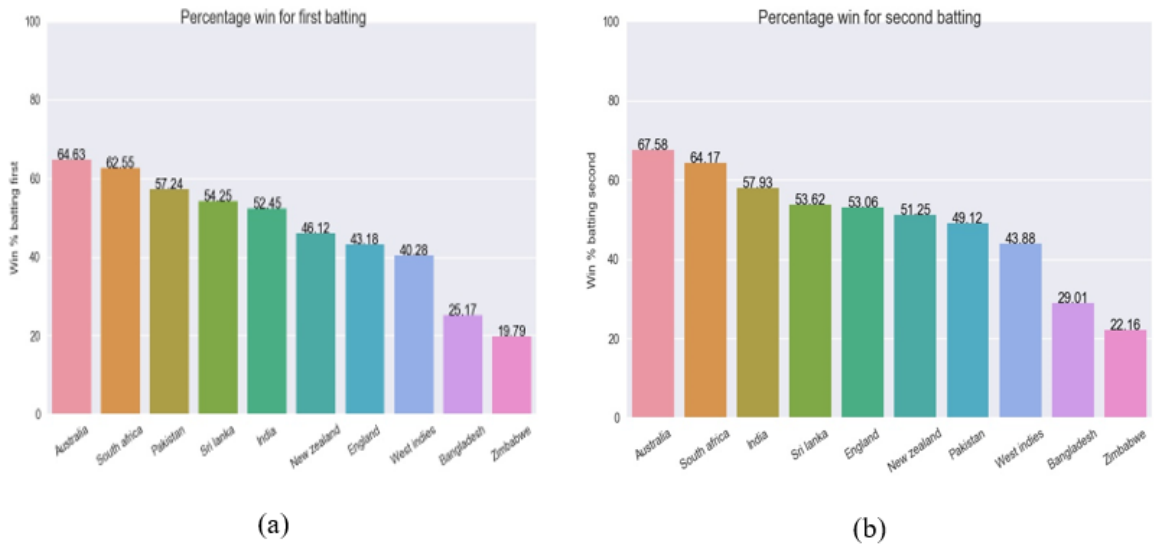

Figure 3: Win percentage according to batting innings of each team; a) batting first, b) batting second

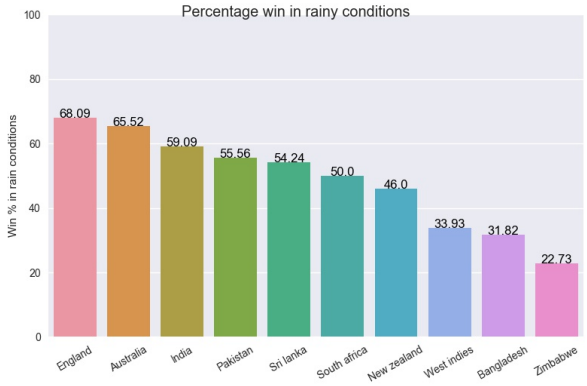

Figure 4: Win percentage of each team for rainy weather

In Figure 3 we can observe the change in win percentage and position of the teams according to the change in the innings in which the teams bat. Similar observation can be drawn from Figure 4. Hence, it is assumed that both innings order and rain are important factors influencing the game.

3.4.3 Is toss an important factor in deciding the winner of a match?

There is some ambiguity observed about the importance of toss from the works of other scholars as discussed in the related works section 2. Hence, a Chi-square test of independence was conducted between toss and winner attributes to establish the importance of toss.

Symmetric Measures

		Value	Approximate Significance
Nominal by Nominal	Phi	1.328	.000
	Cramer's V	.443	.000
N of Valid Cases		2475	

Figure 5: Chi-square test of independence between toss and winner

Referring to Cramer's V, value of .443 which is less than 0.5 and hence does not signify strong association between the two variables. Thus, it is safe to assume that toss is not a significant attribute.

3.4.4 Is there an important point in history from where data should be included in the model?

The thought behind this hypothesis is that if there was a change in the nature of the game due to any reason such as introduction of twenty-twenty format, etc. that could render data previous to that point in time meaning less.

To prove the above stated point an independent sample t-test was conducted on the scores, by splitting the corpus into sets at the year 2008. It was intuitively decided upon 2008 by observing the increase in number of twenty-twenty matches in 2007. The t-test produced a result of **p=0.01**, signifying a significance difference between the scores from before 2008 and that of after 2008.(Pallant; 2013) However on checking for the veracity of the test, it was found that similar results were obtained if the corpus was split at any annual value. Hence it was decided to include whole corpus for analysis.

3.4.5 What is the minimum accuracy that a learner should achieve for the given data corpus?

A robust approach to determine the classification baseline would be to first calculate a-priori probability accuracy and null accuracy, and then consider the maximum between these two and random chance as the baseline for the classifiers.

The probabilistic model is a simple model that tries to predict a match outcome using historical data based on the probability of winning of one team against the other team. In a game between team A and team B, the probabilistic accuracy is defined as below:

$$f(x) = MAX(prob(A), prob(B)), \tag{1}$$

Where,

prob(A) = probability of team A wining versus team B
prob(B) = probability of team B winning versus team A
f(x) = probabilistic accuracy

Null accuracy measures the accuracy that can be achieved by predicting the most frequent label (also called target) class in the data set for every instance label.[2]

[2]http://www.ritchieng.com/machine-learning-evaluate-classification-model/

Hence the baseline for the classifiers can be decided as: -

$$b = MAX(aP, aN, r) \tag{2}$$

Where,

b = base line, aP = accuracy of probabilistic model,
aN = Null accuracy, r = random chance = 50%

3.5 Feature Engineering and Attribute selection

It contained: a) engineering static features and b) engineering dynamic features. Dynamic features help to make the models not dependent on the teams playing the match, thus avoiding the problem of having an imbalanced data set.

In engineering **static features** the following operations were performed to extract information from the available data: -

countries column was used to generate information about the two teams playing the match, a feature to mark the matches that were played in the world cups was created to allow the learners to capture any information regarding the effects of a world cup match, date column was used to generate features regarding the day of week, month and year; although it is easy to comprehend that year may not be informational to the learner, but could be used for exploratory data analysis, ground details was used to create location, country and city features for the match being played at. For few of the ground names, multiple grounds existed at different locations. Such as Green park, New road, Indira Gandhi stadium, Windsor park, etc. But fortunately, it was possible to pin point the particular ground by using the game details.

Following **Dynamic features** were engineered: - *Win percentage*: the percentage of wins for each team generated dynamically till the match day, *Form*: form of the team in last three games represented as percentage; three was chosen because Zimbabwe played only three games in 2012 and choosing data beyond a year will not be a true representation of form, *Location wise performance*: performance of the team according to home, away and neutral locations and *Batting innings performance*: performance of the teams by order of the innings in which they bat. By creating dynamic attributes, the model is able to achieve independence from the particular teams playing the matches; thus avoiding the problem of having an imbalanced data set.

In attribute selection stage, according to Buursma (2010), feature elimination is conducted on the features to arrive at the final set of attributes to be included in the models. From the original set of twenty-three attributes, the finally selected attributes are: location, day and night, world cup match, rainy weather condition, team batting innings, format, win percentage, form, batting innings performance and winning team is selected as target label.

3.6 K-Fold cross validation vs Percentage split

In *percentage split* method, the data corpus is commonly divided into two mutually exclusive sub-sets; training set and test set. A learning algorithm learns or extracts knowledge, using the training set and uses the test set to measure the veracity of the knowledge extracted. Percentage split technique uses less computing resources but produces performance estimations with higher variance as compared to k-fold cross validation.(Wahbeh et al.; 2011)
In *k-fold cross validation*; he data corpus D, is divided into k mutually exclusive subsets or foldsd1, d2, , dk of approximately equal sizes. To create test and train; k-1 folds from D is taken as training set and the knowledge obtained from these dk-1 subsets are tested on dk. The process is repeated by considering each of the dk subsets as test set; while treating the rest of disjoint dk-1 subsets as training data. Finally, performance is measured (cross-validated) as an average of the performance from each of these k-folds. (Kohavi et al.; 1995) recommends the use of stratified ten-fold cross validation.

3.7 Models

3.7.1 Naive Bayes classifier

Naive Bayes is a Bayesian classifier, that works by predicting class membership probabilities to assign target class to test data instances. Naive Bayes is an eager learner. Studies prove that Naive Bayes is competitive in performance when compared to more complex and sophisticated techniques such as selected neural network classifiers and even outperforms others such as logistic regression, etc. given that the independence assumption holds true. It is known to hold true to its performance measures even when applied to large datasets.(Leung; 2007)

Naive Bayes has couple of assumptions: a) Predictors are independent of each other and b) a priori assumption; that past conditions still holds true. It is the former due to which Naive Bayes is called Naive. Our data corpus holds accountability for both first(Pathak and Wadhwa; 2016) and second assumptions. There are various types of Naive Bayes implementations. Some of them are Gaussian, Multinomial and Bernoulli. Since the dataset comprises of both categorical and continuous attributes, hence the paper uses Multinomial Naive Bayes.

3.7.2 Random forest classifier

Random forest is an ensemble method and is proven to be a powerful machine learning algorithm. Random forest classifiers build a set of mutually exclusive disjoint trees that vote for the best class to form the random forest. (Sulaiman et al.; 2015) Each of these trees are drawn at random from a set of possible trees and has an equal chance of being sampled. Random trees are considered to perform better in comparison to other ensemble methods such as bagging and boosting. They are faster than both bagging and boosting, more robust than boosting in respect to noise and produces performance at least as good as boosting without overfitting. Since random trees can be efficiently generated, when combined with collection of trees lead to accurate models. To produce random samples of training set for each tree, random forest builds each new training with replacement from the original training set. Random forest chooses to grow the trees as opposed to pruning them.(Oshiro et al.; 2012)

Oshiro et al. (2012) in their study suggest building sixty-four to one hundred and twenty-eight trees. This paper found **eighty-five** trees to be ideal for this data corpus through model tuning.

There are many other parameters that can be tuned to achieve a better performance from a random forest model. Some of the more important ones are: maximum number of features that each is tree is build using, minimum sample leaf size[3].

This paper uses model tuning to choose the right values for each of them. Further details regarding the same can be found in the section 5.

3.7.3 k-NN classifier

k-NN stands for k Nearest Neighbours. It is a supervised classification algorithm that work by classifying unlabelled observations by assigning them the class label of most similar observations or the nearest neighbours. Despite the simplicity of the idea behind the learner, the nearest neighbour classifiers perform competitively in comparison to some of the other more complex classifiers. The findings of this paper corroborates the same. k-NN is a lazy learner. It does not build models explicitly. This makes for a fast training phase but slow classification phase.[4]

k-NN algorithm utilizes nearest neighbour approach for classification. It takes a set of examples classified into several categories; the training set as input, and classifies the unlabelled instances of the test set into one of those categories. For each instance of the test set, k-NN works by identifying k records in the training data that are "nearest" to the particular test instance. The unlabelled test instance is then assigned the class of majority of the k nearest neighbours.

Distance between two data points can be calculated by using either Euclidean, Hamming, Manhattan, Minkowski, Jaccard, Mahalanobis, etc. distances. (Ye; 2013) Euclidean distance is the most commonly used distance metric among the others. Chomboon et al. (2015) in their research conclude that Hamming and Jaccard distances give lower accuracy and are affected by ratio of members in each class while other distances show similar accuracy. Hence, this paper uses **Euclidean distance**.

This paper found **k=13** to be the right k value for this data corpus by algorithm tuning.

3.7.4 Performance measures

To measure the performance of the classifiers metrics such as accuracy, confusion matrix, precision, recall, f1 measure, Cohens kappa and AUC score was used.

Accuracy gives the percentage of correct predictions made by the classifier. Precision and recall scores are calculated using true values and predicted values, hence first the confusion matrix was calculated, and then precision and recall scores were calculated. However, precision and recall alone are not very informative, they may fail to detect a poorly performing classifier from classifiers that perform well. Hence, we also used f1score, which is the harmonic mean of the precision and recall scores, thus penalizing any classifier with imbalanced recall or precision scores. Cohens Kappa is the measure of agreement between two random classifiers, after adjusting for agreement due to chance alone. AUC (Area Under Curve) score is a measurement with values between zero and

[3] `https://www.analyticsvidhya.com/blog/2015/06/tuning-random-forest-model/`

[4] `https://www.cse.buffalo.edu/~jing/cse601/fa12/materials/classification_methods.pdf`

one, a classifier with large AUC is preferred over a classifier with small AUC. (Seliya et al.; 2009; Altman; 1990)

4 Implementation

4.1 Data collection

Data used in the research was collected by web scrapping using R language with the help of rvest library. Timeouts issue was resolved using Sys.sleep() function. This is the only stage of the project which was implemented though R, rest was implemented using python. The reason for using R was the preference for rvest over other pythonic options such as beautifulsoup. Since some of the data was available as web table, hence rvest was used because it provides more gracious ways to handle web tables as compared to beautifulsoup.

4.2 Data pre-processing

Data pre-processing was performed using pandas python library. Files stored in the local file system was read through pandas read_csv() function and after pre-processing, was stored back using to_csv() function. Other pandas functions such as drop() to drop columns, astype() to change data types of the column, data frame apply(), etc. were used as needed.

4.3 Exploratory data analysis

Pandas was used for calculation as well as building hypothesis specific temporary data frames. To plot the graphs matplotlib and seaborn libraries were used. Statistical tests such as chi-square test of independence and independent sample t-tests were performed through IBM SPSS.

4.4 Probabilistic model and null accuracy

The probabilistic model works by calculating the probability of winning of one team against the other team. Model prepares a matrix of probabilities of winning of each team against every other team and uses the same matrix to predict future instances. The model was built using python math library and pandas.
Null accuracy measures the accuracy that can be achieved by predicting the most frequent label (also called target) class in the dataset for every instance label. Null accuracy was calculated using scikit learn library in python.

4.5 Feature engineering and attribute selection

For both feature engineering and attribute selection pandas along with pythons regular expression library(re) was used. re was used for pattern matching to extract information from raw data to create new features.

4.6 K-Fold cross validation

In the methodology section it was stated k-fold cross validation technique, with stratified folds was chosen over percentage split. k-fold cross validation was implemented using KFold module found in the model_selection package of scikit learn library. Ten folds were selected using n_splits parameter of KFold function set to 10. Scikit learn uses stratified sampling by default.

4.7 Models

All the three models were implemented mainly using scikit learn library. Along with scikit learn numpy and pandas were also used. LabelEncoder module from preprocessing package of scikit learn was used for encoding the target label, while for dummy encoding the rest of the categorical attributes pandas get_dummies function was used with drop_first parameter set to true to avoid highly collinear data. For Naive Bayes MultinomialNB module was used from naive_bayes package, RandomForestClassifier module from ensemble package for random forest and KNeighborsClassifier module from neighbors package was used for k-NN from scikit learn library.Model performance was measured with the help of modules such as confusion_matrix, cohen_kappa_score and classification_report from metrics package of scikit learn library.

5 Evaluation

Evaluation section discusses the evaluation approach and the rationale behind the decisions made.

5.1 Probabilistic accuracy, Null Accuracy and Base line

The table below records the null and probabilistic accuracy calculated along with the random chance.

Probabilistic accuracy	Null accuracy	Random chance
57.34	63.71	50

Table 2: Data for selecting baseline.

Thus, from 2, Base line(b) = **63.71**%

5.2 Static attribute selection

As discussed in section 3, feature elimination round was conducted to select the attributes. Following table gives the details of the performance achieved by combination of the static attributes.

Static Attributes		
Attribute set	**Accuracy**	**Remarks**
team a, team b, winner, location, day and night, toss, world cup match, rained, month, day, first batting, second batting, country, city, format	63.12	Poor performance; less than base line
team a, team b, winner, day and night, toss, world cup match, rained, month, day, first batting, second batting, country, city, format	63.03	Taking location out reduced accuracy
team a, team b, winner, location, day and night, world cup match, rained, month, day, first batting, second batting, country, city, format	63.5	Toss confirmed as not important
team a, team b, winner, location, day and night, world cup match, rained, month, first batting, second batting, country, city, format	63.61	day eliminated
team a, team b, winner, location, day and night, world cup match, rained, first batting, second batting, country, city, format	63.71	month eliminated
team a, team b, winner, location, day and night, world cup match, rained, first batting, second batting, country, format	63.93	city eliminated
team a, team b, winner, location, day and night, world cup match, rained, first batting, second batting, format	**65.32**	country eliminated; performance still not good
team a, team b, winner, location, day and night, rained, first batting, second batting, format	65.21	world cup eliminated;important
team a, team b, winner, location, day and night, first batting, second batting, format	64.7	rain eliminated; important
team a, team b, winner, location, first batting, second batting, format	64.45	day and night eliminated;important
team a, team b, winner, location, first batting, second batting	63.87	format eliminated; important

Table 3: Attribute selection for static attributes.

The attribute set marked in bold face in table 3 achieved highest accuracy for the model. Thus the selected set of attributes from the static attributes are : team a, team b, winner, location, day and night, world cup match, rained, first batting, second batting and format.

5.3 Dynamic attribute selection

Similar to static attribute selection, dynamic attributes were selected by model tuning.

Static and dynamic Attributes		
Attribute set	**Accuracy**	**Remarks**
team a, team b, winner, location, day and night, world cup match, rained, first batting, second batting, format, *win percentage*	65.95	performance improved slightly; still similar to base line
team a, team b, winner, location, day and night, world cup match, rained, first batting, second batting, format, win percentage,*form*	81.73	major improvement
team a, team b, winner, location, day and night, world cup match, rained, first batting, second batting, format, win percentage,form,*location performance*	81.25	slight decline in performance
team a, team b, winner, location, day and night, world cup match, rained, first batting, second batting, format, win percentage,form,location performance,*batting innings performance*	81.74	slight increase in performance
team a, team b, winner, location, day and night, world cup match, rained, first batting, second batting, format, win percentage,form,batting innings performance	81.96	location performance eliminated ; model accuracy improved
winner, location, day and night, world cup match, rained, first batting, second batting, format, win percentage,form,batting innings performance	**82.62**	teams dropped

Table 4: Attribute selection for dynamic attributes.

The attribute set marked in bold face in table 4 achieved highest accuracy for the model. Thus these attributes were selected to be fed to the models.

5.4 Selecting k value for k-NN

k	Accuracy
3	79.39
5	80.10
7	80.01
9	80.82
11	81.87
13	**82.34**
15	82.01
19	81.73

Table 5: Accuracy for different k values.

From table 5, k value is selected to be 13.

5.5 Tuning random forest

As discussed in section 3 there are various parameters that can be tuned to increase performance of a random forest model. The below sub sections document a few experiments conducted to tune the model.

5.5.1 Number of random trees to build?

Table below documents the experiment conducted:

Number of trees	Accuracy
65	79.72
75	83.41
85	**83.55**
95	82.7
105	**83.55**
120	82.74
128	83.08

Table 6: Accuracy for different number of random tree values.

From the table 6 we can see that we have same accuracy for both 85 and 105. We choose 85, because it is computationally less expensive.

5.5.2 Testing for optimal values for parameters max features and minimum sample of leaves.

Minimum sample of leaves	**Accuracy**
1(default)	81.72
10	**83.56**
50	82.78
100	81.5
500	79.15
700	57.39

Table 7: Optimal value for minimum sample of leaves.

Max features	**Accuracy**	**Explanation**
auto	**83.56**	max_features=sqrt(n_features), same as sqrt
log2	83.45	max_features=log2(n_features)
none	82.97	max_features=n_features
.2	83.19	20 % of data is to be considered
.5	83.32	50 % of data is to be considered
hline .75	83.31	75 % of data is to be considered

Table 8: Optimal value for max features.

From table 7 and 8 optimal value for minimum sample of leaves is chosen as 10 and *auto* for max features.

5.6 Discussion

This section discusses the findings of the research and presents the performance of the models using different performance metrics. From the sub sections 5.1, 5.2, 5.3, 5.4 and 5.5 it is noted that baseline for the classifiers is 64%, best set of features is: team a, team b, winner, location, day and night, world cup match, rained, first batting, second batting, format , optimal k value for the data corpus is 13 in case of k-NN, optimum number of random trees for the forest is 85, optimum value of minimum sample of leaves is 10 and for max features is auto.

	team_a_win_per	team_b_win_per	team_a_form	team_b_form	team_a_loc_win	team_b_loc_win	bat_first_perf	bat_second_perf
team_a_win_per	1							
team_b_win_per	0.043972099	1						
team_a_form	0.439093165	-0.227040791	1					
team_b_form	-0.215646972	0.412236206	-0.65504628	1				
team_a_loc_win	0.960854635	0.042425144	0.436282142	-0.22514265	1			
team_b_loc_win	0.041700866	0.968286289	-0.22862205	0.414428503	0.015303896	1		
bat_first_perf	0.504722403	0.499688853	0.118836423	0.070699364	0.487654259	0.490227036	1	
bat_second_perf	0.518219046	0.495124103	0.09661839	0.103675579	0.492460205	0.47122651	0.031485236	1

Figure 6: Correlation matrix for dynamic attributes.

From the selected attribute set, we can observe that along with static attributes one of the dynamic feature, location wise performance for the teams has also been eliminated. From Figure 6 we can see that it may be due to significantly high correlation of the attribute[team_a_loc_win/team_b_loc_win] with win percentage attribute[team_a_win_per/team_b_win_per], this has been highlighted in the figure.

5.6.1 Performance measure of the classifiers

Accuracy	**Precision**	**Recall**	**f1-score**	**AUC**	**Kappa**
82.62(±0.04)	0.83	0.83	0.83	0.89	0.65

Table 9: Performance metrics for naive bayes classifier.

Accuracy	**Precision**	**Recall**	**f1-score**	**AUC**	**Kappa**
82.34(±0.05)	0.82	0.82	0.82	0.91	0.64

Table 10: Performance metrics for k-NN classifier.

Accuracy	**Precision**	**Recall**	**f1-score**	**AUC**	**Kappa**
83.56(±0.01)	0.83	0.83	0.83	0.92	0.65

Table 11: Performance metrics for random forest classifier.

From the tables 9, 10 and 11 we observe that for the given data corpus all the models perform very competitively. Random forest only narrowly outperforms both Naive Bayes and k-NN. Naive Bayes and k-NN perform almost equally, very similar performance metric is being reported for both. Kappa score for all of the models is in the range of 0.6 to 0.8 showing good agreement. For good models we want the AUC scores to be high, as exhibited by our models. All of the models exhibit high precision, recall, good f1-scores. Hence, it can be concluded that our classifiers are good learners and robust.(Conger; 2017; Fawcett; 2004).

The research proves the importance of careful feature engineering. Including dynamic attributes substantially increased the performance of all the models. Among the known literature our models perform the best, with Random forest giving an accuracy of 84%, compared to 71% using random forest as reported by Murdeshwar (2016), for Naive Bayes the best accuracy reported was 61%(lower than even our baseline) (Murdeshwar; 2016); where as our model achieves 83% and Jhanwar and Pudi (2016) reported highest accuracy using k-NN at 70% where as our k-NN model achieved 82% accuracy.

6 Conclusion and Future Work

The purpose of the study was to take the three classifiers; namely, Naive Bayes, k-NN and Random forest and compare them. The intention was to try and improve the performance of the classifiers through careful feature engineering and attribute selection. To do so features (both static and dynamic in nature) were engineered from the row data after cleaning and transforming the data. Then attributes were selected by following an attribute elimination process. All the models showed marked improvement and performed

very competitively in respect to each other. Random forest performs the best among the three, but only slightly. See 12.

Classifier	**Last reported**	**Our research**	**Improvement**
Naive Bayes	61	83	22
Random Forest	71	84	13
k-NN	70	82	12

Table 12: Accuracy improvement in %.

Our research proves the importance of feature engineering in producing a high performing model. The improvement showed by the models after introduction of the dynamic attributes further proved the point. Dynamic attributes also help in making the model more robust by avoiding problems such as dataset imbalance, since it makes model generic by making it independent of the playing parties. One word of cautious; it is important to be careful to not include highly correlated attributes in the model, it may deteriorate performance of the model.

In comparison to other more complex approaches such as modelling the gameplay (Sankaranarayanan et al.; 2014) or Jhanwar and Pudi (2016)'s approach of modelling the players, and predicting a winner from comparative player strength, our model is vastly simple. Although it is considerably simple, but performs better than all other recorded models because it handles features elegantly. Not only are the models more accurate but also quite robust, each of them record high precision, recall, f1-scores and good Kappa agreements, see 5.6.

Although the research concentrates solely on the test playing nations, since the model is generic it can be fitted to handle matches for every nation. The model is robust enough to account for underlying strength difference of the teams, if non-test playing nations are included.

Another way to extend the model would be to provide weather data with better granularity. It was attempted to do so, but since historical weather data was needed, the API returned many null values. One way to handle it could be to use statistical imputation methods, but when too many blanks are returned it may not be the best approach. It can be hoped that with increasingly better infrastructure that will not be a problem in future.

References

Altman, D. G. (1990). *Practical statistics for medical research*, CRC press.

Anjali, S., Aswini, V. and Abirami, M. (2015). Predictive analysis with cricket tweets using big data, *International Journal of Scientific & Engineering Research* **6**.

Bandulasiri, A. (2006). Predicting the winner in one day international cricket, *J. Math. Sci. Math. Edu* **3**(1): 6.

Buursma, D. (2010). Predicting sports events from past results, *14th Twente Student Conference on IT*.

Chomboon, K., Chujai, P., Teerarassammee, P., Kerdprasop, K. and Kerdprasop, N. (2015). An empirical study of distance metrics for k-nearest neighbor algorithm, *The 3rd International Conference on Industrial Application Engineering 2015 (ICIAE2015)*.

Conger, A. J. (2017). Kappa and rater accuracy: Paradigms and parameters., *Educational Psychological Measurement* **77**(6): 1019 – 1047.

De Silva, B. M. and Swartz, T. B. (1998). *Winning the coin toss and the home team advantage in one-day international cricket matches*, Department of Statistics and Operations Research, Royal Melbourne Institute of Technology.

Dwivedi, S., Kasliwal, P. and Soni, S. (2016). Comprehensive study of data analytics tools (rapidminer, weka, r tool, knime), *Colossal Data Analysis and Networking (CDAN), Symposium on*, IEEE, pp. 1–8.

Fawcett, T. (2004). Roc graphs: Notes and practical considerations for researchers, *Machine learning* **31**(1): 1–38.

Jhanwar, M. G. and Pudi, V. (2016). Predicting the outcome of odi cricket matches: A team composition based approach.

Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection, *Ijcai*, Vol. 14, Stanford, CA, pp. 1137–1145.

Leung, K. M. (2007). Naive bayesian classifier, *Polytechnic University Department of Computer Science/Finance and Risk Engineering* .

Murdeshwar, S. (2016). *DATA MINING ON CRICKET DATA SET FOR PREDICTING THE RESULTS*, PhD thesis, Rochester Institute of Technology.

Oshiro, T. M., Perez, P. S. and Baranauskas, J. A. (2012). How many trees in a random forest?., *Machine Learning Data Mining in Pattern Recognition (9783642315367)* p. 154.

Pallant, J. (2013). *SPSS survival manual*, McGraw-Hill Education (UK).

Pathak, N. and Wadhwa, H. (2016). Applications of modern classification techniques to predict the outcome of odi cricket, *Procedia Computer Science* **87**: 55–60.

Rish, I. (2001). An empirical study of the naive bayes classifier, *IJCAI 2001 workshop on empirical methods in artificial intelligence*, Vol. 3, IBM, pp. 41–46.

Sankaranarayanan, V. V., Sattar, J. and Lakshmanan, L. V. (2014). Auto-play: A data mining approach to odi cricket simulation and prediction, *Proceedings of the 2014 SIAM International Conference on Data Mining*, SIAM, pp. 1064–1072.

Seliya, N., Khoshgoftaar, T. M. and Van Hulse, J. (2009). A study on the relationships of classifier performance metrics, *Tools with Artificial Intelligence, 2009. ICTAI'09. 21st International Conference on*, IEEE, pp. 59–66.

Sulaiman, H. A., Othman, M. A., Othman, M. F. I., Rahim, Y. A. and Pee, N. C. (2015). *Advanced Computer and Communication Engineering Technology: Proceedings of ICOCOE 2015*, Vol. 362, Springer.

Trawinski, K. (2010). A fuzzy classification system for prediction of the results of the basketball games, *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*, IEEE, pp. 1–7.

Wahbeh, A. H., Al-Radaideh, Q. A., Al-Kabi, M. N. and Al-Shawakfa, E. M. (2011). A comparison study between data mining tools over some classification methods, *International Journal of Advanced Computer Science and Applications* **8**(2): 18–26.

Wickham, H. et al. (2014). Tidy data, *Journal of Statistical Software* **59**(10): 1–23.

Wirth, R. and Hipp, J. (2000). Crisp-dm: Towards a standard process model for data mining, *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, pp. 29–39.

Ye, N. (2013). *Data mining: theories, algorithms, and examples*, CRC press.