Simulation of IPL matches using PySpark

A Venkateswara Rao
Dept.of Computer Science,AIE
AM.EN.U4AIE19014
Amrita School of Engineering

K.Sahitya
Dept.of Computer Science,AIE
AM.EN.U4AIE19036
Amrita School of Engineering

K.R.Shuktij
Dept.of Computer Science,AIE
AM.EN.U4AIE19062
Amrita School of Engineering

Sri Vishnu Vallabh Yathavakilla
Dept.of Computer Science,AIE
AM.EN.U4AIE19065
Amrita School of Engineering

Vireen Chowdary Vesangi
Dept.of Computer Science,AIE
AM.EN.U4AIE19069
Amrita School of Engineering

***Abstract*—Big Data can be said as a revolution in the field of IT.The advantages of Data Analytics by the companies is improving day by day.The term Big refers to high volume,variety and velocity of data.Big data includes the implementation of machine learning,data mining,natural language processing techniques.With the help of the vast range of tools available for Big Data we are able to store tons and tons amount of data,pre process,analyze and visualize.Every organization wants to implement a cost-effective and efficient to analyze their data and that's where tools such as Hadoop,HPCC,Storm,Quoble,etc.These tools must be implemented in an effective way in order to get good results,which can help the organization in a very good manner.**

I. Introduction

The term big Data means for data sets that are too large data sets where traditional processing of data cannot yield efficient results. Challenges that can be faced are analysis,searching,sharing,storage,visualization,querying using Big Data Technologies such as Apache Spark,PySpark. Using these tools we are going to simulate a Indian Premier League(IPL) match using data from previous seasons of IPL.Given that two teams with players in their batting and bowling sequence, ball by ball prediction which will result in over to over prediction and finally the winner of the match

A. Problem Setting and Description

The intention of the project is to predict the match winner of a match where we can give two teams and their players in the order they play.Using probabilities of a combination of a batsman and bowler face each other we will predict what the batsman will score for a particular ball or the batsman will be eliminated.Based on these probability we will be deciding the winner of the match.

The large amount of data about the previous matches and player profiles cannot be processed normally. So we have used clustering for this.In order to find the correct number of clusters that can yield us efficient results we have plotted an elbow graph using K-means. After plotting the graph we have concluded that 10 clusters would be sufficient for our data to get better results. For the purpose of clustering we have used the classic K-means clustering.

B. Known Methods Of Implementation

There were few ideas about implementing about this project.These methods have inspired us in doing this project:

First is the project on predicting NBA games by Harish S Bhat from University of California where Harish and his team of two students from department of Applied Mathematics have developed an algorithm which can predict the winner of a NBA game using Stochastic models and continuous-time Markov chain model. The model had predicted 80 percent accurately and even in the NBA playoffs.This model does not use any Big Data tools but only pure Applied Mathematics.Given the simplicity of their model employed, their results were encouraging. There are several clear directions in which their model can be generalized and improved. First, at the moment, they used a basic frequent procedure to infer the transition rates of the continuous-time Markov chain. In ongoing work, They seek to compare this procedure against more sophisticated techniques such as variation Bayes and particle-based Monte Carlo inference. Second, the continuous time Markov chain assumes that the holding time in each state has an exponential distribution. They seek to generalize this to a distribution that more accurately models the data; this will yield a semi-Markov process

Second is a blog on Simulating IPL matches from towardsdatascience.The basic idea for the implementation was building a classic machine learning model. This included pre processing and cleaning the raw data set into a form whee we can we use to train and test the data using Logistic regression,DecisionTree,RandomForest classifier,SVM. Spilting the data set into training and testing using sckit learn and exposing the model to data which has never encountered. SVM has resulted highest accuracy with 66 percent.Where as Logistic Regression yielded 61 percent, DecisionTree yielded 62 percent and RandomForest yielded 55 percent.

II. Theriotical Background

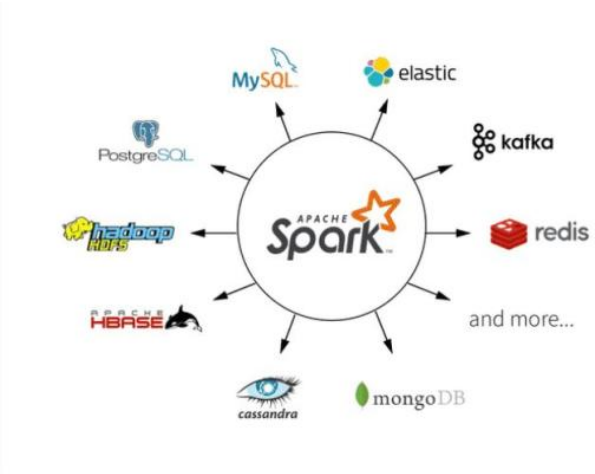
A. Apache Spark

Apache Spark is a lightning-fast unified analytics engine for big data and machine learning. It was originally developed

at UC Berkeley in 2009.Since its release, Apache Spark, the unified analytics engine, has seen rapid adoption by enterprises across a wide range of industries. Internet powerhouses such as Netflix, Yahoo, and eBay have deployed Spark at massive scale, collectively processing multiple petabytes of data on clusters of over 8,000 nodes. It has quickly become the largest open source community in big data, with over 1000 contributors from 250+ organizations.Apache Spark is 100 percent open source, hosted at the vendor-independent Apache Software Foundation.

Apache Spark Ecosystem:

- SparkSql + Data Frames
- Streaming
- Mlib(Machine Learning)
- GraphX
- Spark Core API
  - R language
  - SQL
  - Python
  - Scala
  - Java



B. K-means Clustering

K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on. It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.

It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

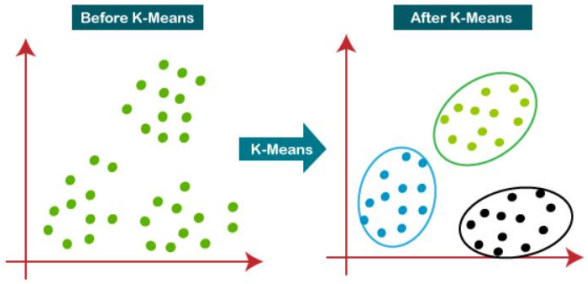
The k-means clustering algorithm mainly performs two tasks:

Determines the best value for K center points or centroids by an iterative process.

Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster. Hence each cluster has datapoints with some commonalities, and it is away from other clusters.

The way kmeans algorithm works is as follows:

- Specify number of clusters K.
- Initialize centroids by first shuffling the dataset and then randomly selecting K data points for the centroids without replacement.
- Keep iterating until there is no change to the centroids. i.e assignment of data points to clusters isn't changing.
- Compute the sum of the squared distance between data points and all centroids.
- Assign each data point to the closest cluster (centroid).
- Compute the centroids for the clusters by taking the average of the all data points that belong to each cluster. The approach kmeans follows to solve the problem is called Expectation-Maximization. The E-step is assigning the data points to the closest cluster. The M-step is computing the centroid of each cluster.



III. Approach

We have a subtle and versatile approach, the approach is divided into two steps, they are explained as follows

A. STEP - 1

After observing the dataset we understood that many batsman-bowler combinations did not occur, which might cause inaccuracies during the prediction.We have solved this challenge by using the statistics of the player which were given in the data set to perform clustering to group the batsmen and bowler.So whenever we come across any non occuring batsmen and bowler combination, outcome is predicted based on the cluster they belong to.This entailed compiling the batting and bowling averages of all players in the IPL from 2008 to 2017. For this, we used BeautifulSoup to write a web-scraping program in Python. The data was cleaned to match our specifications and saved as a csv file for ease of use. This

csv file was loaded into HDFS, Following that, we created a Python script using PySpark MLLIB for clustering batsmen and bowlers.The K-Means technique was used to cluster the data. An elbow plot was created for several values of k (the number of clusters) ranging from 1 to 40 in order to optimise the value of k . The visualisation resulted in a batsman value of 6 and a bowler value of 5.Runs, Strike Rate, Average, Number of 4s, 6s, and 50s were the factors utilised to cluster batsmen. Wickets, Economy, Average, and Strike Rate were the factors used to group bowlers.

B. STEP - 2

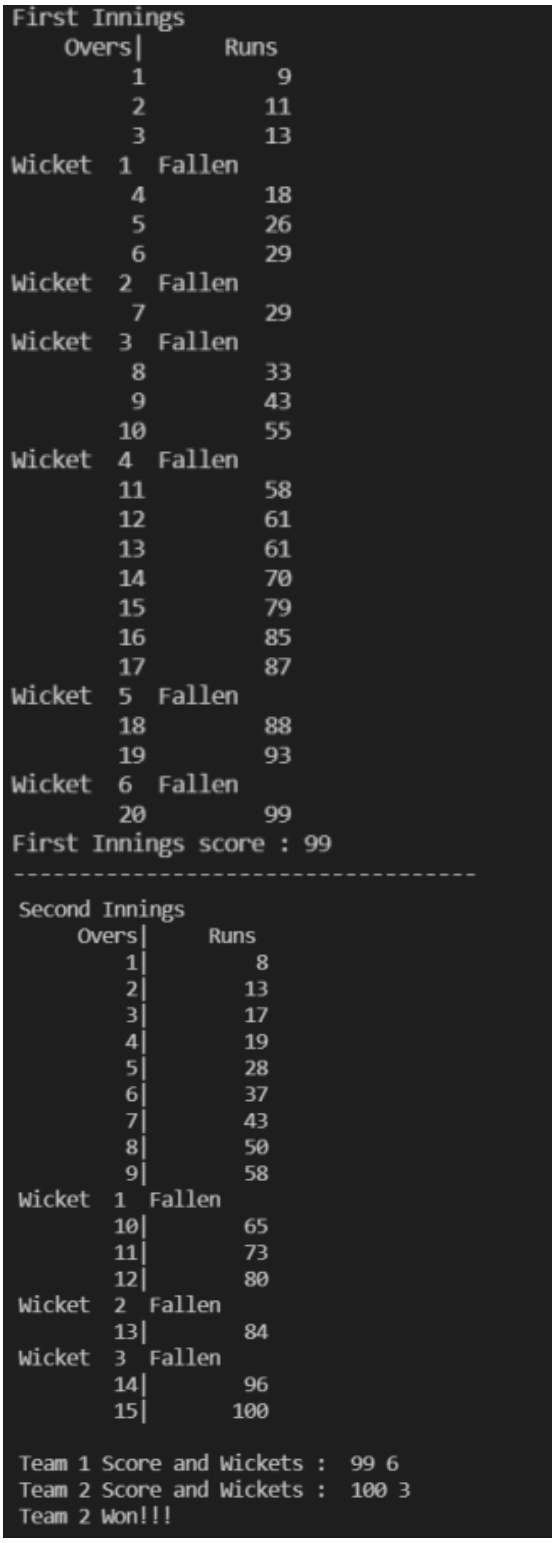
We now calculate cluster vs cluster statistics once the clusters were obtained.We will calculate using the main data set.The main Data set contains ballby ball outcome of all matches from 2008 - 2017.For performing this task we maintain a dictionary of batsmen cluster vs bowler cluster parameter totals(0s,1s,2s,3s,4s,6s,wickets).Whenever we come across a new combination, We obtained the batsman cluster number and the bowler cluster number for each new combination and added the parameters to the associated cluster combination.Finally, for each cluster combination, the probability of scoring 0s, 1s, 2s, 3s, 4s, 6s, and wickets were calculated by dividing by the total number of balls.For each batsman-bowler combination encountered in the main dataset, a summary statistics file is generated, which gives the probability of scoring 0s, 1s, 2s, 3s, 4s, 6s, and wickets when that particular combination faces each other. We created a python script that simulates an entire IPL match using ball-by-ball prediction.We looked for the batsman-bowler combination in the summary statistics file to predict the outcome of a ball. We projected the clusters for the combination and derived the probability for the given cluster combination if the combination is not found or if the batsman has faced less than 15 balls from that bowler. The probabilities acquired are used to create a list of cumulative probabilities.The outcome is determined by the range of cumulative probabilities in which it falls. 0,1,2,3,4,6 or a wicket could be the result. This is done for all balls until the 20 overs are up or no wickets remain. A new way of a wicket falling has been added. Every ball's outcome is saved in a CSV file. We ran ten simulations and averaged the results.

IV. Data Set

The main data set used int his project is taken from cricsheet website, a web page you can find ball-by-ball statistics of each and every cricket match played in world including both men and women matches.We do not every match details but only details about IPL matches with ball by ball statistics.The second dataset that we have used is taken from espncricinfo. This dataset contains the statistics of each player in IPL such as matches,runs,strike rate,wickets,economy.There are two sets of data in the this dataset. First part consists of batsman statistics of each and every player played in IPL according to their teams.Second part consists of bowling statistics of each and every player according to their teams.

V. Experimental Results and Discussion

After simulating the match using the probabilities of combinations with batsman and bowler we got an output with score predicted over by over along with wickets fallen.At the last after comparing the runs scored by each we declare the winner.Note that the results of this prediction are based on data only till 2018. So the results obtained are results in 2018.



CONCLUSION

We finally hereby conclude that we have used the K - Means along with the help of Pyspark to solve our problem i.e., To simulate an IPL match. Our approach is divided into two steps and it based on the range of cumulative probabilities and is able to simulate the match without any user intervention.This model can be further trained in the future to get more and more accurate results.We can use Decision Tree models in order Increase the accuracy in the simulation.

ACKNOWLEDGMENT

We would like to thank Dr.Venugopal Kanavillil sir for his excellence in lecturing the topics throughout the semester.We thank Dr.Venugopal Kanavillil sir for giving us an opportunity to work on this project. The assignments given were very helpful for us which made easier for us to understand and implement the concepts learned.

REFERENCES

[1] Big Data Analytics Beyond Hadoop: Real-Time Applications with Storm, Spark,and More Hadoop Alternatives, Vijay Srinivasa Agneeswaran.
[2] http://spark.apache.org/docs/latest/mllib-clustering.htmlk-means
[3] Learning Stochastic Models for Basketball Substitutions from Play-by-Play Data. Harish S. Bhat, Li-Hsuan Huang, and Sebastian Rodriguez
[4] https://towardsdatascience.com/predicting-ipl-match-winner-fc9e89f583ce