Analyzing and Predicting Outcomes of Cricket Matches

Parikshit Shembekar Dipesh Virkar

ABSTRACT

This work proposes to analyze cricket matches and to make predictions on the match outcomes and compare the different machine learning models. This data set consists of cricket data from 3994 matches consisting of international and domestic tournaments from different leagues. The data is provided in YAML format. CRISP-DM methodology has been followed for this project.

1. INTRODUCTION

Cricket is a game played with bat and ball wherein two teams, each of 11 players, compete against each other on a 20 metre pitch by scoring runs and taking wickets. At a given time, the batting team has 2 players on the pitch and the bowling team has 11 players on the field one of which is a bowler. Each player on the batting team scores runs by running between the pitches or hitting boundaries(4 or 6 runs). The team score is the total score of each individual player. A given match is of selected number of overs, that is each team will have a maximum number of balls to play for. The aim of the batting team is to score as many runs as possible. The bowling team aims to restrict the score of the batting team, as well as, looks to take wickets. Wickets are taken by getting the batsman caught, bowled, stumped, leg-before-wicket, or run-out. Each team has 10 wickets, so the aim of the bowling team is to get the 10 wickets as soon as possible. The goal is to outperform the other team by scoring more runs.

After Soccer, Cricket is the second most popular sport in the world. There are around 104 countries playing this sport officially being a part of International Cricket Council. Cricket is a huge business market, especially in India. Hence analyzing this game, could be considered one of the most interesting data mining projects. There are hundreds of statistics to compare and analyze from, each being important in their own way. Cricket being a big money market, there is a great demand for prediction of the match winner, which can be used for betting purposes.

Cricket data of various leagues and matches is available in different formats in various sports websites like ESPN, Yahoo Sports etc. The data is also available in a cleaned format on various other platforms like [2]. A successful model would be able to determine the results of the match or performance of individual player with a better accuracy.

The objective of this project is to explore the viability of predicting the outcomes of a match for a team, and also explore the various challenges involved in predicting the performance of individual players. We have compared the performances of various classification algorithms like Logistic Regression Classifier, Decision Trees, Random Forests, Support Vector Machines, and Deep Neural Networks. We have used Python 3.6 for our project. We have implemented our Machine Learning algorithms using sklearn package. To implement the Neural Network we have used Keras with Tensor Flow as backend.

2. DATA UNDERSTANDING

This data was obtained from [2] which is a data repository containing ball by ball information about around 3994 cricket matches among different countries and teams in CSV and YAML format.

In the data set we had match results of different leagues including the International ODI, International T20, Test Matches. This information is given in the form of Excel Spreadsheet and YAML. It contains information like match results, the umpire of the match and most importantly ball to ball information of every match. This information was about the current batsman and bowler, how many balls were played, runs scored at each ball, wickets taken at each ball.

We believe we have sufficient background and knowledge in in this game, and hence would be able to take the decisions of considering and not considering features while predicting or analyzing the results.

Thus, the first step of CRISP methodology of Business Understanding and Data understanding was completed.

3. DATA PREPARATION

Data cleaning is one of the most important steps in data mining. It involves processing of the given resource data by correcting or removing data which is redundant and fixing the missing or incomplete data with appropriate values. This may involve modifying the original data resource to fit in better for specific data mining needs.

The data from the website was available in two formats, CSV files and YAML files. As CSV files are easier to parse and operate on in Python, we decided to use the CSV files.

But unfortunately, during data exploration we realized that the CSV files were broken. The data present in the files were incomplete as only first innings data was available, which was also repeated in the second innings data. Much of the second innings data was missing. Operating on these files, would have given completely wrong results. The other option was to use YAML files. YAML stands for YAML Ain't Markup Language, which is a human readable data serialization language. It is used to store data in a structured format. It provides same functionality as a JSON or XML file. These files have specific tags, for each component, and the information related to each component is stored in the corresponding tags. It creates mappings for each component, and has values which can be string, numbers, arrays, lists.

The YAML files were complete. An example for mapping can be Winner: India. In this example winner is the key and India is the value. A key can store a value or a list as well. For example, deliveries: -0.1 -0.2 -0.3. In this example deliveries in the key have sequences 0.1,0.2,0.3. Sequences start with –(dash). Taking the advantage of this syntax, we can load the entire YAML file into a dictionary in Python (also known as HashMap in Java). After loading the file into a dictionary say mydict, one can easily access the components by using the dictionary. For example accessing deliveries as mydict['deliveries'][0] this will let us access the data stored in deliveries: 0.1.

Thus, the second step of CRISP methodology of Data Preparation was completed.

4. OBJECTIVES

Our data set consists of ODI matches and T20I matches. It contains ball by ball data of every match, giving information about runs scored, wickets taken, type of wicket taken, bowler and batsman, number of overs for every ball. This data set can be used to analyze and view team information and performance over the years, we can also visualize how a batsman, or a bowler has performed or statistics about the highest run scorer or highest wicket taker.

We decide to go one step further with this data, we wanted to leverage the fact that this data set gives us ball to ball information of every match. This was a crucial information, with which multiple predictions and conclusions were possible.

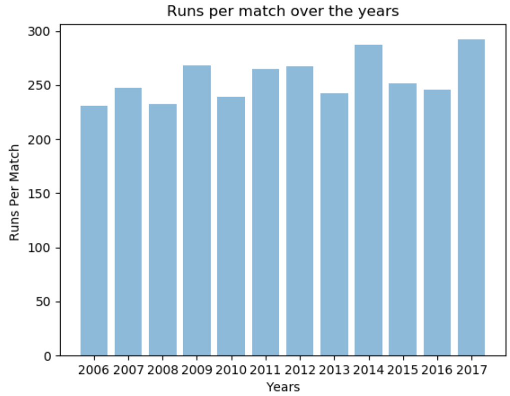
As we had data about the team matches played, it was possible to predict the winner based on past matches. Many features given in the data set like venue, bowler to batsman ball by ball performance, can be used to somewhat predict the winner of the match. Thus, we could predict the winner of the next International ODI match 3or T20 match based on the right feature selection.

5. CURRENT WORK

1. Data Visualizations

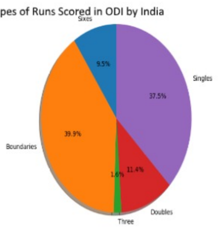
Average Runs scored by Indian Cricket team in ODI over the years

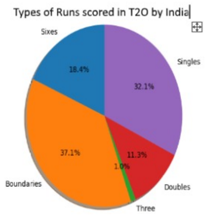
We plotted the average number of runs scored in a match for each year from 2006 to 2017.



One can expect the performance of the teams to improve over the years. As the competition among the teams increases, each team tries to perform better than the other team in some or the other way. In earlier years, due to less competition and less number of teams participating in the leagues, the average scores were less than 300. But now in the recent years, a minimum of 300 runs are expected to win a ODI match. If we compare the average runs scored per match over the years, we expect an increasing trend.

Type of Runs scored by India in T20 and ODI

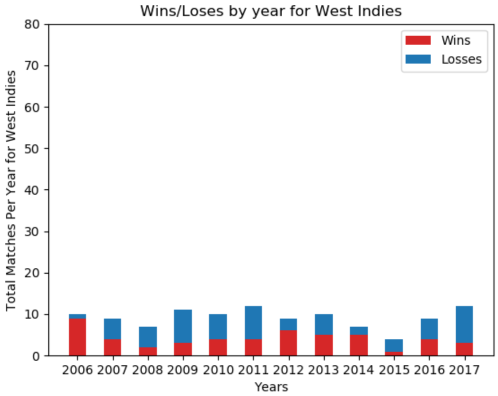




ODI (One Day International) and T20 (Twenty –Twenty) are two different cricket match formats. In ODI, each team gets a total of 50 overs (500x6 = 300 balls) to bat. Whereas in T20, each team gets a total of 20 overs (20x6 = 120 balls) to bat. As the number of overs is less in T20, it's obvious that the batting team would try to score more runs in each ball to get a higher score. So, a batsman would try to hit boundaries (4 or 6 runs) more than single or doubles. The overall play becomes more on the offensive side rather than defensive side. If we compare the type of runs scored between these two types of matches, we can expect the percentage of boundaries and sixes in T20 to be more than that of the ODIs.

Comparing the type of runs scored in ODI and T20 by the Indian Cricket team we can see that, as expected the percentage of sixes scored in T20 is higher than that of ODIS. We can infer that, T20 follows more offensive play than the ODIs

Wins/Losses of West Indies Cricket Team over the years



In the above graph of wins/loses of West Indies over the years, we can see a decrease in the number of wins for the West Indies from the year 2006 to 2017. We can see that West Indies played 11 matches in the year 2006 and won 9 of them, but in the year 2017 it played 12 matches and lost 9 matches.

2. Predict which of the given two teams would win

For this we parsed our data so that we could get ball by ball analysis for each each match. In our code we accept two teams for which we want to predict the winner. Then we extract many features from our data like, we extract the number of runs scored by each team in each of their earlier (training data) matches, then categorize that into four different features like if the runs greater than 300, are they between 250 and 300, are they between 200 and 250, and are the runs less than 200. Similarly, we have done this for wickets as well, by creating four classes for wickets between 0 and 10. We found top 3 batsmen in terms of runs for each team and then checked if each of the batsman were playing in the match that we want to predict.

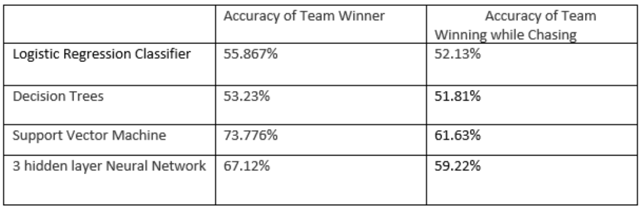
Our other features were like the toss decision, toss winner, the over and ball number and the number of runs scored on that delivery, the stadium name. This was our feature vector. The number of matches between these two teams was our training data. We split this into two sets training data and test data. We found the class labels for all our training files, this class label was 1 if the first team won, or 0 if the first team lost. Team1 or Team2 is taken according to user input. Similarly, we found the feature vector for our test data. We validated multiple models on our data, like SVM, Decision Trees and Neural Network, and finally compared the results of these algorithms

3. Predicting if a given team could win while chasing

In this we needed to keep track of how each team has performed while batting first against our target team. We maintain a similar feature vector as our first case. In our prediction stage, we do not check the 2nd innings score of our team, and on the basis of team batting first's runs scored we make the prediction.

6. RESULTS

These results are for India vs Sri Lanka matches.



	Accuracy of Team Winner	Accuracy of Team Winning while Chasing
Logistic Regression Classifier	55.867%	52.13%
Decision Trees	53.23%	51.81%
Support Vector Machine	73.776%	61.63%
3 hidden layer Neural Network	67.12%	59.22%

We have compared the performances of Logistic Regression Classifier, Decision Trees, SVM, and Neural Networks. We can see that SVM gave us the best results, even better than neural network, the reason behind this is that, our training data is very less, and for neural networks to work best we need large training dataset. Thus, it is reasonable to see than SVM has outperformed neural networks in our case.

Our best accuracy is around 73% which is very good for a game like cricket where a lot of uncertainties are involved and the chance accuracy is of 50%.

Thus models of data were prepared and analyzed, completing the third and fourth step of Modelling and Evaluation

7. FUTURE WORK

Initially, in our project proposal phase we had planned to make prediction for a batsman vs bowler, that is for a given batsman and a given bowler, predict what would happen on the next ball, would it be four, six or a wicket? But as we got into the data exploration phase, we realized that our data was unsuited for this task. To make such kinds of predictions we need ed quantitative information of the swing, spin of the ball, the length of the delivery, the pitch condition. We have stadium information, but even that information is insufficient as, the pitch and outfield quality depends upon the current weather, which is missing in our data. Sites like ESPNcricinfo [1] maintain such data, so, parsing data for each match could give us this dataset.

8. CONCLUSION

Generally, cricket matches are very hard to predict, as right from the toss and the weather condition, a lot of uncertainties are involved. One can easily predict the winner of the match at the beginning with a 50% accuracy as, either of the team has the equal chances of winning. Also, the circumstances of the match like the weather, pitch conditions change over the course of the match which are difficult to account for.

We attempted to observe the performance of machine learning models on predicting match winners, and we have achieved a score of around 73% which is way better than chance accuracy of 50%. Creation of better datasets which takes into account various details like length, speed, swing of each delivery will enable player performance prediction for a given delivery.

Thus all the steps of CRISM-DM methodology were successfully followed and implemented.

9. REFERENCES

[1] ESPN. ESPNCricinfo.
`http://www.espncricinfo.com/`.

[2] S. Rushe. cricksheet. https://cricsheet.org/downloads/#experimental.

Appendices

Appendix A

Parikshit was involved in implementing the neural network in Keras.

Parikshit, along with Dipesh were responsible for implementing the team winner prediction algorithms.

Dipesh was responsible the visualization of the data. The documentation of the project was done by both of them equally.

Appendix B

The pre-requisite packages to run the code are:

Python - 3.6 Keras Version - 2.0.2 Tensorflow - 1.3.0 matplotlib, sklearn, and numpy

The project contains three Python files, namely Visualization.py, PredictTeamWinner.py, TeamWinnerChase.py

All these files can be run by directly, but the user will be asked for the prompt where he/she is supposed to enter the country from the given list.