IPL Match Analysis and Predict Match Results

Gowri Srinivasa
Computer Science and Engineering
PES University
Bangalore, India
gsrinivasa@pes.edu

Ashwin Ashok, Debanik Mishra, Gajendra K.S
Computer Science and Engineering
PES University
Bangalore, India
ashwinashok1998@gmail.com, mishra.debanik@gmail.com, gajendra2ks@gmail.com

***Abstract*—Cricket is the second most watched sport in the world after soccer, and enjoys a multi-million dollar industry. There is remarkable interest in simulating cricket and more importantly in predicting the outcome of cricket match which is played in three formats namely test match, one day international and T20 match. The complex rules prevailing in the game, along with the various natural parameters affecting the outcome of a cricket match present significant challenges for accurate prediction. Several diverse parameters, including but not limited to cricketing skills and performances, match venues and even weather conditions can significantly affect the outcome of a game. There are number of research paper on pre-match prediction of cricket match. Many papers on building a prediction model that takes in historical match data as well as the instantaneous state of a match, and predict match results. It is known in the cricket match with shorter version match, result keep on changing every ball. So, it is important to predict the outcome of the match on every ball. In this paper, I have developed a model that predicts match result on every ball played. Using Duckworth- Lewis formula match outcome will be predicted for live match. For every ball bowled a probability is calculated and probability figure is plotted. For betting industry this model and the probability figure will be very useful for bettor in deciding which team to on and how much to bet.**
From casual to ardent cricket fans, the Indian Premier League (IPL) has captured the world's attention in the last decade. It's the shortest format in the game and it's unpredictable nature is what makes it so appealing to the fans.

***Index Terms*—K-means,**

I. Introduction

Cricket was one of the first sports to use statistics as a tool for illustration and comparison. Although compared to other sports, there has not been much statistical modeling work done for cricket.

Cricket is the sport that is immensely popular in India. The recent format of Twenty20 cricket which was first introduced in 2003 by the England and Wales Cricket Board (ECB), has gained huge recognition in India as well, like several other major cricket playing countries. With India winning the inaugural Twenty20 world cup in 2007 in South Africa, a massive interest in this format of cricket was generated in India.

Soon Subhash Chandra, the promoter of Essel group, started his own private Twenty20 cricket league for Zee TV in 2007 (Malcolm, Gemmell and Mehta, 2009). The name of that tournament was Indian Cricket League (ICL). The league was a six-team competition in its first year (2007) which had expanded to eight in its second year (2008). The tournament drew high profile names to play in its fixtures (Kitchin, 2008). However, the game of cricket got a new dimension in April 2008, when BCCI initiated the Indian Premier League (IPL). It is a Twenty20 cricket tournament being played among eight domestic teams, named after eight Indian cities/states, and owned by franchises (Mitra, 2010). The franchises formed their teams by competitive bidding from a collection of Indian and international players and the best of Indian upcoming talent (Saikia and Bhattacharjee, 2011). Each player has a base price fixed by the IPL authorities. However, there is no upper limit for the bid price. The valuation of players obtained through auction and the availability of players' performances have allowed researchers to infer on different aspects of this format of the game.

II. Related Work

Shubhra Singh and Parmeet Kaur [9] present a data visualization and prediction tool using HBase to keep the data related to IPL (Indian Premier League) cricket matches and players. This data is then visualizes the past performance of players' performance. The data also predicts the outcome of a match through various machine learning approaches.

The tool is used to evaluate the performance of players. This tool provides a visualisation of players' performances. Using IPL T-20 variables related to statistics of batsmen and bowlers, a number of apt variables have been identified that have elucidative power over auction values. Further, several predictive models are also built for predicting the result of a match, based on each player's past performance as well as some match related data.The developed models can help decision makers during the IPL matches to evaluate the strength of a team against another. The tool employs HBase, a distributed, open source and non-relational database for storing the data. HBase is increasingly being used for hosting of tables with billions of rows and millions of columns. It allows automatic and configurable sharding of tables for scalability of applications, The contributions of the presented work are as follows:

- To provide the statistical analysis of players based on different characteristics.
- To predict the performance of a team depending on individual player statistics.
- To successfully predict the outcome of IPL matches.

Preeti Satao et al [8] built a prediction system that takes in historical match data, player performance as well as the scores predicted by spectator, and predicts future match events culminating in a victory or loss. Our system predicts match outcome by analyzing pre-stored match data using simple but effective K-means clustering algorithm. They describes their system and algorithms and finally present quantitative results, demonstrating the performance of our algorithms in predicting the number of runs scored, one of the most important determinants of match outcome.

By using unsupervised learning algorithms, our approach learns a number of features from T-20 cricket dataset which consists of complete records of all games played since the beginning of IPL in the year 2009. For every match that is being played our system predicts the player's score by checking scores from database and also taking into account the scores predicted by fans watching the match. The system outputs a range of probable score that the player will make on that particular round.

Kalpdrum Passi and Niravkumar [4] attempts to predict the performance of players as how many runs will each batsman score and how many wickets will each bowler take for both the teams. Both the problems are targeted as classification problems where number of runs and number of wickets are classified in different ranges. They used naïve bayes, random forest, multiclass SVM and decision tree classifiers to generate the prediction models for both the problems. Random Forest classifier was found to be the most accurate for both the problems.

They have predicted the players' performance in One Day International (ODI) matches by analyzing their characteristics and stats using supervised machine learning techniques. For this, they predicted batsmen's and bowlers' performance separately as how many runs will a batsman score and how many wickets will a bowler take in a particular match.

Fahad Munir et al [3] used decision tree algorithm to design our forecasting system by depending on the previous data of matches played between the teams. This system will help the teams to take major decision when the match is in progress such as when to send which batsman or which bowler to bowl in the middle overs. It significantly expands the exposure of research in sports analytics as it was previously bound between some other selected sports.

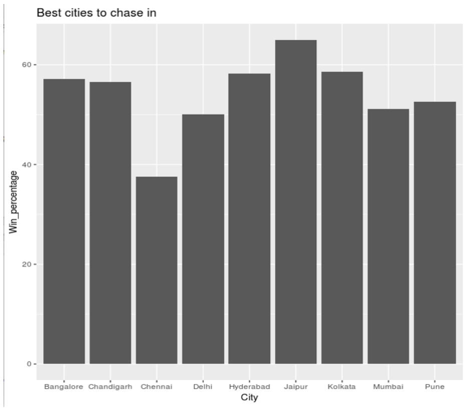


Fig. 1. Best cities to bat second

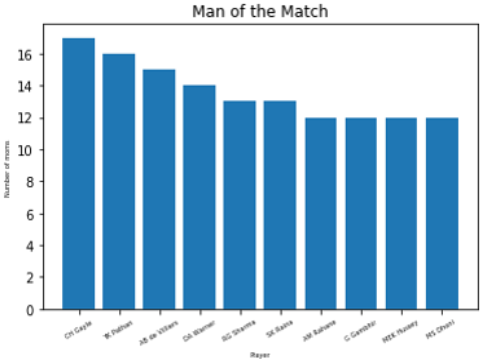


Fig. 2. Player won the most number of Man of the match

The aim is to prepare a model which will predict the result of a T20 cricket game while the match is in progress. Our main objective is to combine pre-game data and in-game data in order to design a good predictive model. Understanding the different attributes is also needed in order to get more accuracy in result.

Hemanta Saikia, Dibyojyoti Bhattacharjee and Hoffie Lemmer [7] analyze and predict the performance of bowlers in IPL, using artificial neural network. Based on the performance of bowlers in the first three seasons of IPL, the paper tries to predict the performances of those bowlers who entered in the league in its fourth season as their maiden IPL venture. The performances of these bowlers in IPL-IV are predicted, and the external validity of the model is tested using their actual performance in IPL-IV. This prediction can help the franchises to decide which bowler they should target for their team.

III. IPL Match Analysis

Few visualizations made as a part of exploratory data analysis have been attached here. Very good insights have been made from the datasets of both the sources which was well explained in the video presentation.

IV. Methodology

The following steps was followed to predict IPL match results.

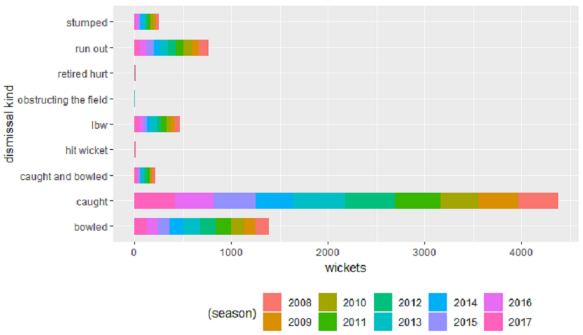


Fig. 3. Breakdown of Dismissal Type

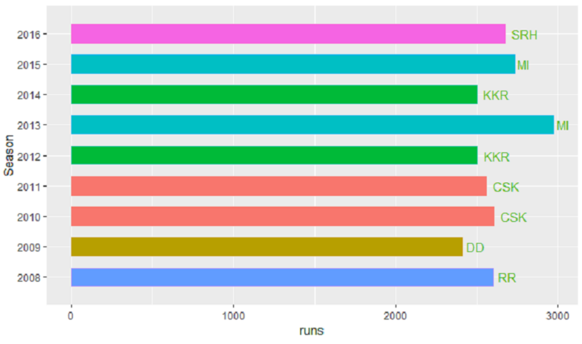


Fig. 4. Total Runs by Tournament winners for every season

A. Design of the dataset

The two main sources of our data were Kaggle for the datasets of deliveries and matches from the year 2008 to year 2017 and ESPN's cricinfo where data was scraped for players statistics. The statistics (runs scored, average, etc) of the players were taken team wise which was then merged to get the statistics for all batsmen and bowlers across all years.

B. Simulation

For each innings, the batting orders and bowling orders of both team1 and team2 is considered. For each ball, the outcome of the ball between striker and bowler is predicted. The score of the over is added to the total team score and the number of overs is incremented by one when the over finishes. Next, checked whether bowling team has bowled all the twenty overs or opponent team got all out within few limited overs. If the same condition evaluates to true, same approach is followed for innings 2. After which, the total score of both the teams and determine the match results.

C. Maximum Likelihood Estimator

Maximum Likelihood Estimator is a method of estimating the parameters of a statistical model, given observations. MLE attempts to find the parameter values that maximize the likelihood function, given the observations. The resulting estimate is a maximum likelihood estimate. MLE is a special case of maximum a posteriori estimation (MAP) that assumes a uniform prior distribution of the parameters. In frequentist inference, MLE is one of several methods to get estimates of parameters without using prior distributions. Priors are avoided by not making probability statements about the parameters, but only about their estimates, whose properties are fully defined by the observations and the statistical model.

For predicting ball by ball outcome to determine the result of the match, the relation between the striker and bowler should be calculated in terms of probability. For example, how likely is it that batsman A scores a four or a six against bowler B which was calculating using Maximum Likelihood Estimator. Suppose Batsman A has faced 20 balls from bowler B, of which he has hit 10 singles. Next time when he faces the same bowler, according MLE the proability of him getting a single is 0.5. A shortcoming with this method, when iterating through all the balls in a match and finding a game with Batsman A and Bowler B and their combinations have not been seen, there is no probability for this combination before. To overcome this drawback of grouping the batsmen and bowlers, k-means clustering technique have been used.

$$P(event) = \frac{numberoftimeseventoccurred}{totalnumberofevents}$$

D. Proposed Model

The proposed model predicts the runs scored in an unbiased manner. For example, on selecting the maximum probability of hitting runs per ball, then everytime the result will be same(which is biased), generated a random number from 0 to 1. In our model, based on random number generated, the interval to which the runs scored is determined from the Cummulative Probability Table.

TABLE I
RUNS SCORED WITH ITS PROBABILITIES

Runs	Prob(Run)	CummulativeProb(Runs)
0	0.4	0.4
1	0.2	0.6
2	0.1	0.7
3	0.0	0.7
4	0.2	0.9
6	0.1	1.0

For predicting wicket, the probabilty of batsman not out for a particular bowler is used. Probabilty of a multiple event depends on product of the probabilities of individual events when events are independent of each other. The probability of batsman getting out changes ball by ball as he faces different bowlers and number of balls faced increases. According to probability theory, two events are indepedent if the occurrence of one does not affect the probability of occurrence of the other. As the batsman plays against each bowler, the runs scored ball to ball are independent events. Eventually the probability of him being not out decreases. For dealing with the probability of not getting out with independent events. Next challlenge is deciding the threshold at which the batsman will get out. The resultant probability of P_{not_out} will get multiplied by $p_{not\_out}(b_i)$ between the batsman and bowler is calculated by

$$P_{not\_out}(b1, ..., bn) = p_{not\_out}(b1) {*} p_{not\_out}(b2) {*} .. {*} p_{not\_out}(bn)$$

P_{not_out} was used to consider whether the batsman out or not. There will be a threshold for which the batsman gets out.

Normal Distribution is a probability distribution that is symmetric about the mean, showing that data near are more frequent in occurrence than data far from the mean. The normal distribution is the most common type of distribution in most of the types of statistical analyses. The standard normal distribution has two parameters, the mean and the standard deviation.

Hypothesis Testing is a statistical method that is used in making statistical decisions using experimental data. It is an assumption that is made about the population parameter.

The following are the three main attributes of any Hypothesis testing:

- Null hypothesis: Null hypothesis is a statistical hypothesis that assumes that the observation is due to a chance factor. Null hypothesis shows that there is no difference between the two population means.
- Alternative hypothesis: Contrary to the null hypothesis, the alternative hypothesis shows that observations are the result of a real effect.
- Level of significance: Refers to the degree of significance in which whether to accept or reject the null-hypothesis. 100% accuracy is not possible for accepting or rejecting a hypothesis, therefore select a level of significance that is usually 5%.

Assuming the probability of the batsman not getting out for the balls he faced in that innings is normally distributed.

Null hypothesis:- Batsman will not get out.

Alternative Hypothesis:- Batsman gets out.

So when the probability gets lessser than 0.05, NULL hypothesis is rejected. Hence the batsman gets out after that. So in this way, the outcome for the balls bowled(runs scored or wicket taken)is calculated and accumulated to get the total team score for both the innings.

Assuming the distribution of P_{not_out} over all matches is normally distributed. To reject Null Hypothesis P should be 0.05 correspondingly 0.05 is the P_{not_out}. So threshold for getting out is 0.05

1) k-means clustering: k-means clustering is a method of vector quantization popular for cluster analysis in data mining. It aims to partition n observations into k clusters in which each observation belongs to a cluster with the nearest mean, serving as the measure of the cluster. This algorithm has a close relationship to the k-nearest neighbor classifier, a competitor when it comes to classification in machine learning techniques.

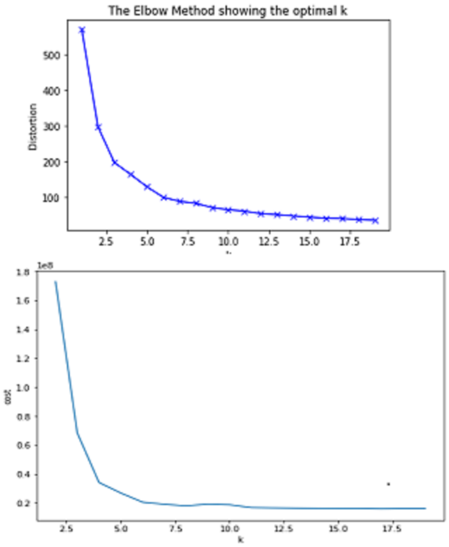


Fig. 5. Calculating k using elbow mathod for batsmen

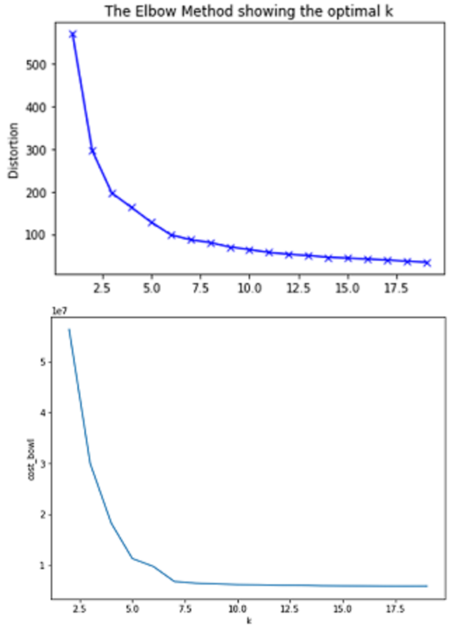


Fig. 6. Calculating k using elbow method for bowlers

Selecting features for k-means plays an important role. The elbow method consists of plotting in the graph, the WCSS(x) value. It is the sum for all data points of the squared distance between one point x_i of the cluster j and the centroid of this cluster j.

$$WCSS(k) = \sum_{j=1}^{k} \sum_{x_i \in clusterj} ||x_i - x_j'||^2$$

where x_j' is the sample mean in cluster j

For batsman clustering : K = 10

For bowler clustering : K = 7

Most of the other works have used batting average and strike rate as the features for batsmen clustering. One probable drawback of these approaches is, it might not judge a batsman properly. For example if a batsman who is actually a bowler has scored 6 runs from 5 balls,his average and strike rate will make him fall into a category of top batsmen. Similarly it goes for bowlers. The features selected should reflect the experience (number of balls faced and number of number of runs scored for the batsman) number of balls bowled and number of wickets taken for bowler) of the players.

V. Other Approaches: Classification Techniques

Apart from using ball by ball prediction to obtain the match result, few classification techniques were used to predict the winner of the match by selecting the most important attributes from the matches dataset. The attributes team1, team2, toss_winner, toss_decision and venue were considered as features to predict the attribute winner.

The following techniques were tried on this dataset:

1) Naive BayesClassifier: Naive Bayes classifiers are a family of simple probabilitistic classifiers based on applying Bayes' theorem with strong independence assumptions between the features. With appropriate pre-processing, it is competitive in this domain with more advanced methods including support vector machines. Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. For some types of probability models, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without accepting Bayesian probability or using any Bayesian methods.

2) k-Nearest Neighbors algorithm: k-Nearest Neighbor algorithm is a non-parametric method used for classification and regression. The input consists of the k closest training samples in the feature space. The output is a class membership. An object is classified by a majority vote of its neighbors, with the object assigned to the most common among its k nearest neighbors. It is a type of instance-based learning where the function is only approximated locally and all computations are deferred until classification. A weight is assigned to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the distinct onces. The neighbors are taken from a set of objects for which the class is known. This can be thought of as the training set for the algorithm, though no explicit training step is required.

3) Decision tree learning: Decision tree learning uses a decision tree to go through the observations about an item to get to conclusions about the item's target value. It is a predicted modelling approach used in statistics, data mining and machine learning. Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels.

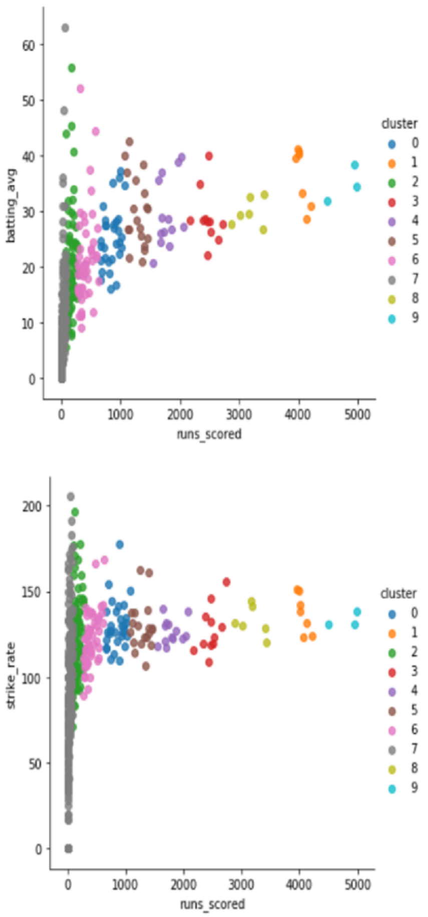


Fig. 7. Batsmen clustered into different groups

4) Support Vector Machines: In machine learning, support vector machines also called support vector networks are supervised learning models with associated learning algorithm that analyze data used for classification and regression analysis.

Multiclass SVM to assign labels to instances by using support vector machines, where the labels are drawn from a finite set of several elements. The approach for doing so, is to reduce the single multiclass problem into multiple binary classification problems.

VI. Results and Conclusions

On simulating the match ball by ball and applying our model for 10 matches and got an accuracy of 75%. On repeatedly running the simulation for 20 matches, the model predicted the results with 90% accuracy.

On trying the classification models, Naive Bayes classifier and Decision tree gave us output, but k-NN algorithm and support vector machines could not predict the classes since all the features were factors. This dataset well suited to the design of Decision tree learning as the tree involves deciding on which features to choose and what conditions to use

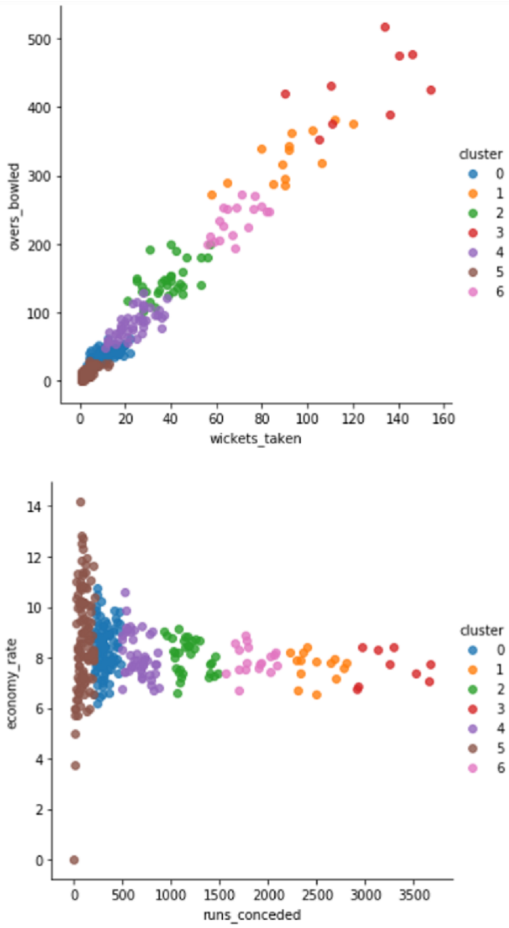
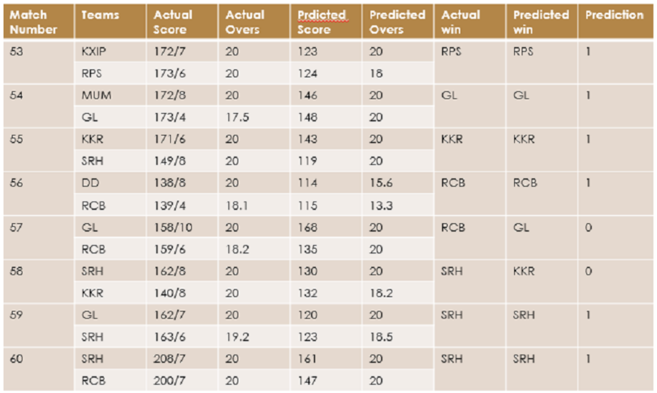


Fig. 8. Bowlers clustered into different groups



Match Number	Teams	Actual Score	Actual Overs	Prdicted Score	Predicted Overs	Actual win	Predicted win	Prediction
53	KXIP	172/7	20	123	20	RPS	RPS	1
	RPS	173/6	20	124	18			
54	MUM	172/8	20	146	20	GL	GL	1
	GL	173/4	17.5	148	20			
55	KKR	171/6	20	143	20	KKR	KKR	1
	SRH	149/8	20	119	20			
56	DD	138/8	20	114	15.6	RCB	RCB	1
	RCB	139/4	18.1	115	13.3			
57	GL	158/10	20	168	20	RCB	GL	0
	RCB	159/6	18.2	135	20			
58	SRH	162/8	20	130	20	SRH	KKR	0
	KKR	140/8	20	132	18.2			
59	GL	162/7	20	120	20	SRH	SRH	1
	SRH	163/6	19.2	123	18.5			
60	SRH	208/7	20	161	20	SRH	SRH	1
	RCB	200/7	20	147	20			

Fig. 9. Ball-by-ball simulation for 10 matches

for splitting. This algorithm is recursive in nature as the groups formed can be sub-divided using same strategy. which classifies this algorithm into the class of greedy algorithm, as we have an excessive desire of lowering the cost. This makes the root node as best predictor or a classifier.

Through this work, it is shown that the scores are different every time the simulation is made because of the random number generated. The scores are little less compared to the actual team score by 20 to 30 runs. This is one method for predicting the result of the match whose average accuracy was around 75%. Among the classification techniques, Decision tree learning algorithm gave fair results. The authors of this paper would further like to apply different models to predict the winner of the match and extend the work in predicting the winner of the IPL season.

VII. Individual Contributions

Ashwin Ashok - Literature Survey, Visualizations on the Kaggle datasets, Classification techniques.

Gajendra K S - Maximum Likelihood Estimator, k-means clustering

Debanik Mishra - Visualizations on the ESPN cricinfo datasets.

References

[1] M. G. Jhawar and V. Pudi, "Predicting the outcome of odi cricket matches: A team composition based approach," in *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD 2016 2016)*, 2016.

[2] G. Kumar, "Machine learning for soccer analytics," *University of Leuven*, 2013.

[3] F. Munir, M. K. Hasan, S. Ahmed, S. Md Quraish *et al.*, "Predicting a t20 cricket match result while the match is in progress," Ph.D. dissertation, BRAC University, 2015.

[4] K. Passi and N. Pandey, "Increased prediction accuracy in the game of cricket using machine learning," *arXiv preprint arXiv:1804.04226*, 2018.

[5] C. D. Prakash, C. Patvardhan, and C. V. Lakshmi, "Data analytics based deep mayo predictor for ipl-9," *International Journal of Computer Applications*, vol. 152, no. 6, pp. 6–10, 2016.

[6] C. D. Prakash, C. Patvardhan, and S. Singh, "A new machine learning based deep performance index for ranking ipl t20 cricketers," *International Journal of Computer Applications (0975–8887) Volume*, 2016.

[7] H. Saikia, D. Bhattacharjee, and H. H. Lemmer, "Predicting the performance of bowlers in ipl: an application of artificial neural network," *International Journal of Performance Analysis in Sport*, vol. 12, no. 1, pp. 75–89, 2012.

[8] P. Satao, A. Tripathi, J. Vankar, B. Vaje, and V. Varekar, "Cricket score prediction system (csps) using clustering algorithm."

[9] S. Singh and P. Kaur, "Ipl visualization and prediction using hbase," *Procedia computer science*, vol. 122, pp. 910–915, 2017.