

CLASS PROJECT – IPL

Predicting the outcome of a match using Big Data

BIG DATA (UE14CS314) | 15 .11 .2016

|  |  |  |  |
| --- | --- | --- | --- |
| **SNo** | **Name** | **USN** | **Class/Section** |
| 1 | Shreyasi Chaudhary | 01FB14ECS234 | D |
| 2 | Yashaswini L | 01FB14ECS298 | E |
| 3 | Savitri Kerakalamatti | 01FB14ECS216 | D |
| 4 | Rida Abid | 01FB14ECS187 | D |

# INTRODUCTION

The class project is based on predictions of score for IPL (Indian Premier League) matches. Given any two teams and the order in which each player is playing (i.e. the batting and bowling order), we aim to predict the score and which of the teams would win the match. This is done using the data of previous year’s matches and analysing the performance of each player.

We downloaded the ball-by-ball data for about 450 matches ranging over different years from the year 2008 onwards and player profiles of different batsmen and bowlers. We then clustered the batsmen into different clusters based on specific criteria. We did the same for the bowlers’ data. We then found out the probability of a batsman belonging to a cluster (of say cluster 2) scoring a particular score or getting bowled out against a bowler belonging to a specific cluster (say, of cluster 8).

## RELATED WORK

We did some prior reading on IPL matches to get familiar with cricket rules. Next we referred to the textbook and familiarised ourselves with the concepts of hive and spark. We also had to go through documentation to understand the working of K-means clustering.

## ALGORITHM/DESIGN

So far, we have completed 3 main steps to meet the final goal of predicting the score and outcome of an IPL match. We only left the last and final step.

The following steps have been completed:

1. Loading the data:

We downloaded the data from IPL website ( [www.ipl20.com/stats](http://www.ipl20.com/stats) ) and using python formatted the structure of data and did the initial data cleaning for the ease of coding. Out of the 450 matches that we had, we divided the data into training and test data (80:20 ratio). We downloaded player profile of each player (i.e. bowler and batsman) and the ball by ball data. We loaded the data into HDFS.

2. Clustering:

We performed clustering of the batsmen and bowler player profiles. We used Apache spark for clustering. Using the MLlib library in Spark, we performed K-means clustering. The batsmen and bowlers were divided into 12 clusters each using a combination of attributes like average run rate, strike rate, etc. The K-means clustering was done by performing 20 iterations.

3. Calculating Probabilities:

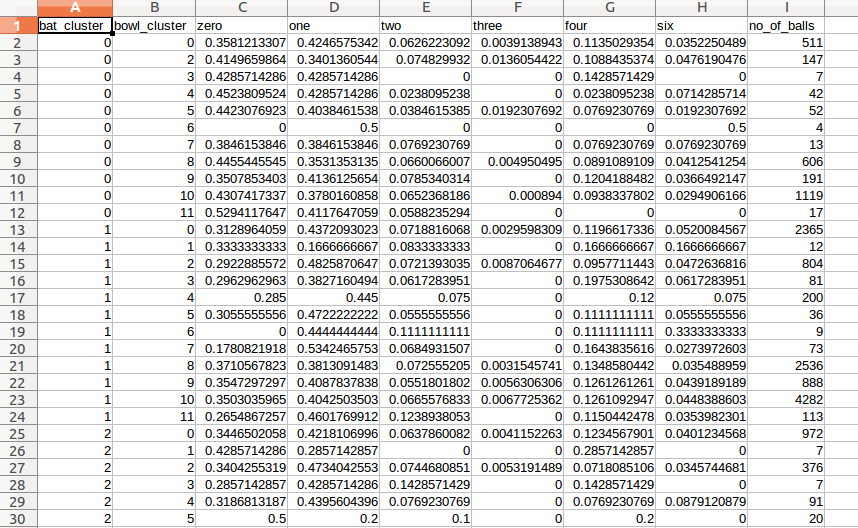
After clustering the data into a number of clusters for batsmen and bowlers, we assigned each cluster a number ranging from 0 to 11. Using Hive queries, we joined the ball by ball data with clustered data of the player profiles. As a result of the pervious step, the batsmen and bowler cluster numbers were added as columns to our ball by ball data. Next we used a query that grouped by the bowler cluster. We used the where clause to specify the batsman cluster number. Then we used aggregation operation to find the probability of scoring a 0, 1,2,3,4 and 6 and probability of getting bowled out for that batsmen and bowler clusters combination.

Next, we intend to predict score, of a batsman playing against a given bowler and finally predict runs scored by each team, and hence the winner of the match.

## EXPERIMENTAL RESULTS

The results obtained so far, after loading data is that each batsmen and each bowler have been categorised into clusters (batsman cluster and bowler cluster). Using these clusters, we calculated probabilities of a batsman scoring a 0, 1, 2,3,4,6 or getting bowled out when a given bowler is bowling.

Screenshot of results obtained after calculating the probability:



## FUTURE ENHANCEMENTS

We plan to enhance our project by adding functionality that allows it to predict the score for one day international and test matches as well. We also aim to improve the efficiency of our algorithm so that it can be used for real time analysis.

## REFERENCES

1.<http://spark.apache.org/mllib/>

2. <https://en.wikipedia.org/wiki/Apache_Spark>

3.<https://sites.google.com/site/dataclusteringalgorithms/k-means-clustering-algorithm>

4. <https://www.tutorialspoint.com/hive/>

## EVALUATION (Leave this for the faculty)

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Evaluator | Comments | Score |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## CHECKLIST

|  |  |  |
| --- | --- | --- |
| SNo | Item | Status |
|  | Source code documented |  |
| 2 | Source code uploaded to CCBD server |  |
| 3 | Recorded video of demo |  |
| 4 | Instructions for building and running the code. Your code must be usable out of the box. |  |
| 5 | Dataset used for project uploaded. Please include a description of the dataset format. This includes input file format. |  |