# Predicting Outcomes of Cricket Games

**ABHINAV MISRA**

## Introduction:

The purpose of this project is to learn how to predict the outcome of a cricket match. Many factors influence the outcome of a cricket game. The key point in predicting match outcomes is the assembly and analysis of match statistics in a meaningful way. In this report, I explored the use of machine learning algorithms to analyse those statistics and correctly predict game outcomes. I experimented with different algorithms, viz, Naive Bayes, Logistic Regression, Multilayer Perceptrons and Support Vector Machines to test the available data. I also explored various feature engineering techniques to improve the performance of the algorithms. The features for this study are extracted from the data taken from sports website:http://stats.espncricinfo.com/ci/engine/records/index.html?id=6;type=team
The site contains the records of all the one-day Internationals played by India since 1970's. The games played between 1974-2009 are taken as train data while the games played between 2010-2013 form the test set. The accuracy of the prediction is calculated for all the matches played by India against other countries.

The rest of the report is organised as follows: Section 1 details the feature extraction procedure from the dataset, Section 2 explores the performance of different machine learning algorithms on the extracted features, Section 3 discusses methods that can be used to enhance the performance of algorithms and Section 4 concludes the report with a brief discussion on future work.

## Feature Extraction:

For the baseline system, I considered the following features/attributes that affect the outcome of a game significantly.

1) **The team India is playing against:** There are some teams against which India has a really good winning record, while there are some against which it hasn't performed well. So the team it is playing against is an important attribute to be considered in predicting the outcome of the game.

2) **Whether the game is being played on home turf or outside the country:** India is notorious for underperforming on grounds outside the country. So, if the match is being played inside the country there are high chances of it winning.

3) **Whether the team is chasing a target or batting first:** Depending upon the type of players in the team, some teams are better in chasing a target, while some are good in setting up the target. For example, if a team has good bowlers it is more likely to win if batting first. Same way,

if the team has good batsmen, the chances of it winning are more while batting second.

All these features were extracted from the data taken from the ESPN website. The data was divided into two text files: train_file.txt and test_file.txt. A perl script was written to convert each of these text files into libsvm format file containing the above mentioned features. The features were discretized to take on values from 1 to 10. The way these values were given is described as below:

The file: country_codes.txt contains the codes given to each country (ranging from 0-9) India played against . The file: Places_in_India.txt contains the names of all the cities in India, where a match was played. Any city in the data other than the cities in this list were treated as foreign turf. All the Indian cities were given a code of 1, while all the other foreign cities were given a code of 0. In the data files, if the winning margin is written in the form of “n wickets”, (where n ranges from 1-10), it means the team won while batting second and was given a code of 0. Similarly, if the winning margin is in the form of “n runs” (n ranges anywhere between 1-300 ), it means the team won while batting first and was given a code of 1.

After converting the data to features following algorithms were run on it.

## Algorithms:

Libsvm implementation of SVM was used along with weka implementations for the following algorithms.
(The libsvm format file generated in the above step was converted to arff file format using a perl script for weka algorithms)

| **Algorithms** | **Accuracy** |
|---|---|
| SVM (Radial Basis Function, C=2, gamma=8) | 61.06% |
| Multi Layer Perceptron (No. of hidden layers=1,Number of nodes in hidden layer=3, number of epochs = 500, learning rate=0.3 ) | 60.18% |
| Multi Layer Perceptron with Adaboosting (number of iterations=30, light threshold=100000) | 60.18% |
| Multi Layer Perceptron with Bagging (number of iterations=30) | 50.44% |
| Logistic Regression | 57.52% |
| Logistic Regression with Adaboost (number of iterations=30, light threshold=100000) | 57.52% |

| | |
|---|---|
| Logistic Regression with Bagging (number of iterations=30) | 52.21% |
| Naive Bayes Multinomial | 58.41% |
| Naive Bayes Multinomial Adaboost (number o iterations=30, light threshold=100000) | 58.41% |
| Naive Bayes Multinomial Bagging (number of iterations=30) | 56.64% |

So the best result of 61.06% was obtained with SVM. From the year 1974-2013, if I calculate the winning percentage of India, it comes out to be **50.12%**. So, 61.06% actually makes sense and is not just based on majority vote or random selection.

## Feature Engineering:

Now, I resort to some feature engineering techniques, in order to improve the results. I am getting the best result using non-linear SVM. So, one of the first things that I did was to square the features and make them non-linear. It was expected that then, the SVM will give a better result. But this technique didn't work, and an accuracy of 58.41% was obtained, which is even lower than the actual accuracy.

Then, several different types of other codes were given to features apart from the values between 1 to 10, but none of these methods improved the accuracy.

Then, I turned to **Sabermetrics**. Sabermetrics is the term given to empirical statistical analysis of baseball. It involves using statistical analysis for evaluation of baseball measures like batting average , pitcher wins etc. One of the Sabermetrics is **Log5**.

Log5 is a formula to estimate the probability that team A will win a game, based on the true winning percentage of Team A and Team B. The basic formula is:

$$Win\%(Avs.B) = \frac{A - A * B}{A + B - 2 * A * B}$$

Where Win% is = the probability that A defeats B in a single game. [1]

So, individual winning percentages were calculated for each team for all the matches played since 1970s. The file winning_percentages.txt contains these percentages in front of the team name. These percentages  were added to the feature list and a relative improvement of 11.6% was obtained using **Bagging over Decision Stumps**, with 30 iterations. Hence, our new improved accuracy comes out to be **68.14%.**

*[1] Source:Wikipedia*

## Conclusion and Future Work:

A new set of features were extracted from the available data on the cricket matches played by India since 1970s. These set of features gave an accuracy of **68.14%** in predicting the outcome of a game. In the future, more work can be done to include other features like comparing the player statistics of each team competing against each other, the number of matches team won in recent time (current form of the team) etc. Such a system would be of interest to cricket followers and could have implications in betting and selection of players in the team.