Predicting the Winning Team of a Cricket League using Machine Learning and Sports Analytics

Abhinay Krishna Vellala
Harshavardhan Subramanian
Vinod Kumar Dasari
Dhyey Patel

A project presented for the course
Sports Analytics - 753A01

Linköping university
SE–581 83 Linköping

1 Introduction

1.1 Background

In the recent years, there has been a tremendous growth and demand for data analysis in Sports domain. This analysis is used to measure performances of the players, help the coaches in taking crucial decisions, retrospect on the strategies and ultimately provide competitive advantage either with respect to a specific team or a player to perform better. This process of analysing certain key attributes either with respect to a team or player is widely referred as Sports Analytics. The evolution of machine learning and deep learning methodologies has found its significant application in sports industry as well. These methodologies has proven to help in predicting future outcomes based on the available data such that better prediction provides insights on formulating better strategies and decisions. Cricket is a widely played sport and is viewed by majority of the world's population. Cricket is the second most watched sport after football. The governing body of international cricket is the International Cricket Council (ICC) under which there are 104 member nations, 12 full members play Test matches and there are 92 associate members. Cricket is a multi-billion dollar sports industry and according to ICC, there are more than 1 billion fans worldwide and the numbers are predicted to increase. Due to increase in popularity at international and domestic level, lot of importance is given in analysing the game and take intuitive decisions by predicting the future outcomes.

Cricket is a bat and ball game played between two teams. Each team consists of 11 players on the ground. At the centre of the ground, a 22 yard pitch is placed with wickets at the both the ends. Before the match starts, bowling team and batting team is decided based on coin tossing. The bowling team is responsible for setting the field players across the ground. A bowler gets to bowl 6 balls in an over to the batsman. The batsman tries to score runs by striking the ball bowled at the wicket. The fielders try to stop the ball without crossing the boundary line. There are certain rules when the batsman are declared out. Also there are different formats in the game of cricket and these formats mainly based on number of overs each team gets to bowl.

At present, there are 3 most followed cricket formats that are been followed internationally, namely, Test cricket, One day International (ODI) and Twenty20 (T20). T20 format of cricket has gained lot of attention at domestic level as well since the format is very short and exciting. Indian Premiere League (IPL) is one of the most liked and top league in T20 format. The league is conducted each year and 8 franchises participates every year for the trophy. Each team can have 4 overseas player (Selected based on bidding of the players by each franchise) and 7 local players. IPL is a 20 over format cricket and each team gets 20 overs and 10 wickets. The batting team scores the runs and set the target for the bowling team. If the bowling team chases the runs scored by the batting team within 20 overs and not losing 10 wickets, then they are declared as the winners of the match. There are other leagues like Big Bash, PBL, CPL, BPL etc., has also gained lot of attention which are franchise-based cricket. With enormous popularity and success of these leagues, the franchise based cricket has become a billion dollar sports industry. Since lot of money has been invested by the franchises on a particular team, every team wants to win and access the performances of the team as well as individual players. All of the decisions with respect to forming an ideal team of ideal players is decided by the coaches and team management. The decisions are supported by statistical analysis made on the data available with respect to each and every player. Hence, there is a demand for methods varying from basic data analytics to ML/DL frameworks that can provide insights to the management to take better decision and in turn help the team win the trophy.

Machine learning models, also known as data driven modelling approaches, is mainly used to predict the future outcome by constructing a model based on extracting features from the existing data. The features used are the independent variables and using this variables a statistical relationship is established that results in a response. The response refers to the target and it is dependent on the independent variables. The independent variables, that is the features, could be any attribute that gives insights on target to be predicted. For example if the players performance is the target response, then number of runs scored, strike rate of the batsman, number of fours and sixes scored, number of centuries, number of maiden overs, number of wickets taken etc, could be the features that helps determining the best performing player. The model builds a symbiotic relationship between independent and dependent variables such that the same can be trained on certain percentage of dataset and later predict for new data. The model also exhibits the correlation between the features. The data here refers to the statistics of number of matches played by each team, number of runs scored by each batsmen and number of wickets taken by each bowler.

1.2 Objectives

In this project, we investigate the Indian Premiere League (IPL) dataset collected from the inception of the league, that is, from 2008 to 2017. The objective is predict the winner of IPL 2018 using the ball by ball match data for IPL seasons 2008 to 2017. This would be achieved using analytics for feature engineering to process the data and machine learning algorithms to cluster the players. Then the clustered players are assigned points and the team with the highest player-count weighted points tally is predicted as the winner. This implementation of the project will revolve around addressing two basic research questions.

The research question we try to answer in this project are:

- Can the players be clustered based on certain manually defined attributes using statistical/ML methods?
- Based on the manually defined scores for players in each cluster, can we predict if a particular franchise team wins the trophy in upcoming season?

1.3 Related works

With gain in popularity for cricket around the globe, the money being invested in cricket industry is also witnessing a potential growth. Lot of research and analysis is being carried out to make efficient predictions even with imbalanced, raw and small datasets. (Vistro et. al, 2019) in [10] applied machine learning techniques like Random forest, SVM, Naïve Bayes, Logistic regression and decision trees to predict winner of IPL league. Their resulting model was used to evaluate the team's strength and analysis of cricket. Decision tree classifier outperformed other methods resulting in 94.87% accuracy in their work. They also claim that their work is majorly used by gambling and media companies. Similarly, (Passi et. al, 2018) in [7] attempted to predict players performance by estimating the number of runs a batsman will score or number of wickets a bowler will take. They setup a classification problem and used Naïve bayes, Multi class SVM and Decision tree classifiers to generate prediction models. Random forest classifier outperformed other methods in predicting with good accuracy. (Sankaranarayanan et. al, 2014) in [9] used combination of linear regression and nearest neighbour clustering algorithms to build a prediction system which is trained on history data and predict future matches resulting in win or loss. They claim in their paper that, they were successful in predicting runs for future segments and the accuracy of winner prediction is the highest reported in ODI cricket literature. (Nimmagadda et. al, 2018) in [6] proposes a model that predicts the score in each innings considering factors like toss, pitch, number of wickets fallen, venue of match as features. They use Multi variable linear regression along with Logistic regression to predict the score and use Random forest to predict the winner of the match.

All of the above-mentioned related works applies different combinations of methods to predict the winning percentage or evaluate the players performance.

2 Data

In this work, we apply our methodology on Indian Premiere League dataset obtained from a Kaggle competition. The original source of the data is obtained from Cricsheet website [3]. The raw data consists of ball by ball data of each match played between 2008 – 2017 seasons. The raw datasets consisted of two csv files involving details related to the match such as location, contesting teams, umpires, results, etc. and ball-by-ball data of all the IPL matches including data of the batting team, batsman, bowler, non-striker, runs scored, etc.

The data had to be pre-processed to apply our methods and the same is discussed below.

2.1 Data pre-processing

The raw data obtained from kaggle contained two files and each file consist of following information respectively,

- match_id, inning—batting_team, bowling_team, over ball, batsman—non_striker, bowler, is_super_over, wide_runs bye_runs, legbye_runs, noball_runs, penalty_runs, batsman_runs, extra_runs, total_runs, player_dismissed, dismissal_kind, fielder.
- id—season, city, date, team1, team2, toss_winner, toss_decision, result, dl_applied, winner, win_by_runs, win_by_wickets, player_of_match, venue, umpire1, umpire2, umpire3.

As mentioned in data section, the datasets contain ball by ball information of each player of each team. Each row represents batting team name, batsman name, runs scored by batsman, extra runs, bowling team name, bowler name, if the batsman was dismissed in that ball. It was important to preprocess the data into a format where the manually defined attributes can be calculated.

A players performance can be identified by certain traits irrespective of batsmen or bowler. The basic attributes of a batsman are the number of runs scored, strike rate, and number of fours/sixes hit and for a bowler the number of wickets taken, economy, and maiden overs. However, to define standard attributes that determine the player performance for both batsman and bowlers, we calculate the following attributes that are described in [4].

Batting Attributes

- Hard Hitting Ability = (Fours + Sixes) / Balls Played by Batsman
- Finisher = Not Out innings / Total Innings played
- Fast Scoring Ability = Total Runs / Balls Played by Batsman
- Consistency = Total Runs/Number of Times Out
- Running Between Wickets = (Total Runs – (Fours + Sixes))/(Total Balls Played – Boundary Balls)

Bowling Attributes

- Economy = Runs Scored / (Number of balls bowled by bowler/6)
- Wicket Taking Ability = Number of balls bowled / Wickets Taken
- Consistency = Runs Conceded / Wickets Taken
- Crucial Wicket Taking Ability = Number of times four or five wickets taken / Number of innings played
- Short Performance Index = (Wickets Taken – Number of Times Four Wickets Taken – Number of Times Five Wickets Taken) / (Innings Played – Number of Times Four Wickets or Five Wickets Taken)

To calculate the batting attributes, the ball by ball data is grouped based on each player who has played between 2008 and 2017 seasons of IPL. The individual runs scored by each of the batsmen for every ball for all the matches is recorded initially. From the raw data file, we consider the column **batsman_runs** to determine number of total runs scored and number of fours/six hit. Number of balls played by each batsman is aggregated based on name column **batsman**. The total innings played by each batsman is again grouped based on columns **batsman** and **match_id**. The number of not out innings of each player is estimated based on the column **player_dismissed**.

After grouping and conditioning on certain columns, each of the batting attribute is calculated for each player for all the seasons between 2008 and 2017. Also, to reduce the bias with respect to players who played less matches and performed well, we induce a filter that filters out the players who has played less than 10 matches. At the end, we have 5 batting attributes for **212 batsmen** who played in IPL editions from 2008 and 2017.

Similarly, to estimate the bowling attributes of each of the bowler, we first stored number of individual runs conceded by each bowler throughout the season played between 2008 and 2017 after grouping based on the column **bowler**. We followed the similar approach to calculate the number of balls bowled by each bowler including extra balls (that is, wide balls / no balls). Based on the column **player_dismissed**, if the batsman is dismissed then the count is increased by 1 to that specific bowler. This count at the end is the number of wickets taken by each of the bowler. Based on the count determined in the previous step along with column **match_id**, the number of wickets taken by each player in each match is filtered. Based on number of times 4/5 wickets taken by a bowler in certain number of matches certain other bowling attributes are calculated.

Similar to the batting attributes, to reduce the bias with respect to bowlers who played less matches and performed well, we induce a filter that filters out the players who has played less than 10 matches. At the end, we have 5 bowling attributes for **191 bowlers** who played in IPL editions from 2008 and 2017.

3 Theory

3.1 Clustering using Gaussian Mixture Model

In the previous section, we manually defined 5 attributes for each of the bowler and batsman that helps in determining the performance of a player. The next step in the methodology is clustering the players based on the attributes. Clustering is grouping of homogeneous data points together such that each group consists of similar data points. It is an unsupervised method where the target label of data points is unaware. Each of such group is called as a cluster and there are many clustering algorithms that are been widely used in machine learning research space.

In this work, we first use Gaussian mixture model (GMM) method for clustering the players. GMM assumes that there are specific number of Gaussian distributions and each of the data points belongs to one of the distribution such that each of the distribution can be considered as a cluster. GMMs are probabilistic model based clustering technique that results in a soft clustering assignment. Since we are working on multinomial data, that is, more than one feature variable, Multivariate Gaussian mixture model (MGMM) is adapted. The probability density function of a Multivariate Gaussian distribution is denoted by below equation,

$$f(x \mid \mu, \Sigma) = \frac{1}{\sqrt{2\pi\Sigma}} e^{\left[-\frac{1}{2}(x-\mu)^t \Sigma^{-1}(x-\mu)\right]} \tag{1}$$

where x is the input vector, μ is the mean vector, Σ is the covariance matrix. Hence, for a dataset with f features will have mixture of k Gaussian distributions (k also represents number of clusters) with a mean vector and a covariance matrix of size $f * f$.

The values of mean vector μ and covariance matrix Σ are usually determined using Expectation-Maximization (EM) algorithm as described in [1]. EM algorithm uses the existing data to estimate the optimal model parameters.

The mixture model and EM algorithm estimates the likelihood based on optimal μ, Σ values for specific k number of clusters. But the optimal number of distributions, that is, number of clusters is not determined yet. We use information criterion for model selection to find the optimal number of clusters. A penalty is added to compare likelihoods for given k number of clusters to determine model's performance and complexity.

In this work, we use two information criterion to determine optimal number of clusters, namely, Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC). AIC was mainly used to approximate expected predictive log-likelihood of the model by assigning a score [8]. AIC score is given as,

$$AIC = -2\log(L) + 2k \tag{2}$$

where L is the likelihood and k is the number of model parameters in complexity term $2k$. BIC is used on the context to approximate Bayes' factor while comparing the models [8]. BIC score is given by,

$$BIC = -2\log(L) + k\log(n) \tag{3}$$

where complexity term is expressed as number of model parameters k times the log of number of observations n. The AIC and BIC score usually agrees with each other but BIC penalizes more as the complexity of the model increases unlike in AIC. Based on the AIC and BIC score for specific number of clusters, the optimal k number of clusters are selected.

3.2 K-Means clustering

K-Means is a partitioning based clustering method which constructs various partitions and evaluates the clusters by some clustering criterion. Suppose we are given a dataset $X = (x_1, x_2, x_3, ..., x_N)$ where $x_n \in R^d$. The K-means clustering problem aims to partition this data into K disjoint groups called clusters $C_1, C_2, C_3, ..., C_K$ such that the clustering criterion is minimized. The common clustering criterion used in K means algorithms is the sum of the squared Euclidean distances between the data points x_i and the centroid m_j of the cluster C_j which contains x_i. This criterion is the cluster error called distortion is computed given the cluster centroids $m_1, m_2, ..., m_K$ and clusters $C_1, C_2, C_3, ..., C_K$ as below.

$$E(m_1, m_2, ..., m_K, C_1, C_2, C_3, ..., C_K) = \sum_{i=1}^{N}\sum_{j=1}^{K} I(x_i \in C_j)||x_i - m_j|| \tag{4}$$

In Equation 4, $I(x_i \in C_j) = 1$ if x_i is in the C_j and $I(x_i \in C_j) = 0$ otherwise. The K-means method is based on the number of clusters K and the initial values for the centroids. The optimal value for the number of clusters K is selected using Elbow method. This method involves the plotting of explained variation as a function of number of clusters K and the elbow of the generated curve in the plot is selected as optimal K [2]. The centroid for a cluster is computed as follows:

$$m_j = \frac{1}{P} \sum_{i=1}^{P} x_{ij} \tag{5}$$

Where P is the number of data points and x_{ij} are data points belongs to the cluster C_j. The algorithm for the method is explained as follows:

- Determine the number of clusters K and choosing initial centroids $m_1, m_2, m_3, ..., m_K$ arbitrarily from the data points for the clusters.
- Repeat step 3 and 4 until there is no change in the centroid values or for some specified number of iterations I.
- Assign each data point x_i to the nearest centroid to create new clusters.
- Update the centroids by computing the centroids for the generated new clusters by using the equation 5.
- The final clusters are saved.

4 Method Overview

The ball-by-ball data is processed and separate batting and bowling attributes are calculated for the respective players in the data pre-processing step. These attributes along with the count of matches played is then taken as input for clustering the players (batsmen and bowlers separately) based on their performance.

GMM and K-Means are the two clustering methods that are used for the studied problem. GMM is applied first, however, this method did not conclusively result in convergence for optimal number of clusters, and the most likely reason for this is explained in detail in section **??**. K-Means clustering results in convergence at 6 distinct clusters for both batsmen and bowlers. These 6 clusters for both batsmen and bowlers are then considered for points allocation. Even though the GMM method do not conclusively converge, we use GMM to verify the performance of K-Means clustering. We verify the players allocated by selecting 6 clusters in GMM with the players from 6 clusters obtained using K-Means. It is observed that most of the players in each cluster remain the same and this acts as validation that most players are being correctly clustered according to the attributes.

The next step is to assign points to clusters based on their overall performance across the attributes. To check the performance across the clusters relative to the mean, the attributes are standardized to bring them to a similar scale. Then the boxplots for all the attributes for each cluster are compared relative to the mean. The cluster with exceptional performance over most number of attributes in comparison to mean (0 mean because of standardization) is the best group of players. The cluster with the second best performance over the attributes is second best and points allocated respectively. This determines the ranking of the clusters and the points are allotted based on the ranking. For the 6 clusters, we have selected a 5-10 points structure with the players in the worst cluster getting 5 points and the players in the best cluster getting 10 points.

There are some players that are in both the batting as well as bowling datasets. This creates a rating problem because some of the players will have either batting or bowling rating only. Also there are some new players in the 2018 season that don't have a score as they are not in the dataset before. We have solved these issues by assigning a 5 (lowest rating) to those attributes or players. Thus we calculate the overall rating of the players by taking average of both the batting and bowling ratings. By this method the overall rating of the new players will also be 5. Thus, while predicting the winner for 2018, more importance will be given to the players with the best statistics in the previous seasons.

The last step is to take the players in new teams according to the 2018 season and add up the overall points tally for each team. Since the number of players in each squad is different, this would give an advantage to teams with more players in the squad. To solve this issue, we divide the overall points tally of each team by the total number of players in their respective squads and that is the final points tally taken into consideration. Thus, the predicted winner for the 2018 season is the team with this highest overall points tally.

5 Implementation and Results

The pre-processed datasets have attributes of batting and bowling along with the number of matches each player played in all the seasons from 2008-2017. In total there are 461 batsmen and 356 bowlers who played at least one match. Players are filtered based on the number of matches they have played across all the seasons. The ones that have played at least 10 matches are considered and rest of the players are discarded. This is because attributes of the players with less than 10 matches are contradicting with the experienced players. For example, for the attribute "hard hitting ability", batsman who played 3 matches have higher score than the one who played 141 matches. This situation might induce a bias by the clustering approaches to recognize this difference and generate a valid cluster. Hence, we discarded the players whose appearances are in less than 10 matches for both batsmen and bowlers. After filtering, there were **212 batsmen and 191 bowlers**.

Clustering is performed on both the datasets with batting attributes and bowling attributes individually. A grid of k values are considered to cross validate the optimal number of clusters where, $k \in [2, 10]$. Initially, clustering is done using Gaussian Mixture Model and AIC/BIC is used to determine optimal clusters. With 212 players in batting attributes dataset, GMM failed to show optimal number of components within the data, the AIC scores are decreasing and BIC is increasing with the increase in number of k value. This is because of the less number of rows in the data as BIC is penalizing the model complexity more heavily, in addition, AIC and BIC scores are disagreeing to each other. We have verified this by taking all the 461 batsmen without the filter of 10 matches and the AIC and BIC agree each other and the optimal cluster is determined at $k = 6$ components. The plot that shows the trend of AIC/BIC scores in both cases is shown in section 5. However, the objective is to consider the players who appeared in more matches. Hence, we have selected K-Means clustering over GMM method of clustering.

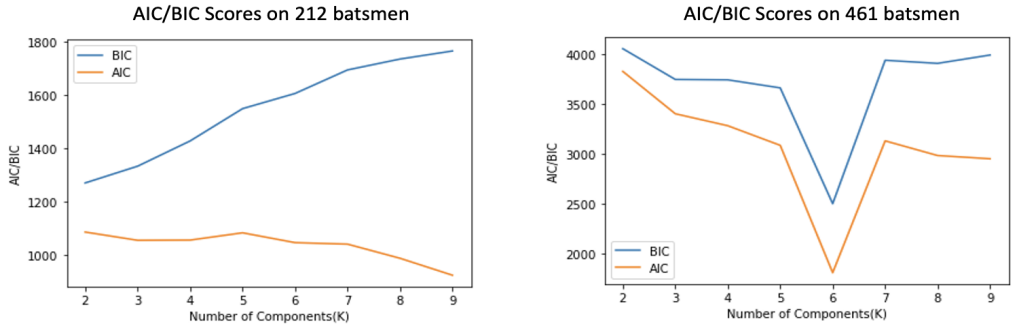

Figure 1: Trend of AIC/BIC scores on K components using Gaussian Mixture Model

5.1 K-Means Clustering - Result

The K-Means clustering is performed on the two datasets on a grid of k values where, $k \in [2, 20]$ and optimal k is determined by using elbow method in which distortion is computed for each cluster. The K-Means algorithm tries to minimize distortion, which is defined as the sum of the squared distances between each observation in the vector and its dominating centroid [5]. A within cluster sum of squared distances (WCSS) is computed for each k value to check the distortion in each cluster. Elbow method suggests that the optimal k value is the one where the distortion starts decreasing in linear fashion [2]. The distortion values are plotted for both batting and bowling data to check the elbow point as shown in subsection 5.1. In both the cases, after $\mathbf{k = 6}$, the distortion values started decreasing linearly. So, the optimal cluster is chosen to be 6.

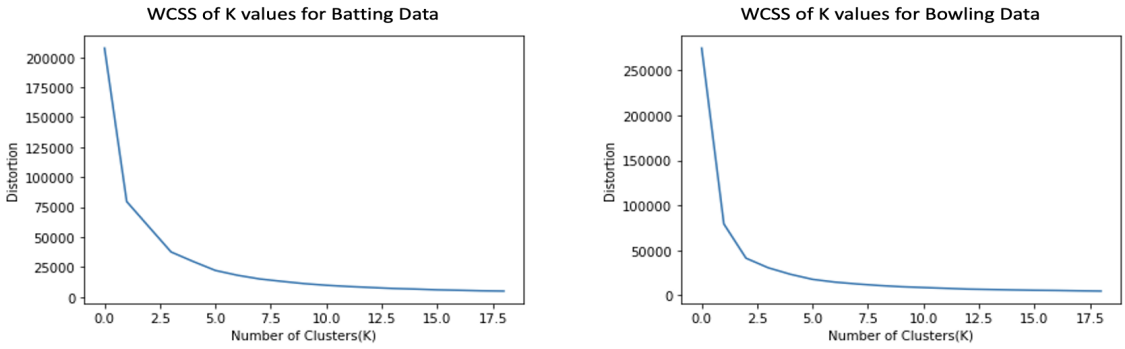

Figure 2: Within Cluster Sum of Squared distances(WCSS) for multiple K values. Plot in the left is for batting data and the right for bowling data

5.2 Cluster Analysis

The WCSS plot shown in subsection 5.1 determines the optimal number of clusters using elbow method as 6. Using the optimal k value, K-Means clustering is performed on both the filtered batting and bowling observations. The clusters do not explicitly provide any idea about the type of cluster. A detailed analysis on clusters to determine the type of the clusters was carried out and based on the type, players in the clusters are scored. The following sections discuss about the analysis on batting and bowling data.

5.3 Batting Cluster Analysis

With k = 6, the clustering is performed on all the 212 batsmen that clusters them as per their performance in the attributes derived. The attributes are standardized where mean = 0 and standard deviation is 1 so that it would be easy to compare the performance of players and derive the type of cluster. subsection 5.3 shows the performance of players based on the standardized attributes of each cluster. For cluster 3, all players in each attribute are above the mean except 'finisher'. All the players are experienced which is represented by 'matches' attribute. There are two players who are above the mean for 'finisher' attribute as well. Based on the prior knowledge we have on type of players in the cluster and results of performance plot in the cluster, we have given the Cluster 3 as best performing cluster.

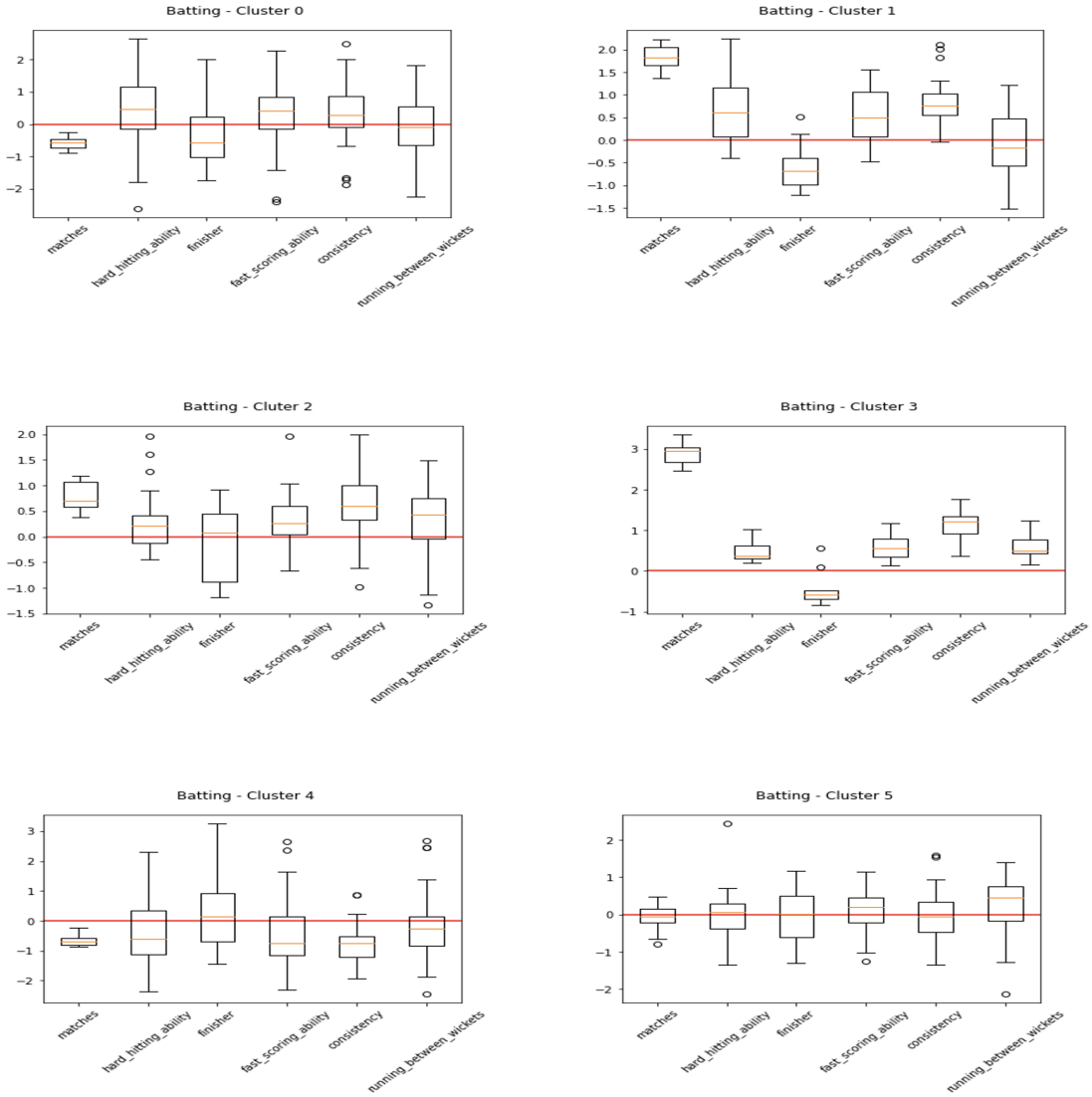

Figure 3: Analysis of attributes on cluster for Batting data

Similarly, for cluster 4, all the players are not experienced and their performance is below the mean as shown in the plot except finishing attribute. 'Finisher' attribute is computed based on number of times the

batsman remained 'Not Out'. Since most of these players come to bat at end of the innings, there are high chances that these players remain 'not out'. So, even if it is above the mean, the other attributes are not good compared to other clusters. After looking the players and results of attributes, we have chosen Cluster 4 to be worst performing batsmen cluster.

A similar approach is implemented for all the clusters and the clusters are ranked as per the performance of players based on batting attributes. Table 1 shows the rank of each cluster from best performing to least performing groups. For each player in the cluster, points are assigned. The highest points are given to the best performing clusters that is 10 and the following clusters have points as shown in Table 1. The inspiration to assign points is from the fantasy leagues in IPL moderators [1]. For convenience of this project, we have only taken the integer values for points.

Cluster	**Rank**	**Points**
3	Best (1)	10
1	Great (2)	9
2	Good (3)	8
0	Okay (4)	7
5	Average (5)	6
4	Worst (6)	5

Table 1: Top performing clusters to least performing clustering based on Batting attributes and the points assigned to each cluster

Each player is assigned to the respective cluster and the points are given as per the cluster as shown in Table 2. The table shows the assignment of points to each player based on the cluster. Similar approach is followed for all the clusters.

Player	**Cluster**	**Points**
V Kholi	3	10
CH Gayle	1	9
MS Dhoni	3	10
G Maxwell	2	8
Yuvraj Singh	1	9
B Lee	4	5

Table 2: Assignment of points to each batsman based on the cluster

5.4 Bowling Cluster Analysis

A similar approach is implemented for bowling data as shown in subsection 5.3 where the data is clusters using optimal number of clusters k = 6 as discussed in subsection 5.2. The attributes are standardized and a box plot is plotted for each cluster to determine the type of cluster based on the attributes. subsection 5.4 shows the performance of the players of each cluster. T20 leagues are mostly batting dominant leagues and it is expected that the bowlers will not be as good as batsman. Based on the results we have got, the attributes of each cluster are examined. For cluster 1, the mean of three attributes is above the global mean. The bowlers are experienced, have good economy rate, have taken at least 1 wicket in a match and there are players who have taken 4 or 5 wickets in a match. This is a good quality bowling for a short format league. Cluster 1 is considered as the best performing bowlers cluster. For cluster 5, the number of matches played are extremely low, none of the bowlers have taken a 4 or 5 wickets in a match and the economy is also not great. Based on the prior knowledge on the players and attributes, we have chosen cluster 5 as worst

[1]IPL Fantasy League

performing cluster. A similar analysis is done for all the other clusters. The players are ranked as per the clusters as done in subsection 5.3. The rank of each cluster along with the points are assigned is shown in Table 3.

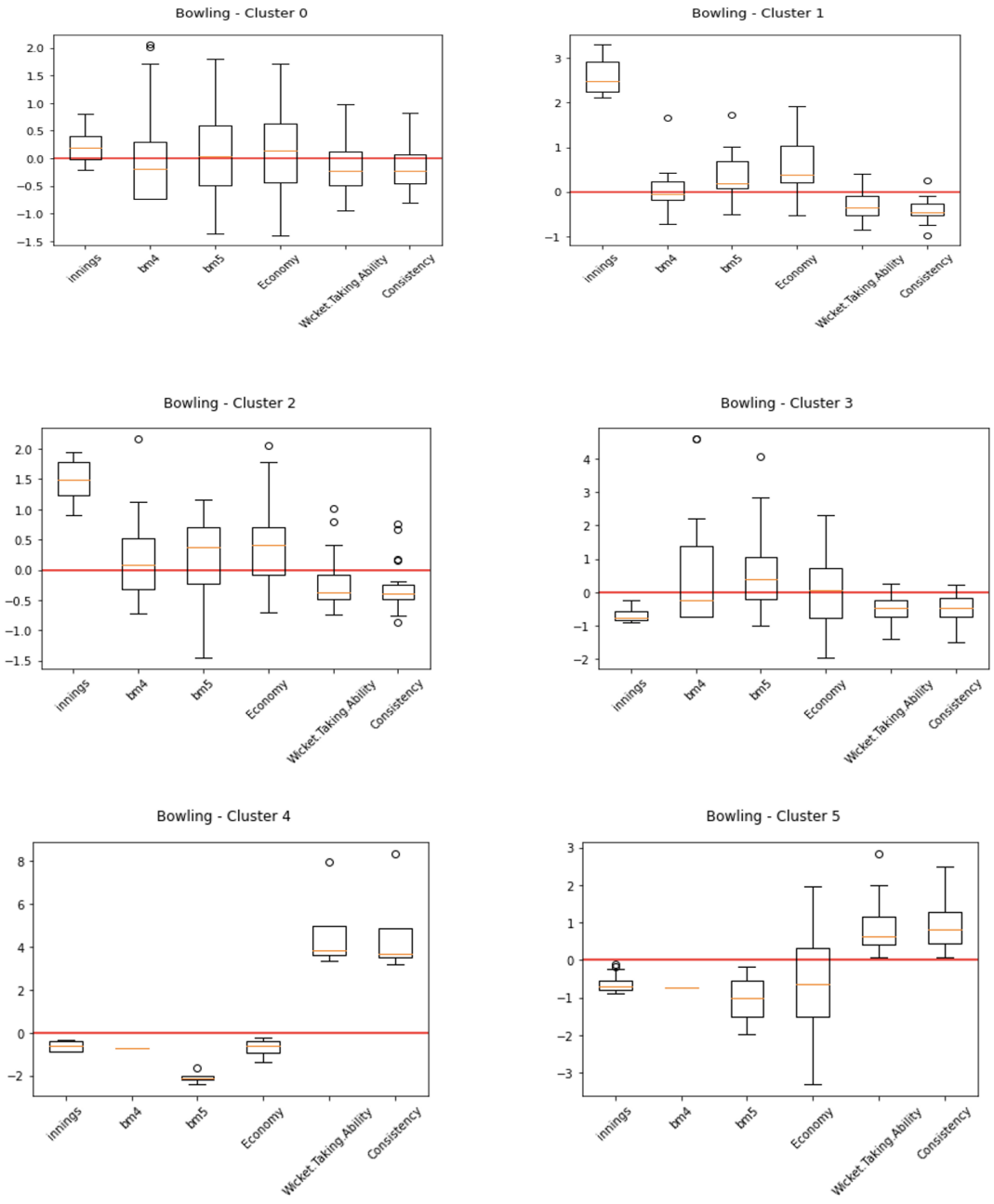

Figure 4: Analysis of attributes on cluster for Bowling data

Cluster	Rank	Points
1	Best (1)	10
2	Great (2)	9
0	Good (3)	8
3	Okay (4)	7
4	Average (5)	6
5	Worst (6)	5

Table 3: Top performing clusters to least performing clustering based on Bowling attributes and the points assigned to each cluster

As shown above, each player is assigned to the respective cluster and the points are given as per the table as shown in Table 4.

Player	Cluster	Points
KH Pandya	5	5
A Zampa	3	7
B Kumar	2	9
RA Jadeja	1	10
Yuvraj Singh	2	9
SL Malinga	1	10

Table 4: Assignment of points to each bowler based on the cluster

5.5 Winner Prediction

As discussed in methods, the team squads from 2018 season are pulled and each player is assigned both batting and bowling score. The players who don't have bowling or batting statistics from previous seasons are given 5 points which is minimum. Also, the new players who debuted in 2018 season are given 5 points in both bowling and batting categories. The scores are averaged and overall score shown in Table 5.

Team	Player	Batting Points	Bowling Points	Overall Points
Chennai Super Kings	DJ Bravo	8	10	9.0
Sunrisers Hyderabad	YK Pathan	10	9	9.5
Mumbai Indians	JJ Bumrah	5	8	6.5
Chennai Super Kings	MS Dhoni	10	5	7.5
Kings XI Punjab	Yuvraj Singh	9	9	9.0
Royal Challengers Bangalore	V Kohli	10	6	8.0

Table 5: Points assignments in 2018 Squad

The overall points of each team are summed up and factored with the number of players in the squad. The expression is shown as:

$$Team\ Overall\ Points = \frac{Sum_{team}(Overall\ Points)}{Number\ of\ players\ in\ team} \tag{6}$$

For each team the score is computed and Table 6 shows the overall team score based on the players they have in the squad.

Team	Team Overall Score
Chennai Super Kings	6.75
Kings XI Punjab	6.35
Kolkata Knight Riders	6.34
Mumbai Indians	6.27
Sunrisers Hyderabad	6.2
Royal Challengers Bangalore	6.075
Delhi Daredevils	5.93
Rajasthan Royals	5.79

Table 6: Team Overall score (2018)

The tables shows 'Chennai Super Kings' as the strongest team with highest of 6.75 points. The actual winner of IPL 2018 was the same team.

6 Conclusion

The research objective of this project was to predict the winner of IPL 2018 based on historical data from 2008-2017 seasons using conventional statistical/ML methods. We were successful in predicting the right winner of IPL 2018 season after applying clustering methods like K-Means algorithm. Even Gaussian Mixture Models could have perfromed better with more number of players by choosing right number of components using probabilistic model selection using Akaike Information Criterion, Bayesian Information Criterion. The raw data was preprocessed to calculate the 5 manually defined attributes for both batsman and bowlers each to assess the players performance. Based on the attributes, we applied K-Means algorithm to cluster the batsmen and bowlers separately. The optimal number of clusters for both batsmen and bowlers was found to be 6. Hence the players played between 2008 and 2017 were clustered into either one of the cluster depending on the player being batsman or a bowler. After clustering, the players in each of cluster were assigned manually defined scores. Based on cumulative scoring of players in each for each of the team, the team with highest cumulative score was considered as favorites of winning the 2018 season. The cumulative score for Chennai Super Kings (CSK) was found to be the highest based on the calculations (2008-2017 IPL data) carried out in our work and in reality Chennai Super Kings did win the 2018 IPL season. Therefore, confirming that our proposed methodology proves to be effective for the given historical data in predicting the winner of next season by statistically scoring each player. The methodology can be made more robust by including certain features like fielding attributes for players, toss, venue, pitch and weather conditions. The application of this methodology can be extended to ODI and Test cricket as well to predict the performance of the team.

References

[1] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[2] *Elbow Method for optimal value of k in KMeans*. `https://www.geeksforgeeks.org/elbow-method-for-optimal-value-of-k-in-kmeans/`. Accessed: 2021-06-12.

[3] *Freely-available structured ball-by-ball data for international and T20 League cricket matches*. `https://cricsheet.org/`. Accessed: 2021-06-12.

[4] *IPL and bigdata analytics*. `https://www.firstpost.com/long-reads/ipl-and-big-data-analytics-a-match-made-in-heaven-4438611.html`. Accessed: 2021-06-12.

[5] *K-means clustering Documentation - SciPy*. `https://docs.scipy.org/doc/scipy/reference/cluster.vq.html`. Accessed: 2021-06-12.

[6] Akhil Nimmagadda, Nidamanuri Venkata Kalyan, Manigandla Venkatesh, Nuthi Naga Sai Teja, and Chavali Gopi Raju. "Cricket score and winning prediction using data mining". In: *International Journal for Advance Research and Development* 3.3 (2018), pp. 299–302.

[7] Kalpdrum Passi and Niravkumar Pandey. "Increased prediction accuracy in the game of cricket using machine learning". In: *arXiv preprint arXiv:1804.04226* (2018).

[8] *Probabilistic Model Selection with AIC, BIC, and MDL*. `https://machinelearningmastery.com/probabilistic-model-selection-measures/`. Accessed: 2021-06-12.

[9] Vignesh Veppur Sankaranarayanan, Junaed Sattar, and Laks VS Lakshmanan. "Auto-play: A data mining approach to ODI cricket simulation and prediction". In: *Proceedings of the 2014 SIAM International Conference on Data Mining*. SIAM. 2014, pp. 1064–1072.

[10] Daniel Mago Vistro, Faizan Rasheed, and Leo Gertrude David. "The Cricket Winner Prediction With Application Of Machine Learning And Data Analytics". In: *International Journal of Scientific & Technology Research* 8.09 (2019).