BEST IPL TEAM OF 11 PREDICTION

Submitted for the Evaluation of Machine Learning
In
Third Year
Of
Information Technology

By

1. **DIVYANK PATIL (2020BIT017)**
2. **AMIT PANDHARE (2020BIT038)**
3. **AASHUTOSH KARALE (2020BIT054)**
4. **SHLOK INGLE (2020BIT068)**

Under the Guidance of
PROF. RACHANA POTPELWAR

ACADEMIC YEAR
(2022-23)

CERTIFICATE

This is to certify that the project/ research work entitled "**Best 11 IPL Team of 11 Prediction**" being submitted by Mr. Divyank Patil, Amit Pandhare, Aashutosh Karale and Shlok Ingle to Shri Guru Gobind Singhji Institute of Engineering & Technology, Nanded for the award of the degree Bachelor of Technology in Information Technology is a record of bonafide work carried out by him under my supervision and guidance. The matter contained in this dissertation has not been submitted to any other University or institute for the reward of any degree or diploma.

Prof. Rachana Potpelwar Project Guide	Dr. Balaji Shetty Head Department of Information Technology	Dr. M. B. Kokare Director SGGSIE&T, Nanded

ACKNOWLEDGEMENT

It is a privilege for me to have been associated with Prof. Rachna Potpelwar, our guide during this project work. I have greatly benefited from her valuable suggestions and ideas. It is with great pleasure that I express our deep sense of gratitude to her for her able guidance, constant encouragement and patience throughout the work.

I am also thankful to Dr. Manesh Kokare, Director and Dr. Balaji Shetty, Head of Information Technology Department for their constant encouragement & cooperation.

I take this opportunity to thank Divyank Patil, Amit Pandhare and Aashutosh Karale for providing company during the work.

Shlok Ingle

ABSTRACT

One of the most exciting outdoor games that reached everyone heart is cricket. There are several series held and one among that created a magnificent history in the arena of sports is Indian Premier League (IPL). It has reached its popularity with successful brand in the world of sports and usually will be conducted among 8 teams. This proposed paper is specifically concentrating on enactment and measuring the difference between the models to foretell the captivating team of an IPL match. Data is accessed by the computer programs developed using Machine learning to build models. As of now, data analysis is need for each and every fields to examine the sets of data to extract the useful information from it and to draw conclusion and as well make decisions according to the information. The algorithm first analyzes the data to create a model, specifically for understanding the patterns or trends. For creating the mining model, the model is optimized by selecting parameters and iterating. To extract actionable patterns and detailed statistics, the parameters are then fed into the dataset. This work focuses on finding the meaningful information about the IPL Teams by using the functions of R Package. R reduces the complexity of data analysis as it displays the analysis results in the form of visual representations. The dataset is loaded and a set of pre-processing is done followed by feature selection. Four machine learning algorithms Decision Tree, Naive Bayes, K-Nearest Neighbour and Random Forest are applied and the results are compared to measure the accuracy, precision, recall and sensitivity. The best of the four machine learning techniques are then applied to predict the winner and visualize the results as graphs.

TABLE OF CONTENTS

INTRODUCTION

England first introduced T20 Cricket in 2003. Because of its shorter format, it became very popular. Due to its popularity of high voltage action, T20 came to India also. The BCCI initiated a 20-20 cricket tournament Indian Premier League (IPL) in 2008. BCCI has been organizing the IPL T20 cricket tournament every year. The use of analytical methods in various aspects of cricket including results prediction is very important. There is a huge demand for the algorithm that best predicts the result of cricket because of its popularity and huge amount of money involved in the game. Thus the analysis of IPL results becomes more important. Prediction of the outcome of a match using machine learning algorithms is an important aspect in cricket. Records of the past performance of players and other related data can be analyzed to create models that predict the winning team. This model can be created using the machine learning algorithms such as Decision Tree, Naive Bayes and K-Nearest neighbor and their results can be compared based on the Evaluation Measures as accuracy, precision, recall, sensitivity and error rate. The proposed paper is organized as follows. In section 2, various works in the field are discussed and the gap in exploring using machine learning techniques available in R has been highlighted. Section 3 discusses the methodology of the approaches applied in this paper using a block diagram. Results and discussions are detailed in section 4. This section explores the results for better understanding. It is also important that the performance metrics derived from the models prove the high accuracy and efficiency of the built model. Section 5 concludes the work done in this paper.

LITERATURE REVIEW

The research conducted on Data Mining of Cricket datasets explores the application of various data mining techniques, including Decision Tree, Naive Bayes, KNN, and Random Forest, on the IPL dataset. The objective is to build a model that can predict the outcomes of matches. To achieve this, the study employs feature selection methods such as Wrapper and Ranker to identify the best attributes for classification. The popular data mining tool, WEKA, is utilized throughout the research process.

According to Gupta et al. [2], selecting the best team is crucial for achieving optimal outcomes in IPL. This paper proposes an optimal solution that employs Data Mining Techniques instead of traditional and laborious methods. By leveraging these techniques, the selection of the best team becomes more efficient, enhancing their chances of becoming champions.

In another study [3], the authors introduce a fuzzy clustering logic to group the IPL batting statistics into relevant clusters using the Data Mining Technique - Clustering. The approach, implemented using MATLAB, offers accurate results and effectively classifies the fuzzy data into appropriate clusters.

Raza Ul Mustafa et al. [4] present a study investigating the feasibility of using Twitter data to forecast match results. The research explores the effectiveness of machine learning techniques, including Support Vector Machine, Naive Bayes Classifier, and Linear Regression, when applied to data collected from social media networks and real-world events. Notably, the Support Vector Machine technique demonstrates promising results.

The work on Live Cricket Score and Winning Prediction [5] focuses on building a model that predicts the score for the chasing team and estimates the second innings' score. The model incorporates Linear Regression, Naive Bayes Classifier, and Reinforcement Learning Algorithm. Factors such as toss result, team ranking, and home team advantage are considered in the prediction process.

Sankaranarayanan [6] proposes a prediction system that utilizes historical data to forecast the victory or loss of forthcoming matches. The system applies Linear Regression, Nearest Neighboring, and clustering methods to provide mathematical results that exhibit the performance of the prediction algorithm.

Parag Shah [7] introduces a model that predicts the outcome of live matches after each ball. The model incorporates the concept of the par score, using Duckworth & Lewis methods, and calculates the probability of which team will win the match.

Kaluarachchi [8] employs artificial intelligence techniques, specifically bayes classifiers in machine learning, to classify factors that affect match results, such as home game advantage, day/night effect, toss, and batting order. The outcome of this work is a software tool called CricAI, which provides the probability of winning based on input factors. CricAI can be applied in real-world cricket scenarios to modify certain factors and increase the chances of winning.

The models mentioned above utilize data analytics methods from the machine learning domain. However, predicting match results can be challenging in rain-affected matches. In such cases, factors like batting, bowling, fielding, team selection, result prediction, and target revision become crucial. Mathematical models based on insights from previous matches can aid in predicting the outcome. The Predictor models in this research leverage Support Vector Machine (SVM). Additionally, a proposed approach named Deep Mayo Predictor [9] is employed to address rain-affected matches.

Jayshree Hajgude [10] has developed a software tool based on Statistical Analysis and Data Mining techniques, which forms a dream team with Bayesian Prediction Technique and parameter-based filtering. The tool utilizes a database containing details of current IPL players, allowing for data mining and statistical analysis of each player.

In the study focused on predicting cricket scores and player performance using ODI datasets [11], a predictive model is developed. The model employs supervised methods like SVM and Naïve Bayes, as well as clustering methods like KNN and MLP, to achieve.

METHODOLOGY

The proposed method consists of five sub modules, namely, loading the dataset, pre-processing, feature selection, classification using various algorithms and comparison of algorithms as shown in Fig.1.

3.1 LOADING THE DATASET

The dataset used in this study is called "matches.csv," which contains IPL matches data from 2008 to 2017. The dataset, obtained from the Kaggle Repository, has a size of 117,096 bytes. It consists of 18 attributes and a total of 637 records. The attributes present in the dataset include:

1. id: Unique identifier for each match.
2. season: The IPL season in which the match was played.
3. city: The city where the match took place.
4. date: The date on which the match was held.
5. team1: The first team participating in the match.
6. team2: The second team participating in the match.
7. toss_winner: The team that won the toss.
8. toss_decision: The decision made by the team winning the toss.
9. result: The result of the match (normal, tie, no result).
10. dl_applied: Whether the Duckworth-Lewis method was applied (1 if applied, 0 otherwise).
11. winner: The team that won the match.
12. win_by_runs: The margin of victory in terms of runs.
13. win_by_wickets: The margin of victory in terms of wickets.
14. player_of_match: The player awarded as the Man of the Match.
15. venue: The stadium where the match was played.
16. umpire1: The first umpire officiating the match.

17. umpire2: The second umpire officiating the match.
18. umpire3: The third umpire officiating the match.

To load the dataset into the Python tool, the command `read.csv()` is used to upload the data. The loaded dataset is stored in a variable named "IPL data."

3.2 DATA PREPROCESSING

Data preprocessing plays a crucial role in machine learning as it converts raw data into a usable format, often serving as an initial step to clean and prepare the data for further processing and classification. In this study, data pre-processing is applied to transform the dataset into a format that is easier to process and minimizes errors.

The first step in data pre-processing involves handling missing data. Null attributes and records containing NA attributes are removed from the dataset. Since the attribute "umpire3" has no values, it is initially eliminated from the dataset.

Next, the fields "date" and "player_of_match" are converted to numeric fields for ease of processing. Records with NA values in the "winner" and "player_of_match" attributes are also removed from the dataset.

To simplify the "winner" attribute, the levels within it are dropped, resulting in a non-factor variable. These pre-processing steps, including the removal of null attributes, NA records, and unnecessary levels, must be performed prior to feature selection and classification techniques.

By conducting these data pre-processing steps, the dataset is prepared for subsequent analyses, ensuring a more accurate and efficient classification process.

3.3 FEATURE SELECTION

Feature selection, also known as variable selection, is a crucial phase in machine learning as it improves performance by eliminating redundant and irrelevant features from the dataset. This process not only enhances the learning task but also speeds it up. In this study, two functions, namely Boruta() and importance(), are utilized for feature selection.

The Boruta() function, available in the Boruta package, conducts a narrow-down search for relevant features by comparing them with the original attributes. On the other hand, the importance() function, found in the randomForest package, estimates the importance of features by using their permuted copies and progressively eliminating irrelevant ones through stabilization tests.

By applying the Boruta() function, the features "date," "dl_applied," and "umpire2" are confirmed as unimportant. Similarly, the importance() function identifies "umpire1," "umpire2," "venue," "result," and "dl_applied" as having the least Mean Decrease Accuracy. Consequently, through a comparison of both algorithms, it is determined that the fields "umpire1," "umpire2," "venue," "dl_applied," and "result" should be removed from the dataset.

By performing feature selection using the Boruta() and importance() functions, the dataset is refined to include only the most relevant and significant attributes, thus improving the efficiency and accuracy of the subsequent classification tasks.

3.4 CLASSIFICATION

In the field of machine learning, classification is a fundamental technique used to categorize data into different classes. It is a supervised learning method where a computer program learns from training data and uses this knowledge to classify new data instances. In this study, four different classification algorithms are applied: Decision Tree, Random Forest, Naive Bayes, and K-Nearest Neighbor.

3.4.1 Decision Tree: Decision Tree is a supervised learning algorithm that can be used for both classification and regression tasks. It represents decisions and their potential outcomes using a tree-like structure. The "caret" and "rpart.plot" packages are utilized in this study. Functions such as rpart(), createDataPartition(), trainControl(), train(), prp(), and predict() are employed to implement and obtain the results of the Decision Tree algorithm.

3.4.2 Random Forest: Random Forest is another supervised learning method used for classification, regression, and other tasks. It is an ensemble-based method that generates multiple decision trees. The "randomForest" package is employed, which includes functions like sample(), randomForest(), and plot() to implement and visualize the Random Forest algorithm and obtain accurate results.

3.4.3 Naive Bayes: Naive Bayes classifier is a supervised learning algorithm that works based on Bayes' theorem. It uses conditional probability to classify data instances. Naive Bayes classifiers assume independence between attributes of data points, and they are commonly used in text categorization problems. The "naivebayes" package in R provides the naïve Bayes function () that is used to implement this algorithm and obtain classification results.

3.4.4 K-Nearest Neighbor: K-Nearest Neighbor (KNN) is a simple yet powerful classification algorithm. It determines the class of a data point based on the classes of its nearest neighbors. KNN can be used for both classification and regression tasks. In this study, the "RWeka" package and the IBk() function are used to implement the KNN algorithm and obtain classification results.

By applying these four classification algorithms, the study aims to analyze and compare their performance in classifying the dataset, ultimately identifying the most suitable algorithm for the task at hand.

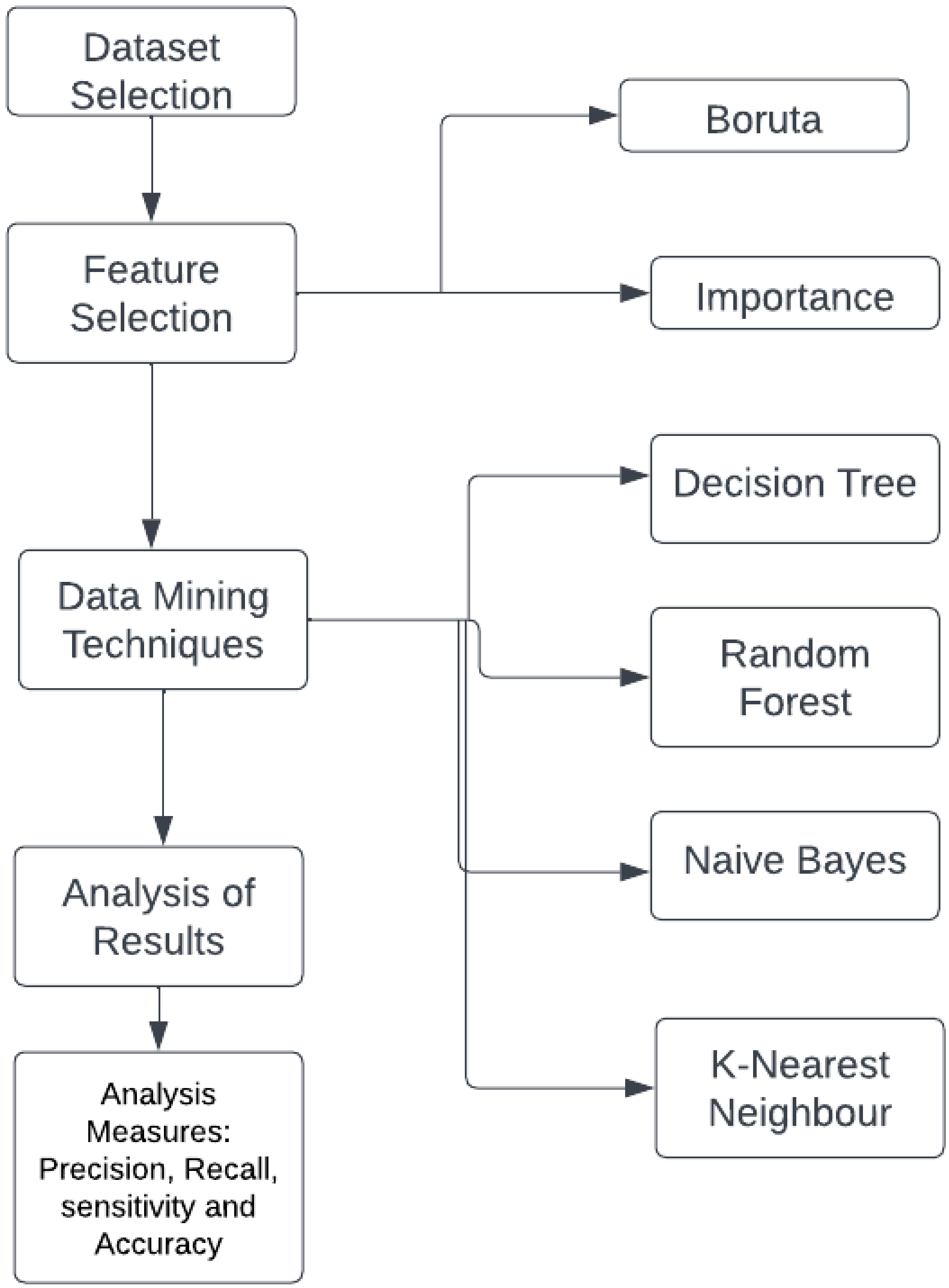
3.5 COMPARISON OF CLASSIFICATION ALGORITHMS

The selection of the best classification algorithm for a given dataset is a crucial task in order to obtain optimal results. It involves making various methodological choices, which can be complex. One important aspect is the assessment of classification performance and the ranking of algorithms. In this study, we focus on popular performance measures and discuss their properties. Over the years, numerous measures have been proposed to evaluate classification performance.

A commonly used tool in machine learning is the confusion matrix, also known as an error matrix, which provides a visual representation of a classification model's performance. Each row of the matrix represents samples of the predicted class. In predictive analytics, a confusion matrix is a 2x2 matrix that reports the counts of false positives, false negatives, true positives, and true negatives in the total sample. Performance measures derived from the confusion matrix include accuracy, true negatives, precision, recall, sensitivity, specificity, and error rate.

All of the performance measures mentioned above are based on the concepts of the positive class (P), negative class (N), true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

Refer to Fig.1 for the methodology diagram.



P: Positive Class = True Positive + False Negative = Samples predicted as CSK N: Negative Class = False Positive + True Negative = Samples predicted as Non-CSK

In this study, we aim to utilize these performance measures to evaluate and compare the classification algorithms applied to the dataset. By analyzing the properties and results obtained from these measures, we can make informed decisions regarding the best algorithm for the given task.

Table.1. Prediction Results

	Actual/Predicted	Predicted	
Actual		CSK	Non CSK
	CSK	TP	FN
	Non CSK	FP	TN

Based on the factors mentioned above, let's discuss the performance measures of the classifiers:

1. Accuracy: Accuracy measures the rate of correct classifications made by the classifier. Accuracy = (TP + TN) / (P + N)
2. Precision: Precision is the ratio of true positives among the total number of instances classified as positive. Precision = TP / (TP + FP)
3. Recall: Recall, also known as sensitivity or true positive rate, is the ratio of true positives among the true positives and false negative instances. Recall = TP / (TP + FN)
4. Sensitivity: Sensitivity measures the ratio of positive instances that are correctly identified as positive by the classifier. Sensitivity = TP / P
5. Specificity: Specificity measures the ratio of negative instances that are correctly identified as negatives by the classifier. Specificity = TN / N
6. Error rate: Error rate quantifies the rate of inaccurate predictions made by the classification algorithm. Error Rate = (FP + FN) / (P + N)

These performance measures provide valuable insights into the classification performance of the algorithms and help evaluate their effectiveness in correctly classifying instances. By analyzing these measures, we can assess the strengths and weaknesses of each classifier and make informed decisions regarding their suitability for the given task.

4. RESULTS AND DISCUSSIONS

The dataset is loaded into the R software using the read.csv() function, and preprocessing steps are performed. For feature selection, the Boruta algorithm is utilized, which conducts a narrow-down search. It assesses the importance of the original attributes and extracts the best features. By eliminating irrelevant features, it enhances the stability of the performance test.

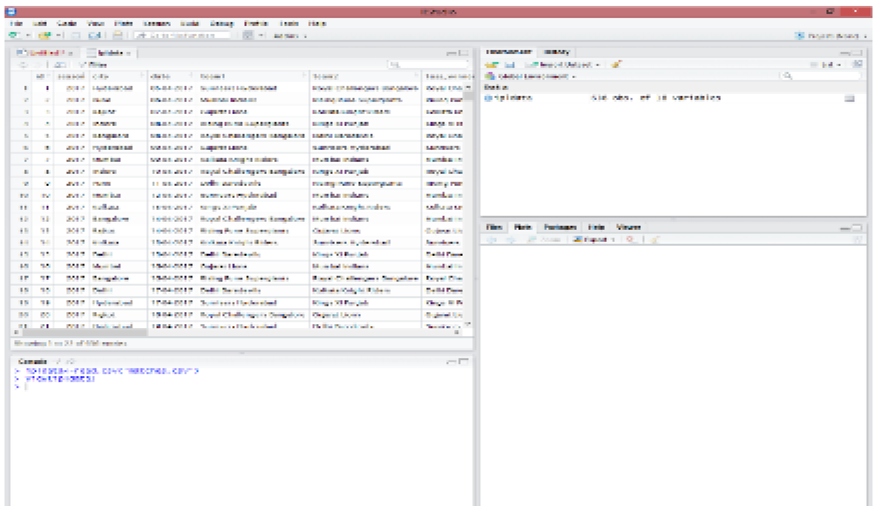


Fig.2. Loading of the Dataset

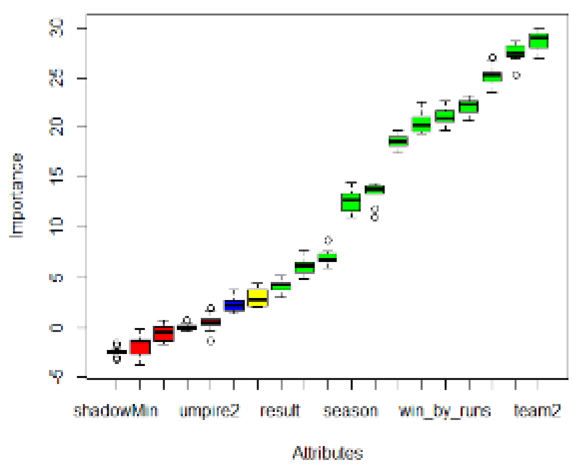


Fig.3. Results of Feature Selection using Boruta

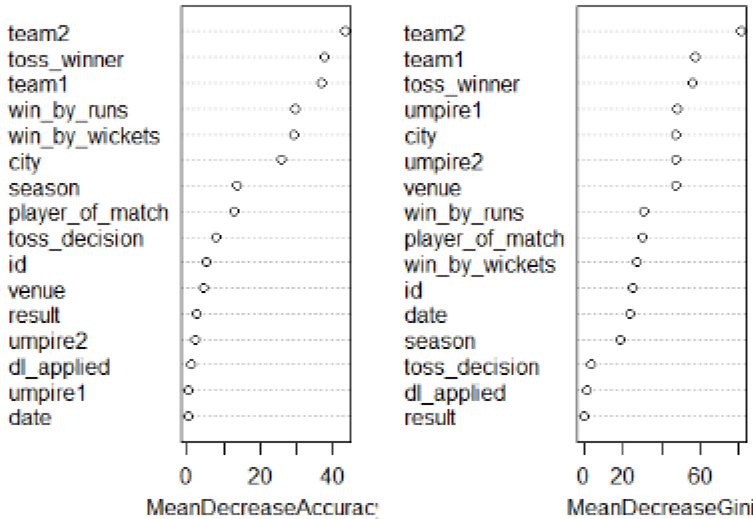


Fig.4. Results of Feature Selection using importance

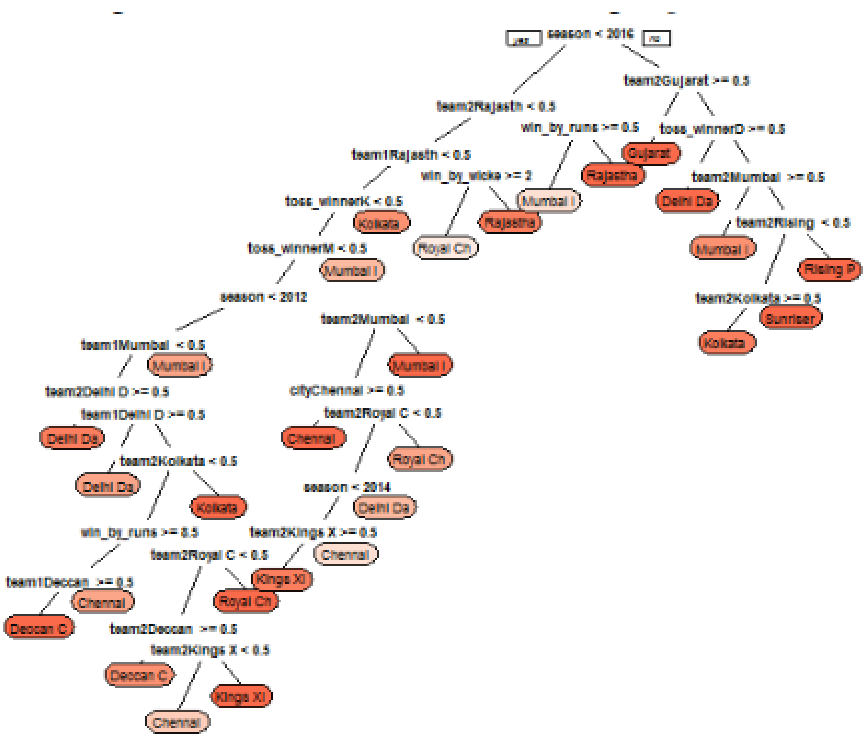


Fig.5. Decision Tree Results

Table.2. Decision Tree – Confusion Matrix

	CSK	DC	DD	GL	KXIP	KTK	KKR	MI	PW	RR	RPS	RCB	SRH
CSK	7	0	1	0	2	0	0	2	1	0	0	0	2
DC	0	2	0	0	0	0	0	0	0	0	0	0	0
DD	0	1	8	0	3	0	0	1	0	0	0	0	1
GL	0	0	0	2	0	0	0	0	0	0	0	0	1

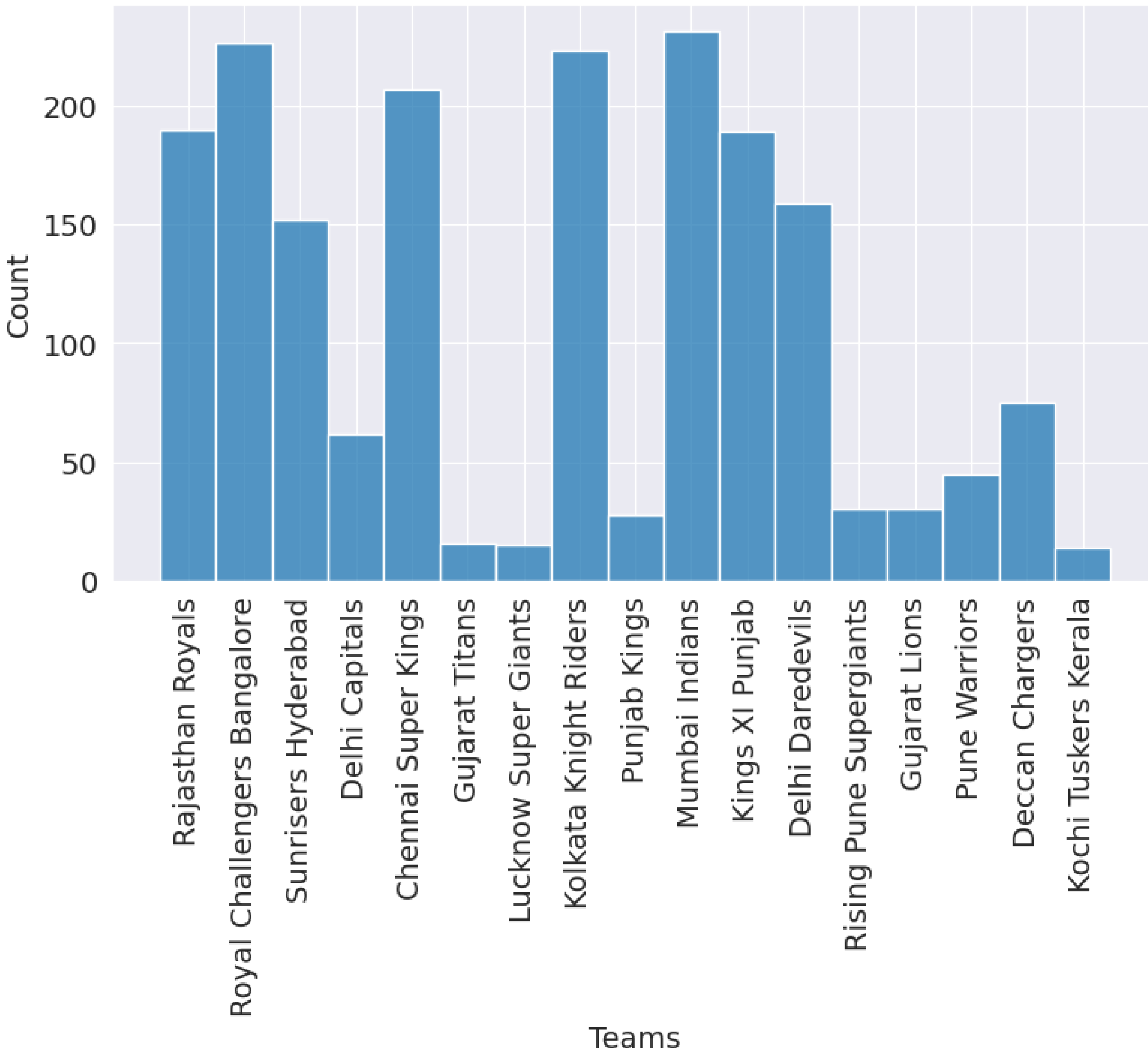
KXIP	0	1	0	0	6	0	0	0	0	0	0	0	0
KTK	0	0	0	0	0	0	0	0	0	0	0	0	0
KKR	3	1	1	0	1	0	9	1	0	0	0	0	1
MI	3	0	1	0	1	1	0	12	1	0	0	0	1
PW	0	0	0	0	0	0	0	0	0	0	0	0	0
RR	0	0	0	0	0	0	0	0	0	12	0	0	0
RPS	0	0	1	0	0	0	0	0	0	0	7	1	1
RCB	2	0	0	0	0	0	4	0	0	0	1	0	6
SRH	0	0	0	0	0	0	1	1	0	0	1	1	1

Table.3. Random Forest - Confusion Matrix

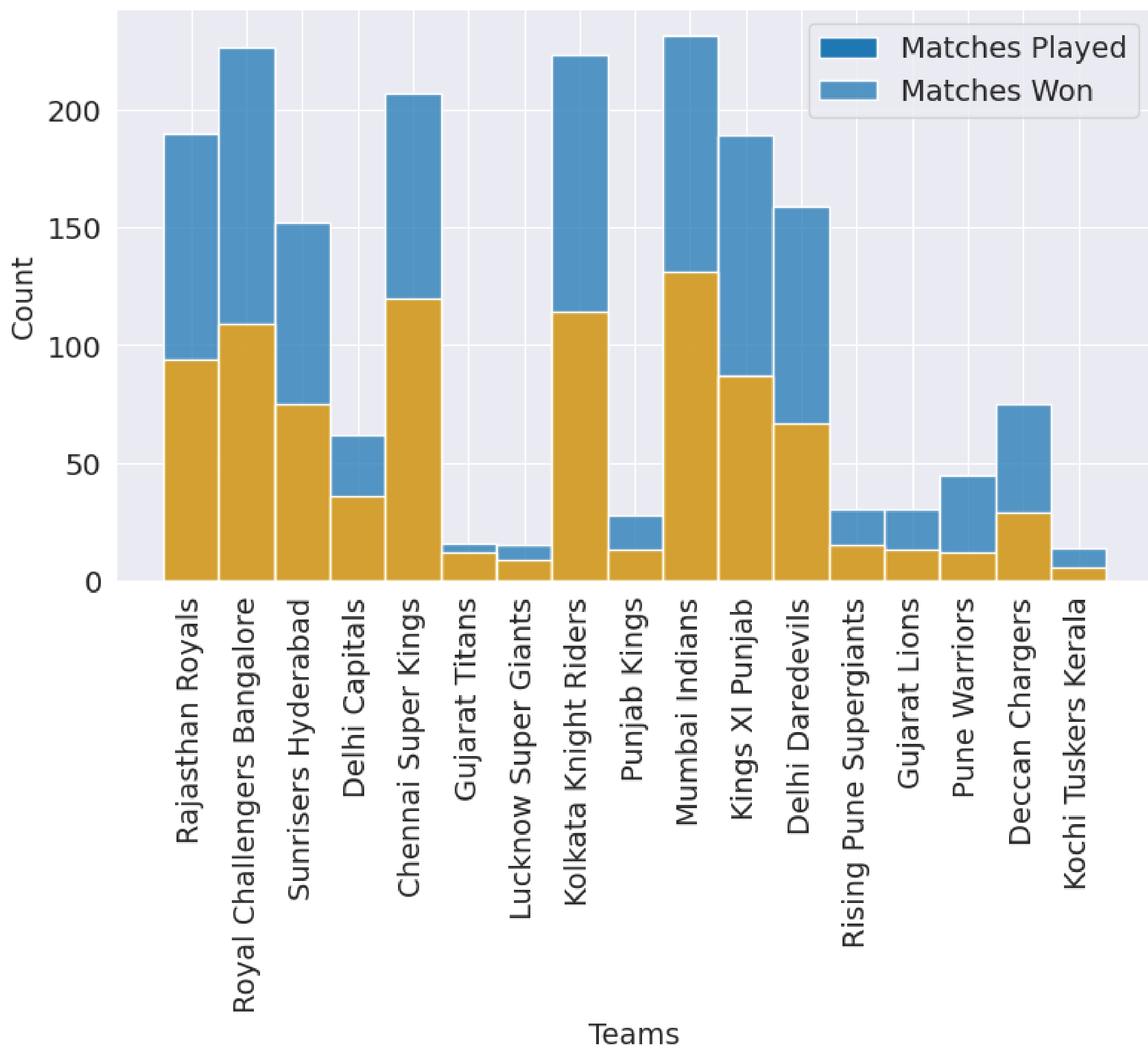
	CSK	**DC**	**DD**	**GL**	**KXIP**	**KTK**	**KKR**	**MI**	**PW**	**RR**	**RPS**	**RCB**	**SRH**
CSK	50	0	0	0	1	0	2	4	0	3	0	0	0
DC	3	9	1	0	1	0	1	0	0	2	0	1	0
DD	2	1	28	0	3	0	0	2	0	3	0	3	0
GL	0	0	1	6	1	0	0	0	0	0	0	1	0
KXIP	4	0	2	0	31	0	5	4	0	2	0	2	2
KTK	0	0	0	0	0	3	2	0	0	1	0	0	0
KKR	1	0	2	0	3	0	50	4	0	0	0	4	1
MI	1	0	0	0	2	0	1	53	0	2	0	2	2
PW	2	0	1	0	2	0	2	1	1	1	0	0	0
RR	1	0	1	0	3	0	2	0	0	37	0	1	0
RPS	0	0	0	0	0	0	2	1	0	0	7	1	1
RCB	4	1	1	0	3	0	0	1	0	2	1	44	0
SRH	1	0	1	0	1	0	2	2	0	0	1	0	26

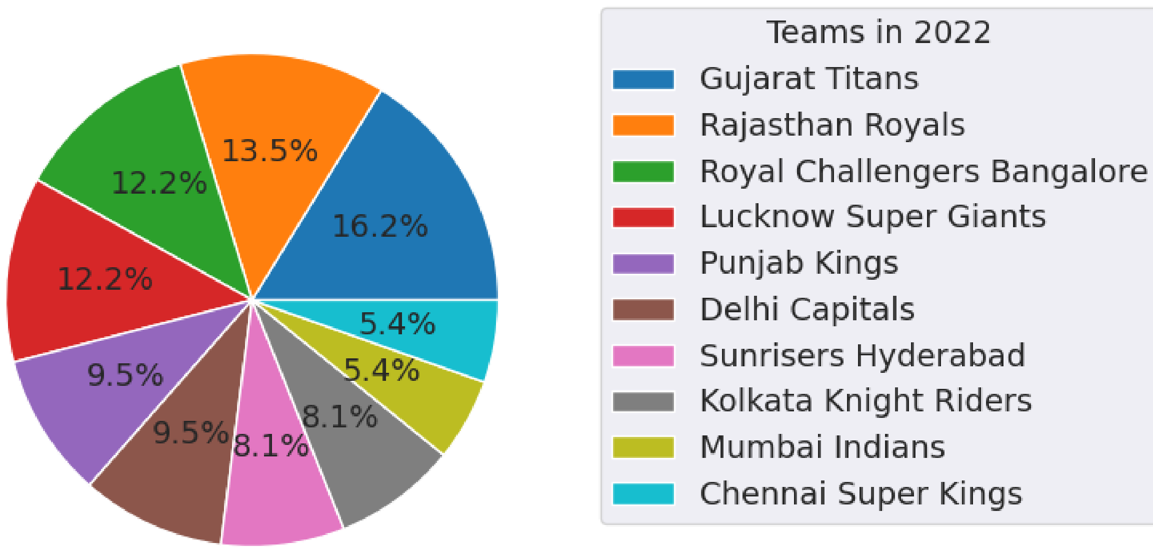
Table 4. Naive Bayes – Confusion Matrix

	CSK	**DC**	**DD**	**GL**	**KXIP**	**KTK**	**KKR**	**MI**	**PW**	**RR**	**RPS**	**RCB**	**SRH**
CSK	51	1	2	0	4	0	3	3	1	1	0	3	0
DC	1	24	3	0	2	0	2	3	1	0	0	3	0
DD	4	2	37	1	2	0	4	1	1	2	0	6	1
GL	0	0	3	11	2	0	0	2	0	0	0	2	0
KXIP	4	1	2	0	42	0	3	5	0	3	0	3	4
KTK	0	0	0	0	0	6	0	0	0	0	0	0	0
KKR	3	0	2	0	3	0	54	4	0	1	0	4	1
MI	5	0	2	0	5	0	1	55	1	3	2	6	1
PW	2	0	2	0	1	0	1	1	8	1	0	0	0
RR	3	0	2	0	1	0	1	3	0	46	0	3	2
RPS	0	0	1	0	0	0	0	1	0	0	10	1	0
RCB	2	1	3	0	6	0	3	6	0	3	1	38	2
SRH	1	0	2	0	1	0	2	2	0	2	1	1	29

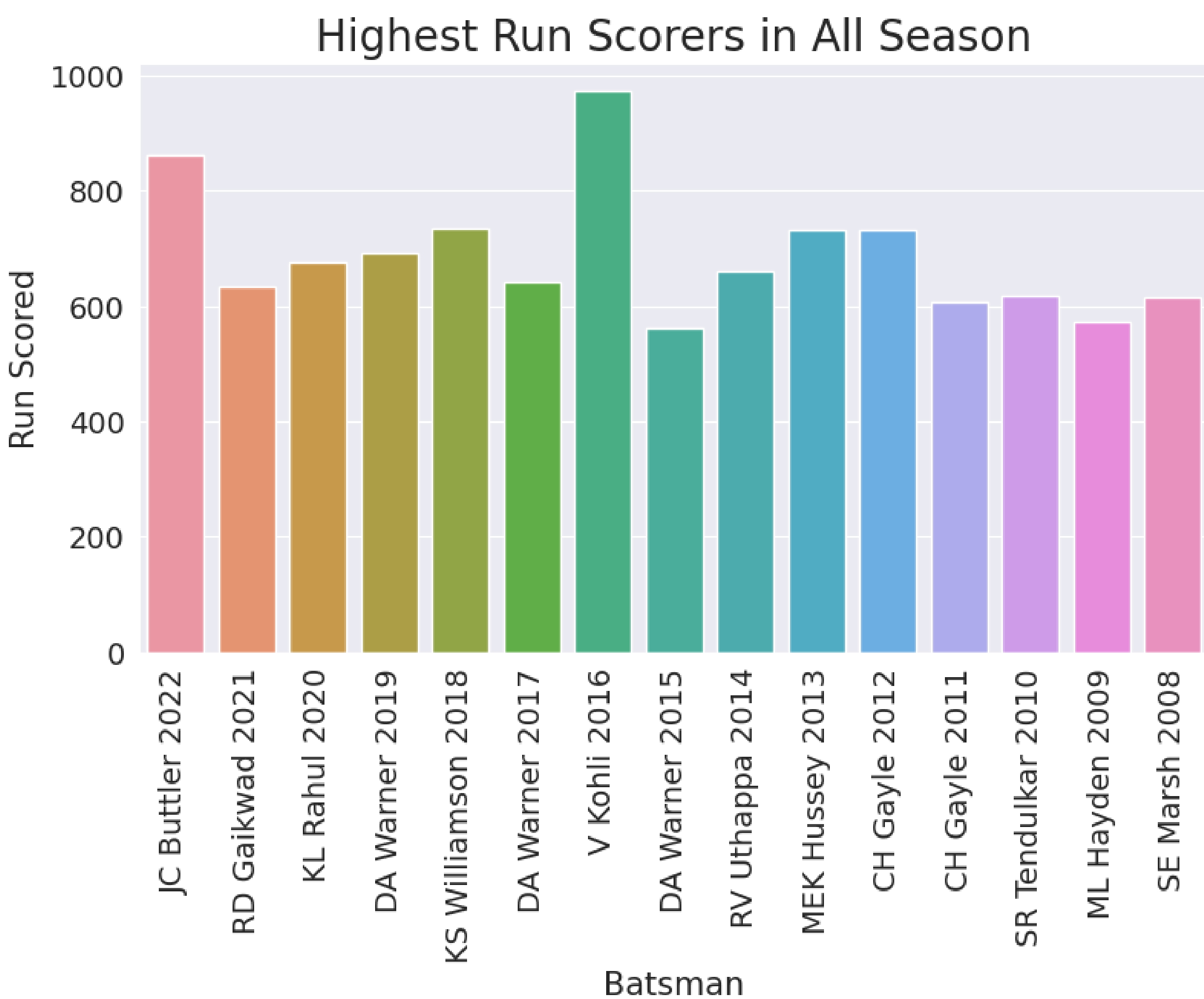


Count
0
50
100
150
200
Rajasthan Royals
Royal Challengers Bangalore
Sunrisers Hyderabad
Delhi Capitals
Chennai Super Kings
Gujarat Titans
Lucknow Super Giants
Kolkata Knight Riders
Punjab Kings
Mumbai Indians
Kings XI Punjab
Delhi Daredevils
Rising Pune Supergiants
Gujarat Lions
Pune Warriors
Deccan Chargers
Kochi Tuskers Kerala
Teams

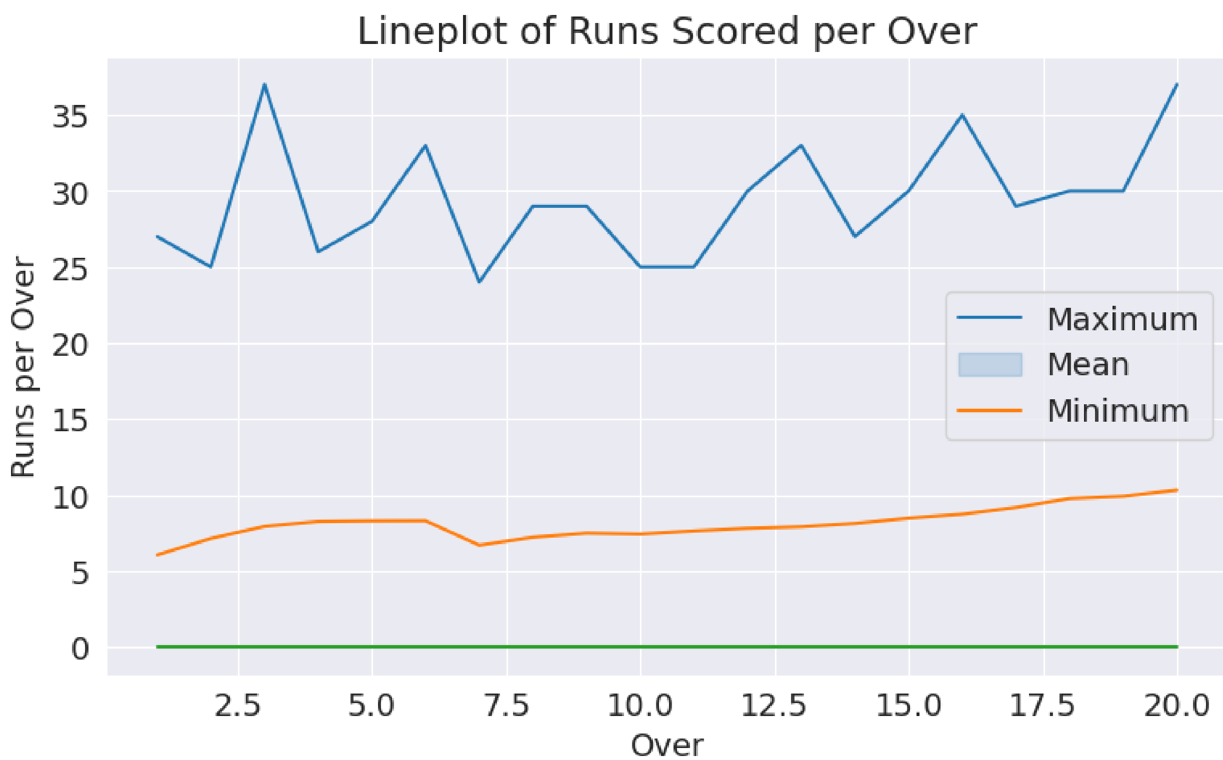

Matches Played
Matches Won
Count
0
50
100
150
200
Rajasthan Royals
Royal Challengers Bangalore
Sunrisers Hyderabad
Delhi Capitals
Chennai Super Kings
Gujarat Titans
Lucknow Super Giants
Kolkata Knight Riders
Punjab Kings
Mumbai Indians
Kings XI Punjab
Delhi Daredevils
Rising Pune Supergiants
Gujarat Lions
Pune Warriors
Deccan Chargers
Kochi Tuskers Kerala
Teams



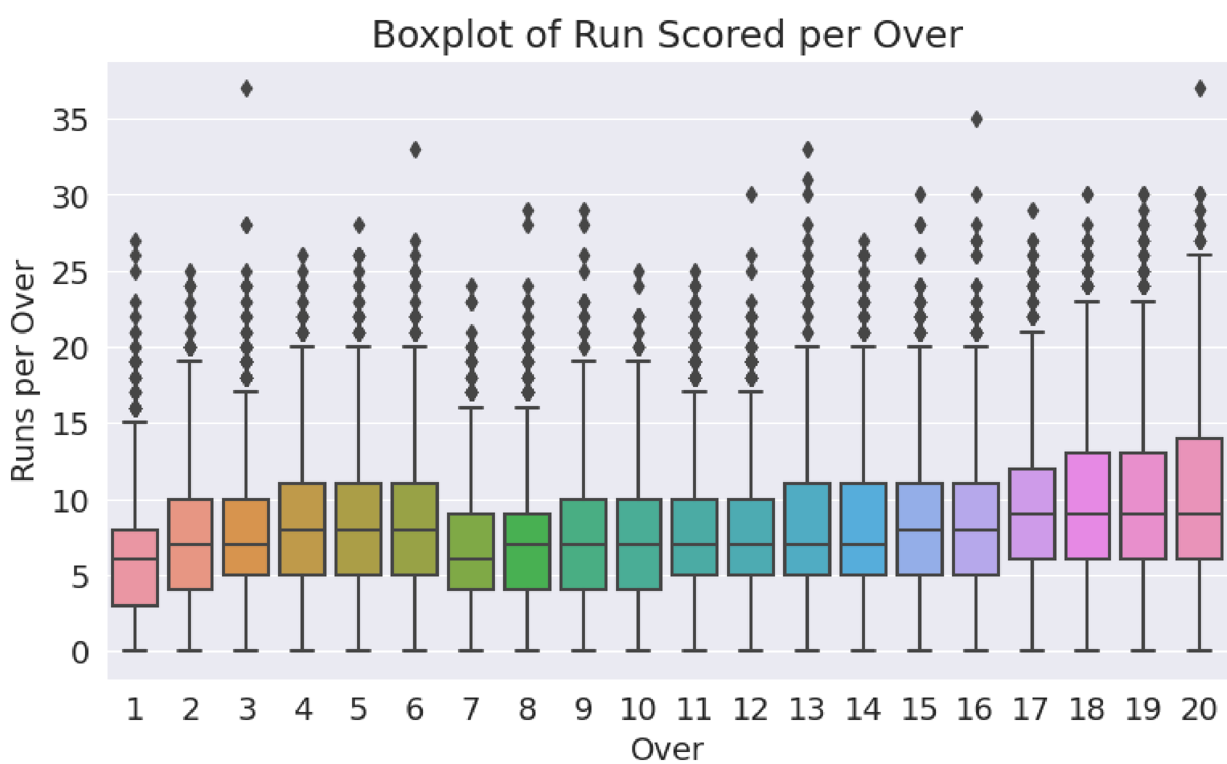
Teams in 2022
Gujarat Titans
Rajasthan Royals
Royal Challengers Bangalore
Lucknow Super Giants
Punjab Kings
Delhi Capitals
Sunrisers Hyderabad
Kolkata Knight Riders
Mumbai Indians
Chennai Super Kings
16.2%
13.5%
12.2%
12.2%
9.5%
9.5%
8.1%
8.1%
5.4%
5.4%



Highest Run Scorers in All Season
Run Scored
Batsman
0
200
400
600
800
1000
JC Buttler 2022
RD Gaikwad 2021
KL Rahul 2020
DA Warner 2019
KS Williamson 2018
DA Warner 2017
V Kohli 2016
DA Warner 2015
RV Uthappa 2014
MEK Hussey 2013
CH Gayle 2012
CH Gayle 2011
SR Tendulkar 2010
ML Hayden 2009
SE Marsh 2008



Lineplot of Runs Scored per Over
Runs per Over
Over
Maximum
Mean
Minimum
35
30
25
20
15
10
5
0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0



Boxplot of Run Scored per Over
Runs per Over
Over
35
30
25
20
15
10
5
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

The dataset for the four classification algorithms has been divided into training and testing subsets. A model was built using the training subset. Then, the model was used to test the data from the testing subset and make predictions, which were used to create the confusion matrices.

The confusion matrices were formed based on the results obtained from the testing data using the predict() function. In order to assess accuracy and predict results, it is crucial to consider the overall winning pattern. Taking into account the past ten-year records of the IPL results, CSK has won two times, KKR has won two times, and MI has won three times. The accuracy of the predictions was derived by analyzing the confusion matrices, which include values such as true positive, false positive, true negative, and false negative.

Additionally, based on the visualization of the decision tree, it can be inferred that KKR has a higher chance of winning based on the past ten-year history. Furthermore, all four algorithms consistently predict KKR as the team with the highest probability of winning, as indicated by the results obtained from the confusion matrices.

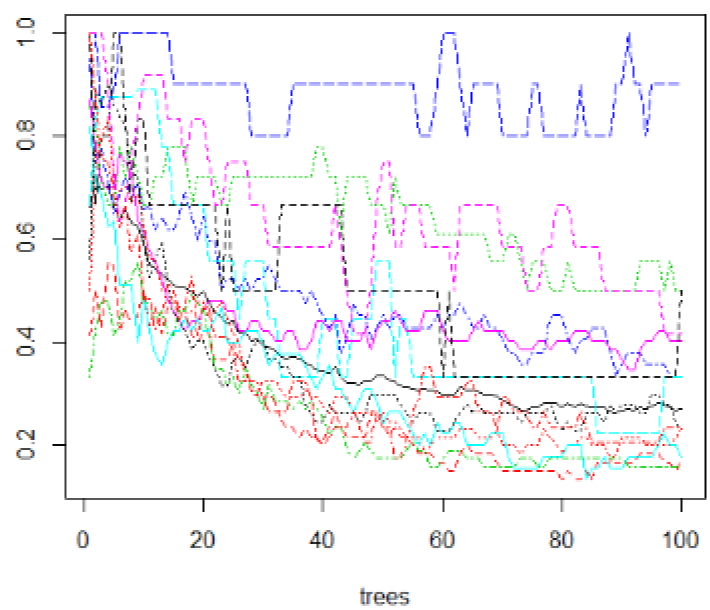


Fig.6. Random Forest Result

Based on the results mentioned above, we can conclude that the Random Forest algorithm is performing better than the other algorithms. It exhibits outstanding measures, including high accuracy and a lower error rate. The performance of the Random Forest algorithm surpasses that of the other algorithms used in the analysis.

5 CONCLUSION & FUTURE SCOPE

This work aims to explore and understand the dataset of the past 10 years of IPL history. It focuses on comprehending the working principles of four different machine learning algorithms and their implementation in R. The process involves creating a model and training dataset, utilizing the model to make predictions, and comparing the results. Several measures such as accuracy, error rate, precision, recall, sensitivity, and specificity are taken into consideration.

Based on the evaluation of these measures, the Random Forest algorithm is identified as the best performing algorithm. It demonstrates high accuracy, low error rate, and favorable precision, recall, sensitivity, and specificity values. The main focus of this work is to delve into the IPL data, extract valuable insights, and present them through graphical representations and comparative analysis.

By leveraging the findings of this analysis, both the Indian Premier League and its fan followers can make informed decisions regarding team performance and predict potential trophy winners. This knowledge can contribute to the success of teams in the future.

- Catboost is by far the best model for predicting the best 11.
- If Credits were involved as constraints, the error rate may have reduced.
- Feature Engineering can result in more filtered data that may provide with better prediction results.
- Can develop an approach which accounts for the "form" of players with time and other factors such as age.

6 REFERENCES

[1] S. Abhishek, Ketaki V. Patil, P. Yuktha and S. Meghana, Predictive Analysis of IPL Match Winner using Machine Learning Techniques", International Journal of Innovative Technology and Exploring Engineering, Vol. 9, No. 1, pp. 430-435, 2019.

[2] Sanjay Gupta, Hitesh Jain, Asmit Gupta and Hemant Soni, "Fantasy League Team Prediction", International Journal of Research in Science and Engineering, Vol. 6, No. 3, pp. 97-103, 2017.

[3] Pabitra Kumar Dey, Gangotri Chakraborty, Purnendu Ruj and Suvobrata Sarkar, "A Data Mining Approach on Cluster Analysis of IPL", International Journal of Machine Learning and Computing, Vol. 2, No. 4, pp. 351-354, 2012.

[4] Raza Ul Mustafa, M. Saqib Nawaz, M. Ikram Ullah Lali, Tehseen Zia and Waqar Mehmood, "Predicting the Cricket Match outcome using Crowd Opinions on Social Networks: A Comparative Study of Machine Learning Methods", Malaysian Journal of Computer Science, Vol. 30, No. 1, pp. 63-76, 2017.

[5] Rameshwari Lokhande and P.M. Chawan, "Live Cricket Score and Winning Prediction", International Journal of Trend in Research and Development, Vol. 5, No. 1, pp. 30- 32, 2018.

[6] Vignesh Veppur Sankaranarayanan and Junaed Sattar,"Auto-play: A Data Mining Approach to ODI Cricket Simulation and Prediction", Proceedings of SIAM Conference on Data Mining, pp. 1-7, 2014.

[7] Parag Shah, "Predicting Outcome of Live Cricket Match using Duckworth-Lewis Par Score", International Journal of Latest Technology in Engineering, Management and Applied Science, Vol. 6, No. 7, pp. 72-75, 2017.

[8] Amal Kaluarachchi and S. Aparna, "A Classification based Tool to Predict the outcome in ODI Cricket", Proceedings of 5th International Conference on Information and Automation for Sustainability, pp. 233-237, 2010.

[9] C. Deep Prakash Dayalbagh, C. Patvardhan and C. Vasantha Lakshmi, "Data Analytics based Deep Mayo Predictor for IPL-9", International Journal of Computer Applications, Vol. 152, No. 6, pp. 6-11, 2016.

[10] Jayshree Hajgude, Aishwarya Parameshwaran, Krishna Nambi, Anupama Sakhalkar and Darshil Sanghvi, "IPL Dream Team-A Prediction Software Based on Data Mining and Statistical Analysis", International Journal of Computer Engineering and Applications, Vol. 9, No. 4, pp. 113-119, 2015.

[11] Sonu Kumar and Sneha Roy, "Score Prediction and Player Classification Model in the Game of Cricket using Machine Learning", International Journal of Scientific and Engineering Research, Vol. 9, No. 2, pp. 237-242, 2018.