IEE-577 FINAL REPORT-GROUP 5

Venkata Saisrikar Gudivada (1217522111)
Pranay Reddy Vancha (1217852909)

Analyzing and Predicting the Outcome of IPL Matches

INTRODUCTION:

Cricket, the most exciting and fascinating game that the people of all age group are very crazy to see and play. It is considered to be the most interesting and uncertain game. For many it becomes a billion dollar market as they speculate financially, hope of being able to earn profit. With technology growing more and more advanced in the last few years, an in-depth acquisition of data has become relatively easy [1]. As a result, Machine Learning is becoming quite a trend in sports analytics because of the availability of live as well as historical data. Various machine learning algorithms have been applied and tested for their efficiency in solving the problems in sports [2]. The relation between machine learning and games dates back to the initial days of artificial intelligence when Arthur Samuel, a pioneer in the field of gaming and artificial intelligence studied machine learning approaches using the game of checkers [3]. Cricket is a huge business market, especially in India. Hence analysing this game, could be considered one of the most interesting data mining projects. There are hundreds of statistics to compare and analyse from, each being important in their own way. There is a great demand for prediction of the match winner, which can be used for betting purposes or winning fantasy leagues. The Indian Premier League officially Vivo Indian Premier League is a professional Twenty20 cricket league in India contested during summer. The upcoming season of 2020 IPL or IPL-13 is going to be held from 29th March to 24th May. Our Objective is to use the past data of previous seasons and predict the future outcome of a match between any of the eight IPL teams.

MOTIVATION:

Given the Grandeur of the league it is imperative than the amount of betting on the matches is huge. The IPL betting market in India is worth **$40 billion** a year according to the Doha-based International Centre for Sports Security [4], which promotes integrity and security in sports. So, this betting market has a huge potential in the coming years and many start-ups are focused on building their market share in this area. Betting on match results is one of the basic and widely punted bets. If we can provide some insights on what factors influence the match results, it can be of a great value to the sports gambling industry.

OBJECTIVE:

There are hundreds of statistics to compare and analyse from, each being important in their own way. There is great demand for prediction of the match winner. Our Objective is to use the past data of previous seasons and predict the future outcome of a match between any of the eight IPL teams. We want to leverage the data that is already present and apply various Machine Learning concepts and predict the outcome of the match.

ABOUT THE DATA:

The latest dataset was obtained from the URL https://www.kaggle.com/nowke9/ipldata which is in Kaggle and sourced from Cricsheet. The data consists of all the 756 matches played till now from the year 2008 to 2019. There are 18 columns in total, which represents various features as well as the outcome of the match. As there are some unnecessary columns, we plan to eliminate them to avoid a wrong outcome from the developed model. The key features which can determine the result of a match are opposition, toss decision, home game or away game. Also we wanted to include another important feature in determining the outcome of a game i.e., Eigen Factor (EF) score a player.

The EF score determines how much a player is contributing in the winning of their team, higher the EF score higher is his contribution. An average team, at any point in a game, has a certain likelihood of winning the game. With each change in the score, number of wickets or the number of overs, there is a change in the team's probability of winning the game. This change is credited/debited to the batsman and the bowler involved in a specific play. EF score is dependent on the game situation, for example, a six in a close game is worth more than a six in a one-sided match. The EF score of players were obtained by conducting web craping from the URL www.cricmetric.com. This is an interesting website with unique statistic and facts about the game.

DATA CLEANING:

In this section we have removed unwanted features such as win by runs, win by wickets and player of the match as they are all related to the outcome of the match and does not help in predicting the data. Also, we also removed umpires column as well, as this feature does not decide the outcome of a match as umpire is a neutral decision maker and is unbiased. We also plan on ignoring the effect of Duck-Worth method which is applied on a game only when there is an interruption of rain, considering the fact that IPL is played during summer and only 19 games were effected by rain among the total 756 matches played, naturally makes us to ignore this feature into consideration of model building. Also, there were matches which resulted in a tie or had no result, we also removed them to make sure that there were no NaN values in the train dataset.

As the data was mostly categorical, we implemented indexing technique to avoid them, the team names were replaced by the below numbers and the toss decision which was either to bat or field was replaced by 1 and 2 respectively. Although there were about 13 teams in the train dataset (2008-2018), the test dataset i.e., 2019 season has only eight teams playing.

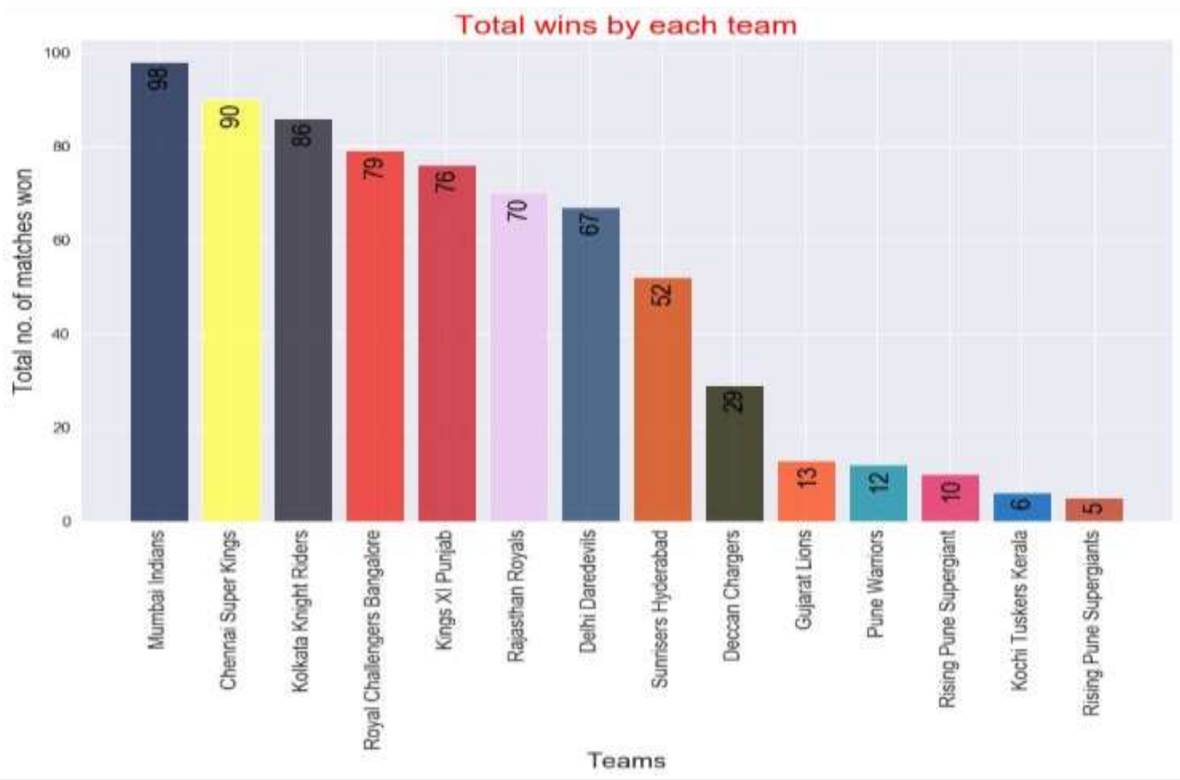
Toss Decision-Fielding	1
Toss Decision-Batting	2

Sun Risers Hyderabad (SRH)	1	Delhi Daredevils (DD)	5
Mumbai Indians (MI)	2	Kings XI Punjab (KXIP)	6
Royal Challenger Bangalore (RCB)	3	Chennai Super Kings(CSK)	7
Kolkata Knight Riders (KKR)	4	Rajasthan Royals(RR)	8

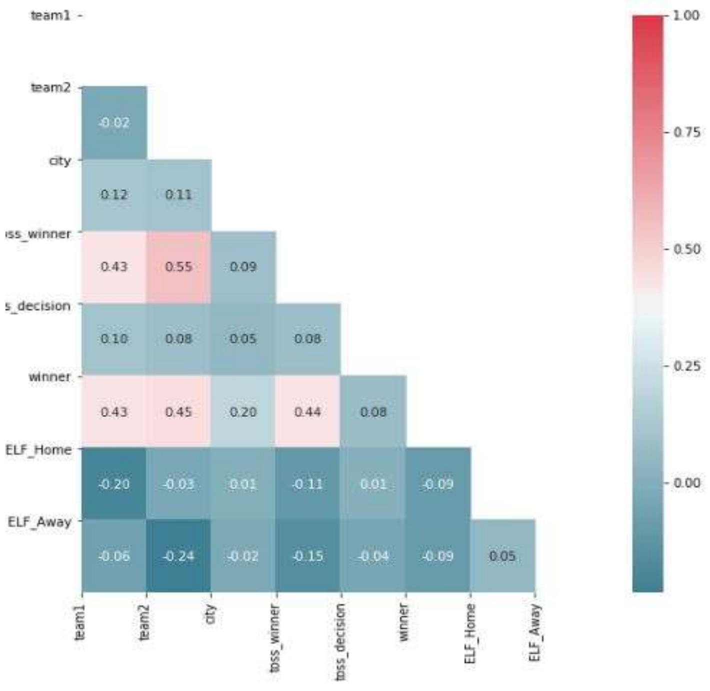
DATA ANALYSIS:

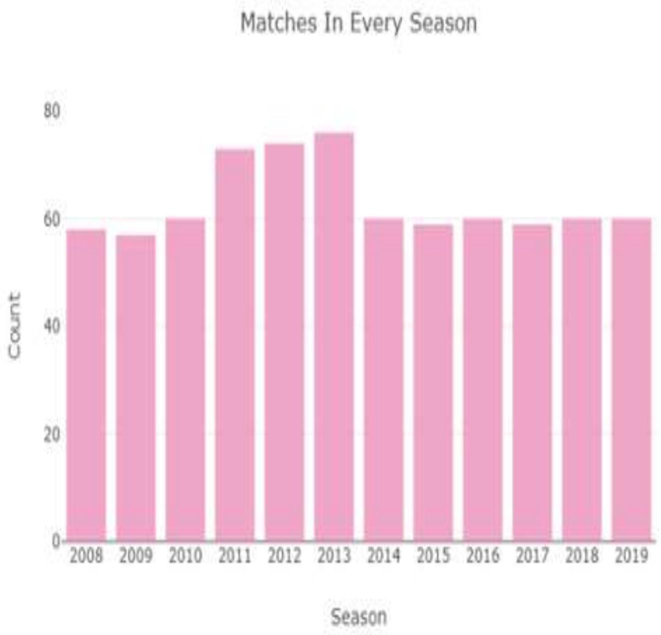
The dataset was initially analysed to get some insights into the data, graphs such most no of wins, most number of toss wins, most of number of Man of the Match awards were plotted to give a better understanding of how the features selected impact the data.

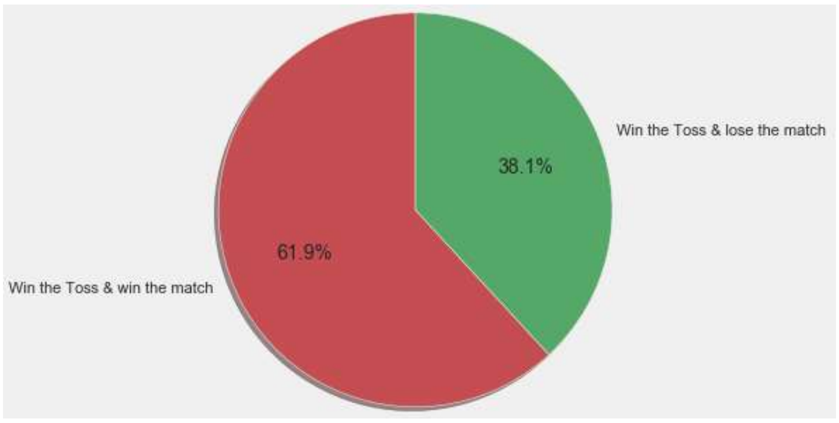
The graph below shows the most successful team in IPL history over the years i.e., Mumbai Indians.



A correlation matrix was plotted, which describes the relationship between the features and how they are correlated to each other. Correlation analysis, as a lot of analysts would know is a vital tool for feature selection and multivariate analysis in data pre-processing and exploration. Correlation helps us investigate and establish relationships between variables. This is employed in feature selection before any kind of statistical modelling or data analysis. Higher is the correlation higher is the strength between two quantitative variables.







The graph on the left shows the impact of winning the toss on the outcome of a match. It says that 62% of the times a team which has won the toss has also won the match. This is an essential and influential aspect of a match and hence we are considering this feature in building our machine learning model.

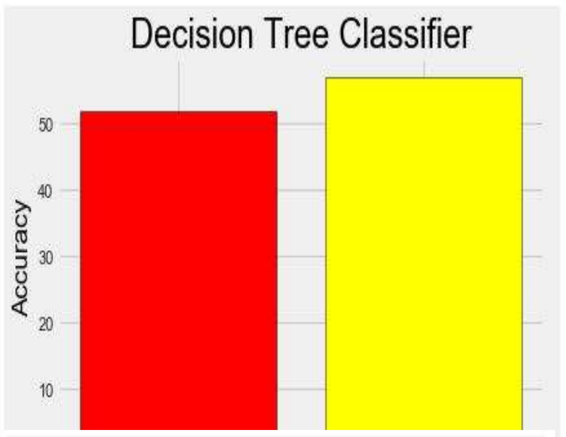
METHODOLOGY:

Once all the necessary pre-processing of the dataset is completed, then by using this cleaned dataset we plan to predict the winning team. For this to be achieved a classification model must be implemented. There will be two classes, the winning class and the loosing class. We plan on considering 4 important features after data cleaning which would be used to predict the result. Classification algorithms such as K- Nearest Neighbours, Support Vector Machines, and Logistic regression etc. were implemented to decide which class the playing teams will go into. For all the models, the objective is to find the best parameters/weights corresponding to the given features. Using the train data we will train the respective model and using the test data we predict the outcome using the model that we build, these predicted values will be compared with the actual values and the accuracy, precision score will be calculated to decide the best model among all.

Another way to make sure which model performs better, is by not considering the latest edition i.e., the 2019 edition. By this way, we only consider data from 2008 to 2018 to train the model and use the 2019 results as test data to judge the best model, which makes more sense.

Among all the classifier we found the best performing one's to be decision tree classifier, random forest classifier and gradient boosting classifier based on their accuracy and F1 scores. We also implemented stacking technique using the above three classifiers, and trained the data obtained from stacking on random forest classifier. The results were not that satisfactory and hence we did not concentrate much on stacking techniques. So for decision tree and random forest classifier we wanted to criticise the model further by tuning its hyper parameters as the accuracy was above 50%, initially we used RandomSearchCV and ran the function multiple times to obtain a closer range to optimal values and then used GridSearchCV to find the best and optimal parameter values. This technique was employed as GridSearchCV is computationally expensive in time.

DECISION TREE CLASSIFIER:

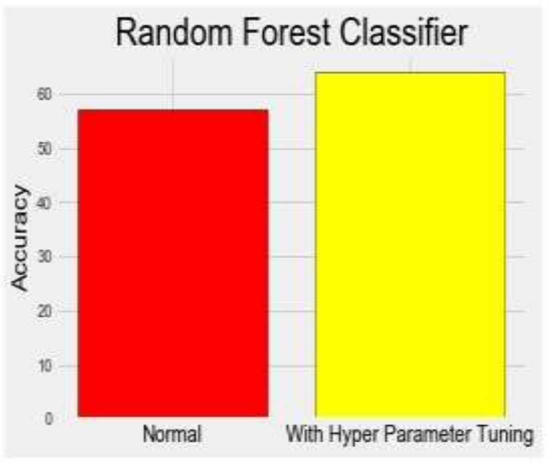


The Decision Tree Classifier yielded the following results when implemented. Accuracy without the hyper parameter tuning was **52%**. Now the parameters were trained, and the following parameter combination yielded the maximum accuracy.

1) Minimum sample split = 40
2) Maximum depth = 15

Accuracy with hyper parameter tuning was **57%**

. RANDOM FOREST CLASSIFIER:



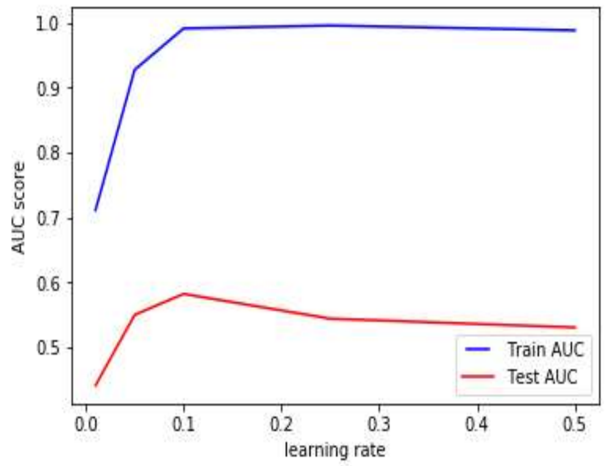
The Random Forest Classifier yielded the following results when implemented. Accuracy without the hyper parameter tuning was **57%.** Now the parameters were trained, and the following parameter combination yielded the maximum accuracy.

1) Minimum sample split = 40
2) Maximum depth = 15
3) No. of estimators = 30

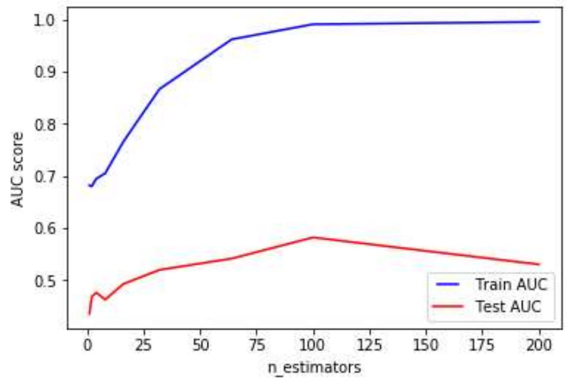
Accuracy with hyper parameter tuning was **64%**

GRADIENT BOOSTING CLASSIFIER:

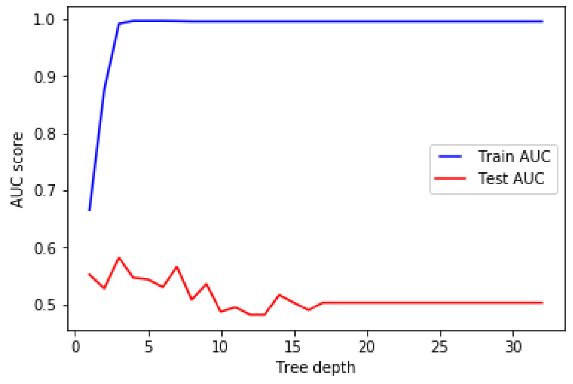
Gradient boosting classifiers are a group of machine learning algorithms that combine many weak learning models together to create a strong predictive model. Decision trees are usually used when doing gradient boosting. Gradient boosting models are becoming popular because of their effectiveness at classifying complex datasets. So as this model was giving us the best results in terms of accuracy we wanted to further criticise the model further by understanding how each parameter was influencing the learning rate. We referred a blog by Mohtadi Ben Fraj published on Medium [5].



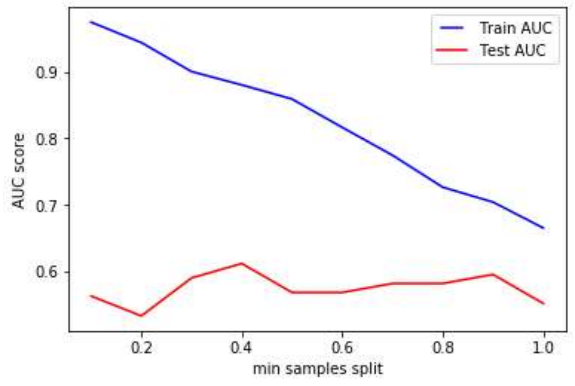
We will AUC (Area Under Curve) as the evaluation metric. Our target value is binary so it's a binary classification problem. AUC is a good way for evaluation for this type of problems. Learning rate shrinks the contribution of each tree by learning rate. We see that using a high learning rate results in overfitting the data. For this data, a learning rate of 0.1 is optimal.



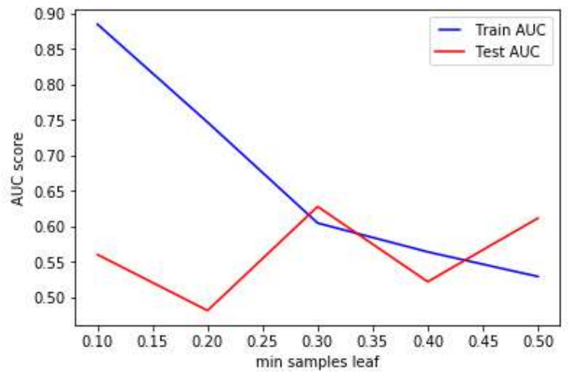
The number of estimators parameter represents the number of trees in the forest. Usually the higher the number of trees the better to learn the data. However, adding a lot of trees can slow down the training process considerably, therefore we do a parameter search to find the sweet spot. Increasing the number of estimators may result in overfitting also. In our case, using 100 trees results in optimal value.



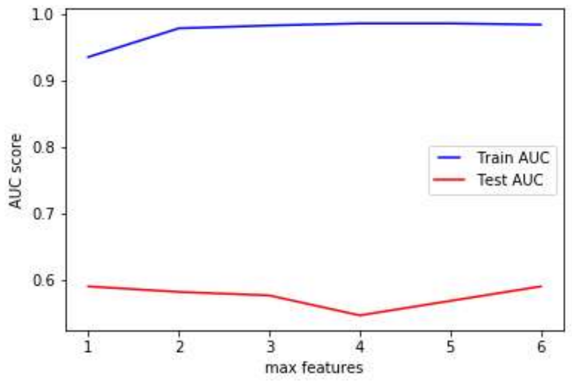
The maximum depth Parameter is an important one to be included in the model. This indicates how deep the built tree can be. The deeper the tree, the more splits it has and it captures more information about how the data. We fit a decision tree with depths ranging from 1 to 32 and plot the training and test errors. We see that our model overfits for large depth values. The tree perfectly predicts all of the train data, however, it fails to generalize the findings for new data. The optimal value is around 10.



The minimum sample split represents the minimum number of samples required to split an internal node. This can vary between considering at least one sample at each node to considering all of the samples at each node. When we increase this parameter, the tree becomes more constrained as it has to consider more samples at each node. Here we will vary the parameter from 10% to 100% of the samples. We can clearly see that when we require all of the samples at each node, the model cannot learn enough about the data. This is an underfitting case.



Minimum samples leaf parameter is the minimum number of samples required to be at a leaf node. This similar to minimum samples splits parameter, however, this describe the minimum number of samples of samples at the leafs. Same conclusion as to previous parameter. Increasing this value can cause underfitting.



The maximum features represents the number of features to consider when looking for the best split. Increasing max features to consider all of the features results in an ideal case for this model. Using max features = 6 seems to get us the optimal performance.

Once exploring the hyper parameter of gradient boosting classifier are completed, we use the best tuned values to build the model again and this resulted in an improved accuracy of **67%**. Among all the models Gradient boosting classifier with hyper tuned parameters gave us the best result.

RESULTS:

The Following table gives the accuracy results of all the model that are implemented for this project.

CLASSIFIER	Without Hyper parameter tuning	With Hyper parameter tuning
Decision tree	52	57
Random forest	57	64
Gradient Boosting	64	67

CONCLUSIONS:

Generally, cricket matches are very hard to predict, as right from the toss and the weather condition, a lot of uncertainties are involved. One can easily predict the winner of the match at the beginning with a 50% accuracy as, either of the team has the equal chances of winning. We attempted to observe the performance of machine learning models on predicting match winners, and we have achieved an accuracy score of around 67% which is better than chance accuracy of 50%. This has increased the chance of us predicting the match outcome by 34%. The Gradient Boosting Classifier gave the highest accuracy in predicting the match outcome for the dataset we choose. Here we were able to get an accuracy of 67% with a dataset of only 670 samples. As the years goes by and more matches are played and added to the dataset, it can eventually lead to increase in the accuracy of prediction. The Twenty20 (T20) format of cricket carries a lot of complexity and randomness, because a single over can completely change the ongoing pace of the game. Indian Premier League is still at infantry stage, it is just above a decade old league and has way less number of matches compared to test and one-day international (ODI) formats. Hence, designing a machine learning model for predicting the match outcome of a Twenty20 format, Indian Premier League with an accuracy of 67% and F1 score of 0.65 is highly satisfactory at this stage of our ongoing work.

FUTURE WORK & SCOPE:

The ML models we used here were one of the basic predicting models. Advanced models like Naive Bayes Prediction algorithm or Recurrent Neural Network model can be used and the results can be compared. More features like pitch type, balance of the team and time of the day the match is played can be added to the dataset and the models can be implemented. The results of this dataset can be compared to the results of our project.

REFERENCES

1.. Shilpi Agrawal, Suraj Pal Singh, Jayash Kumar Sharma “Predicting Results of Indian Premier League T-20 Matches using Machine Learning” 18959646 IEEE, Nov 2018.

2. Rabindra Lamsal, Ayesha Choudhary “Predicting Outcome of Indian Premier League (IPL) Matches Using Machine Learning“ June 2019.

3. A L Samuel, “Some studies in machine learning using the game of checkers. iirecent progress,” in Computer Games I,   pp. 366–400, Springer, 1988.

4. International Centre for Sport Security (ICSS) https://www.gulf-times.com/story/392189/140-billion-lost-in-sports-betting-ICSS-report

5. Mohtadi Ben Fraj, “Parameter tuning for Gradient Boosting” 3363992e9bae Medium, Dec 2017.