Cricket : An Unfairly Random Game

Hitesh Mohite
Department of Computer Science
Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, NY 14586
hm7181@g.rit.edu

Abstract—Cricket is one of the most popular games in the world and is only after Soccer[1]. It is relatively a long and complicated sport with respect to other sports with many factors effecting the game at any point during its course. While, this game comes in three different formats - Test, One Day International(ODI) and Twenty-20 International(T20I), this paper analyses the ODI format of the game. There are not many resources that have been published in the past explaining or analyzing the game of Cricket in any format. This paper focuses on two aspects of this particular format - Predicting the winner of a game along with the margin of victory exploiting the Machine Learning algorithms - Regression and Classification, and by using python libraries - Pandas, Numpy, Scikit-learn, Beautiful Soup and Matplotlib.

Index Terms—One-Day International; Regression; Classification; Scikit-learn; Beautiful-Soup;

I. INTRODUCTION

Often, it is the team that is going into the contest with better form of its players in the recent past with the same squad has the upper hand in the game of Cricket. Although, Upset has been a rarity in this game for a long time, this notion has been subjected to change since 2007 when both India and Pakistan were ousted in early group stages from 2007 World Cup after shocking defeats to the then minnows, Bangladesh and Ireland respectively. Due to the increased uncertainty and randomness in the game since the advent T20I cricket World Cup in 2007, upsets have been on the uprise and the certainly losing team have been able to upset a particular game from any point in the game leading to the unpredictability. This report analyzes about the predicting the game with the margin of victory with respect to its datasets and its features.

A. Game Details

The ODI format of the game is held between two teams and it starts with a toss. Each side will play for 50 overs for their batting and bowling. Each over has 6 balls to be bowled. Depending upon the outcome of the toss a team will choose to bat or bowl leaving the other team to either bowl or bat respectively. Each team plays the game with 11 players in their side. The team batting first scores runs and sets up a target in their allotted 50 overs or before being bowled out. The team batting second has to chase down that target with in the allotted 50 overs without being bowled out, in order to win the game or they lose.

B. Dataset

There are completely three Datasets used for this project - Team Data, Batsman Date and Bowler Data. All these datasets comprise information about all the games played since the inception of ODI Cricket. There were completely 4007 ODI games played so far, but the Team Data consisted data about 3867 instances which had a final outcome for a game. The outcome could be either of a win, loss or a tie. The Batsman and the Bowler had 88296 performances for each of them, and this includes games which had no outcome. All of this data has been extracted from official website of stats web page of EspnCricInfo.com.

II. RELATED WORK

There are quite a few papers available which concentrated on different ideas and tried to predict the winner of a cricket game, first innings score, predict score for a duck-worth lewis affected match or determining the game’s winner at a certain point. To help the report gain better approach and results, a few of them were analyzed. Stylianos and William[2] have Machine intelligence concepts to predict the winner of an English county game. They have implemented the model using Naive Bayes and Random Forest algorithms and have included and excluded their feature set to get better accuracy rates for games between two specific teams. They had an accuracy of 64.5 percent for 2009 to 2014 time period. They have included both team and player features into their feature set to train the model and have implemented Chi-Square and Pearson correlation scores for categorizing the appropriate features.

Another insight into the game prediction was provided by Madan Gopal[3]. The author explains the factors that were considered to derive features in defining a player’s statistics. The batscore and bowlscore features have taken many features such as average for a batsman and runs conceded by a bowler along with their strike rates. Implementing this approach reduces the feature set avoiding over-fitting of the model.

As discussed by Munir, Hasan, Ahmed and Md Quraish[4], implementing Decision trees and Multivariate Regression algorithms. They have implemented these algorithms at three different points during a game for each of innings for a specific team to predict its winning possibilities.
III. CHALLENGES

While there were many challenges during the course of its implementation, the major ones were associated to the Datasets - Data collection, structuring the data, inconsistency across all the datasets, organizing the input and output scenarios and Feature selection.

A. Data collection

The Data collection and preprocessing has been the tedious amongst all the challenges, as there were a very limited sources for collecting the data. The sources which had data either had partial set of data or had data about different format. Therefore, scrapping the web data from the Cricket’s major websites was the best possible way and this was achieved by using Beautiful Soup library. The scrapping of web data demanded a lot of time as the code written to automate the extraction procedure still has to load every page in web, parse it later and then extract the required data. This was done differently for each of the datasets mentioned before. Despite that, the data gathered for the team is relatively little as the 3867 instances of the game were spanned across 26 nations. Similarly, the 88296 instances in each of the other datasets for all the players was shared among 2412 players leading to problems while making a model learn from respective instances.

B. Inconsistent data

As the data scrapped from web was in its crude form, it had a lot of inconsistencies. The data was not in its standard format and had many special characters attached to it. These were handled by performing Regular Expression operations appropriately. Some of its data was partial in the Batsman and Bowling datasets due to early completion of games. This data was was removed from the dataset as it did not help in training the model which in turn resulted in shortening the length of the dataset considerably.

C. Structuring the data

The structuring of data was a difficult task as the data was in its raw form and getting it standardized into the code readable format was a challenging. There were many attributes that came with the dataset when extracted among which only a few were important. The idea of structuring it in a specific format was subjected to many changes before getting the data to be ready for training. There were many scenarios that were considered as it had impact on feature selection.

D. Input and Output scenarios

The input and the output scenarios for this project were tricky in the beginning of the project as the data was not ample and as it was subjected to changes constantly, leading the scenario vulnerable to change. But this was overcome once the datasets were finalized along with the implementation models and its features.

E. Feature selection

Feature selection was a critical subject amongst all the challenges as there was not much data to implement a model. Involving a lot of features was over-fitting the model and less features would mean there is a scope for bias. Therefore, there was enough thrust on what features and how many of them will go into the model. As there was very little data, there was not much freedom to incorporate more powerful algorithms such as Support Vector Machines(SVM), Neural Networks(NN) or Hidden Markov Model(HMM). This observation was a critical as it turns to be decisive with respect to implementation and accuracy of the model.

IV. DESIGN

The Design of the model has been consistent throughout the project and has been classified into six steps as it shown in the Fig 1. The initial step is to identify the sources of the data. After going through various resources, it was understood that there were not many sources which can provide all the ODI format data available. Therefore, it was identified that the data has to be the raw data from the web pages of the site mentioned. The scrapping of data is a time taking procedure as it involves loading of hundreds of web pages during its execution time. This has been implemented using a python library called Beautiful Soup. It parses the web page into a object on which one can perform extract operations to gather data. All of this was implemented in a python library called Pandas which is best known for the exhaustive set of data operations. The next step is to make the data consistent by removing any kind of abnormal data stored during the process of extraction. This was achieved by the use of regular expressions but has taken considerable amount of time as the data was extensively random. This step was followed with standardizing the unstructured data into a specific format. This step involved numerous tasks such as fine-tuning the attributes, removing unwanted rows categorizing the data accordingly with respect to players and teams and also feature extraction. After structuring the data, the models were decided, trained and tested them over the datasets. Given the size of the data, not many models were tried in this approach but analyzing the output was a crucial step.

V. IMPLEMENTATION

The implementation has been accomplished in three different approaches viz., Team, Batting and Bowling prediction models. The design remained same for all the models except for the feature extraction and data structuring which changed accordingly with their respective datasets. Let us look into each of the implementations.

A. Data cleaning

This is the only common step for all the prediction models in which the cleaning of data happens. Most of the challenges discussed before are involved in this phase. All the three datasets have its own cleaning and standardizing tasks accordingly. The initial datasets that was collected from the
web was in its crude format and had many anomalies. There were many missing values to begin with which were removed from the dataset. There were a few of the attributes such as score, opposition in the team data, runs in the batsman dataset which had combined information such as wickets being added to the total score and extra information such as having special characters in the numeric data which was needed for feature development and it was inconsistent across all the rows. These were dealt with care to make it consistent by splitting the whole raw data columns into appropriately needed columns for the features. All of this data was also formatted with respect date specific information. The player data of both the batsman and bowler had names and were inconsistent thus leading to the mismatch of number of instances between two datasets. This was rectified by implementing regular expressions across all such attributes. All the particular specific player instances were stored in individual data-frames in pandas. As all the data was available in string format, it was converted into numeric format which is compatible to Scikit-learn library. Any redundant and duplicate information was removed making the instance of the dataset to be unique. Each team information was stored separately and were operated separately for feature extraction.

B. Team prediction model

As a result the Data cleaning step, the dataset was completely standardized to perform feature extraction as shown in Figure 2. There were seven features in complete as inputs for the Logistic regression model - Match Order, Current form, Recent form, Cumulative win percentages for each teams.

- Match order: This feature helps identify the sequence of interactions between two teams involved in prediction.
- Current form: This feature will have information about a team’s current form, it depicts the number of wins a team had in its last 10 encounters.
- Recent form: This is similar to the information of current form but it tells about the win percentage in its last 5 interactions.
- Cumulative win percentage: This is win percentage a team has achieved over all the last matches till that particular instance against that team. This tells the team’s form against a particular team since their last interaction.

This model trains over all the training instances with these input features. The model is then tested over the test dataset to find the accuracy and then will be provided with the input features for next unseen game to predict the winner between those two teams. Each teams data is stored separately and features are extracted for each of them.

C. Batting prediction model

As shown in Figure 3, similar to Team prediction model, after the cleaning and standardizing the dataset, feature extraction takes place. Unlike the teams prediction model, this model implements Linear regression model to predict number of runs a player scores in the next unseen game against a specific team. Feature set includes the 6 following features for a batsman.

- Match order: This feature helps identify the order of games a player has played till that date in the dataset.
- Recent average: This is average amount of runs a batsman has scored in the last 3 performances.
- Recent strike rate: Strike rate is the rate at which batsman scoring runs for every ball. This feature calculates the strike rate for the batsman in the last 3 performances.
- Cumulative average: This is similar to recent average except that it takes all the runs scored in the every match till that particular match.
- Cumulative strike rate: This is similar to recent strike, instead it calculates the cumulative strike rate for a player till that particular match.

The averages were calculated taking the instances into account where the batsman remained not out in a particular innings and this has impact on a players average as those runs will be carried forward to the next innings. After being trained and tested over the batsman dataset, the model predicts the runs a particular batsman scores against a particular team given the input list.
D. Bowling prediction model

This is similar to Batting prediction model, the cleaning and standardizing happens similar to previous both models removing any kind of anomalies in the dataset leading to feature extraction. But in the cleaning and standardizing phase, the data lost in this model is relatively more compared to other two datasets. This model is implemented using the Linear regression algorithm to predict number of wickets a player takes in the next unseen game against a specific team. Feature set includes the 6 following features for a bowler.

- **Match order:** This feature helps identify the order of games a player has played till that date in the dataset.
- **Recent Economy:** This is average amount of runs conceded per over by a bowler after bowling his share of overs in the last 3 games.
- **Recent Maidens:** This tells us about the status of an over, whether it is a maiden or not. This feature calculates the total maidens a bowler achieved in the last three games.
- **Cumulative Economy:** This is similar to recent economy but for all the games till that instance.
- **Cumulative Maidens:** This is culmination of all the maidens till the current instance.

The economy has to be calculated considering all the cases where a partial over was bowled, and the economy rate changes with that as it had its ripple effect on the next games. There were scenarios where a few overs had 8 balls instead of 6 balls and these were normalized to 6 balls and the adjusted economy rate was calculated. Similar to the previous two models each players data was trained and tested using Linear regression algorithm and predict the wicket(s) that a player takes in the next unseen game against specific opponents.

E. Another approach

This is another Logistic regression model which was tried to predict the winner between a match prior to the current approach. The data for this model gathered from a third party website cricsheet.com. This source had data in YAML format. While, this data did not mention all the details about the match outcome but had each delivery details in a respective individual file for a match. As the goal of this project was to predict the winner along with the margin of victory, the data comprised of 1370 outcome oriented games which is less. While the model, the report discussed before consisted of 3876 data instances and had all the batting and bowling instances since the inception of the ODI format, this YAML dataset model did not have enough player information to predict the margin of victory. Also, the team dataset from this source do not have ample data to make the model learn based on the previous instances. There was relatively less data cleaning performed on this compared to the current approach but, there was a lot of structuring the data was needed as it was YAML format. The meta data had a few details but was able to be extracted whereas the match and player details of that particular match was hard to be extracted. The whole model was implemented in the Python language and all the data was parsed using a package called YAML package.

VI. ANALYSIS

The whole project was segregated into two parts - Team prediction model and Margin of victory model.

A. Team Prediction Model

The team prediction model has been implemented on training set of 87 instances and test set of 30 instances. These instances are the result of interactions between two teams in the past. The model cannot generalize the winner on a whole set of 3867 data instances.

VII. CONCLUSION

To conclude, these models predict the winner of a match between any given two teams. The margin of victory predicts the scores that a batsman makes in the next unseen game and the bowler predicts the number of wickets that he takes in the next unseen game. These models do not have a better accuracy and can be improved on this aspect.

FUTURE WORK

The main challenge in building this model is the lack of ample data. Despite collecting the entire data, the model still cannot get trained properly on the training dataset. Also, proper feature extraction can be applied to the training model to get better accuracies while predicting the winner of a match using player data.

ACKNOWLEDGMENT

I would like to take this opportunity to thank Professors Zack Butler and Reynold Bailey for their suggestions in making this project to reach this level. I would also want to thank Nihar Vanjara, my friend who supported me during this project.
REFERENCES


