Data Analysis of the Indian Premier League

1st Nandagopal Vidhu
Centre for Machine Intelligence and Data Sciences
IIT Bombay
Mumbai, India
Email: nandagopal@iitb.ac.in

2nd Prayas Jain
Centre for Machine Intelligence and Data Sciences
IIT Bombay
Mumbai, India
Email: 180100088@iitb.ac.in

***Abstract*—Cricket comes second on the most popular sports list after football. The Indian Premier League (IPL), which started in 2008, is the biggest domestic league in the whole world. There is a lot of scope of data analytics in the game of cricket as it involves a lot of data and statistics. Many parameters can be used to predict a match's outcome or the tournament's outcome also. Machine Learning and Data Science can be used in combination with these statistics to predict the outcome of a match. Players and teams are rated according to some insightful plots in addition to analyzing the most popular cities and most preferable toss decisions. A few important features such as the current scorecard, on-crease batsmen's scores, and remaining overs are used to predict an inning's final score. Machine learning algorithms like Random Forest, XG Boost, Extra Trees are used in addition to Deep learning neural networks. Results, based on which algorithm performs the best, are plotted along with learning curves.**

***Index Terms*—Sport, Cricket, Indian Premier League, Machine Learning**

I. Introduction

Sports analytics is a field that is becoming widely popular due to the competitive edge that it can give both to sports teams as well as stakeholders involved in the sport. Various data which is available such as the players and team statistics, environment conditions, etc is made use of to predictive models which can help stakeholders make informed decisions on the game. The main objective is to improve the performance of the team and assist in creating strategies which would help the team perfectly counter its opponents. This can be done both prior to a game as well as dynamically as the game progresses. In recent times, it has been observed that the audience themselves are also interested in the data analysis that goes on in the game and hence, sports analysts try to present this data to the audience by making simplifications to it and making use of pictorial elements such as graphs and charts to capture their attention.

A. About Cricket

Cricket is a sport that is played by two teams, each having eleven members. A team consists of batsmen, bowlers, and all-rounders. The role of the batsmen is to score as many runs as possible in the limited time/overs available, while the bowlers try to restrict the score that the batsmen try to make. All-rounders are players that play both roles and have sufficient expertise in both batting and bowling. The performance of a team depends on various factors such as the constitution of the team in terms of types of players, the venue in which the match is being held, the environmental conditions, and the type of opponents that they're playing against. Data analytics can be made use of to help the teams management figure out which players to play in a specific match, the odds of them reaching a specific stage in a tournament, the environmental conditions that they're going to play in, etc. It can also be used during a match to help the team adjust their strategy according the state at which the match is in, to provide them a competitive edge against their opponent. These days, data science techniques are being made use of by every team that competes in the sport professionally. When used correctly, it can help teams bridge the gap in skill by formulating an effective strategy to counter their opponents.

B. About the Indian Premier League

The Indian Premier League (IPL) is the worlds biggest domestic cricket tournament. It is a 20-over format of the game that makes for short, fast-paced games which is one of the reasons for its massive fanbase. It is an annual tournament and has seen 13 such tournaments conducted so far. There are 8 teams involved in the tournament and the teams themselves consist of players from all around the world. The tournament generates a large revenue and has many stakeholders heavily invested in it. So teams will do everything they can to get an edge over their opponents in a game. Data Analysis is now heavily used by all teams to try and gain this edge.

C. Scope and Overview

Section II talks about the Literature Review of the papers and resources referred. Further, in Section III, the datasets used for performing the analysis. Section IV talks about the Analysis Pipeline followed. This paper aims to create a forecasting model for teams to use during the match. Based on the scores data of a team and players at any stage, it tries to predict the final score of the team. The seasons under consideration are the 2008-2020 seasons. Due to the pandemic, the 2020 season was held in the UAE instead of India and provided a considerable challenge for the analysis as the data previously available was for Indian playing conditions which are considerably different from that of the UAE. In addition to this, the teams have changed considerably in their constitution as compared to the past seasons. Section V showcases the results obtained by the predictive models made. Section VI

discusses the results obtained and Section VII concludes the paper along with further scope of research.

II. Literature Survey

Quite a bit of research goes on in the field of data science in sports, and cricket being the second most popular sport in the world, is no exception to this.

Barot et al. [1] made measures of the performances of individual players in a team and used this along with the playing conditions to predict the winner of a match with good accuracy. Kalpdrum Passi and Niravkumar Pandey [2] presented a detailed analysis of the performance of various Machine Learning (ML) frameworks to make predictions of how many runs a particular batsman will score in a match and how many wickets a particular bowler will take. Rabindra Lamsal and Ayesha Choudhary [3] formulated a multi-variate regression based model to calculate the points earned by each player in a team based on their past performances and the points awarded to each player was used to compute the relative strength of each team. This data was then used to predict the winner of a match immediately after the toss took place. C. Deep Prakash, C. Patvardhan, C. Vasantha [4] were one of the first to use ML in cricket to make predictor models for predicting the outcome of a match. Priyanka S, Vysali K, Dr K B PriyaIyer [5] analysed the results of IPL matches in the duration 2008-2019 and applied data mining algorithms on this data to predict the outcome of the 2020 edition of the IPL.

D. Thenmozhi et al. [6] made a dynamic forecasting model which predicts the match winner of IPL matches at various phases of the game using ML algorithms and compared their model across the eight teams to evaluate its performance.

III. Datasets

The datasets used for analysis and prediction were collected from www.kaggle.com [7], where the data of all editions of the IPL so far was available. Two datasets have been used. One for overall matches data and one for ball-to-ball data for the full 2008-2020 period. Both the datasets are linked by the 'id' column which represents the matches uniquely. Some of the useful features present in the dataset are date of match, venue, run(s) and wicket(if any) on every ball, toss decision, batsman and bowler, result of match with margin etc. There are some minor discrepancies in data such as missing values in 'bowling team' column and duplicate team name but it doesn't hurt the predictions task as team data is also present in 'team1','team2' columns. The dataset consists of 2 lakh data points with 18 features in total.

Data was also collected from www.cricsheet.org [8], which is a relatively small dataset with 76k data points but has useful features like current score, wickets, overs, striker and non-striker runs. These features are crucial to predict the final score of innings and hence is used for this particular prediction task in the work.

IV. Analysis Pipeline

As observed from the literature survey conducted, a large majority of the predictive models that were made are used to predict the outcome of the match and this prediction is made before the start of the match. This prediction will be useful for the team to make long-term decisions for the team to perform better in the tournament as a whole but is not very useful during the match itself as no changes can be made to the team in the middle of a match. The work discussed in this paper seeks to fill in this gap by providing data to the team at various phases of the match so that the team can make informed decisions such as what batting order and bowling order to use for the rest of the game.

Firstly, an exploratory analysis of the data is conducted to get a better understanding of what parameters affect the performance of the team as a whole as well as the individual contributions of the players.

A. Interesting Insights drawn from the datasets

Various manipulations are performed on the available datasets to extract some insightful information from them.

1) *Total runs scored over the years:* As seen in Fig. 1, total runs scored in a season dipped from first to second season, might be due to different playing conditions in South Africa where the second season was played. Thereon, due to more teams being added on and hence more matches in a tournament, total runs increased from tournament to tournament and dips again in 2014, in the year when teams were again reduced from 9 to 8. Afterwards, total runs scored are more or less constant.

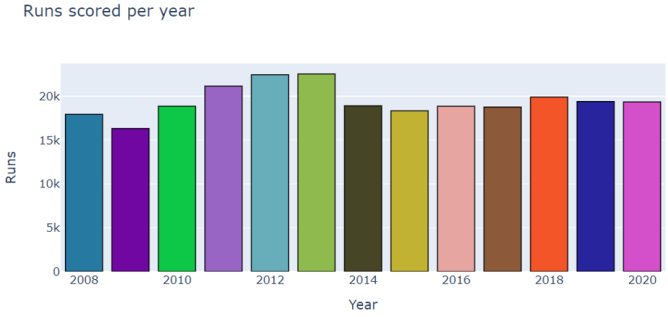

Fig. 1. Total runs scored in every year of IPL

2) *Toss Decisions:* In Fig. 2, we can see clearly that most teams prefer to chase targets from the analysis. About 61% teams chose to bowl after winning the toss.
3) *Toss influence:* As we can see in Fig. 3, despite teams favoring to bowl first, results tend to not depend on toss decisions. Of course, winning a game depends on the team's quality and ability to chase too hence we can't conclude that teams can choose arbitrary whether to bat/bowl first, but this gives an interesting insight on toss' influence on winning chances.
4) *Effect of powerplay/death overs:* Fig. 4 shows runs scored and wickets taken in every over on an average.

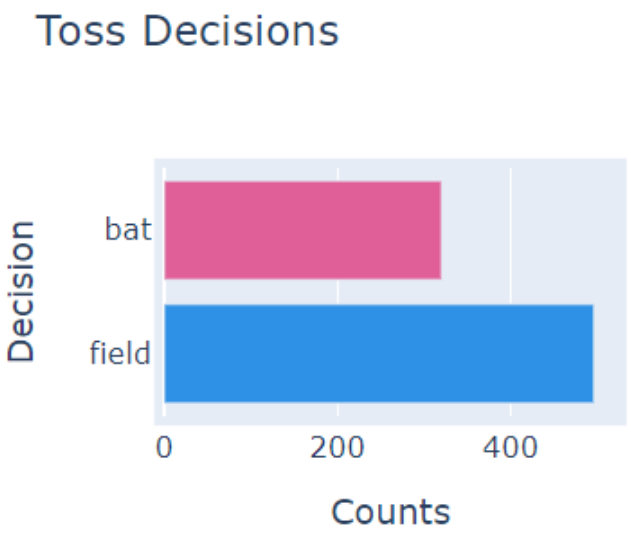

Fig. 2. Toss decisions taken by teams

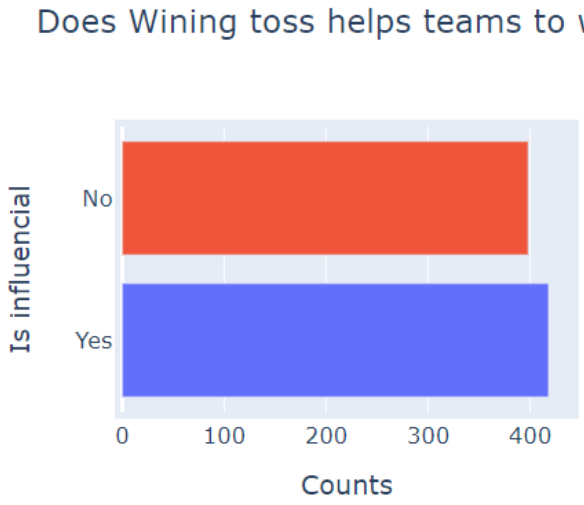

Fig. 3. Influence of toss on winning matches

As expected, runs scored are quite high relatively in powerplay overs when field restrictions are in effect and also in death overs when batting teams tend to risk to get maximum runs on board without worry of getting out. This is also reflected in average wickets taken as the number of wickets taken also increase in death overs by the same logic.

5) *Total runs scored in a match:* Fig. 5 is a frequency-plot (Histogram) of total runs scored in a match. We can see on average around 320 runs are scored in a match with maximum going as high as 450.
6) *Most successful teams:* In Fig. 6, we can see most wins by all teams ever competed in IPL. As being the most crowned team of IPL, Mumbai Indians has the most number of wins as expected. What is interesting to note

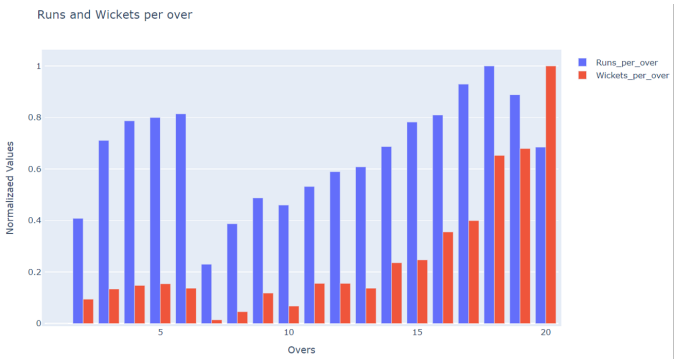

Fig. 4. Average runs scored and wickets taken in an over

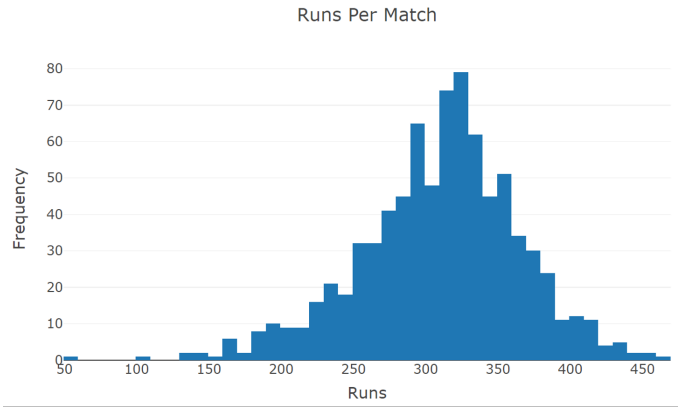

Fig. 5. Histogram of total runs scored in a match

is that despite not competing in 2 full tournaments (due to ban), CSK are just 14 wins away from MI in terms of total wins, a proof why MI-CSK rivalry is most popular in IPL.

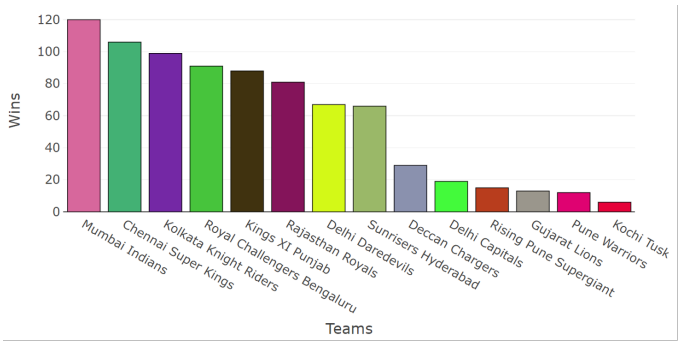

Fig. 6. Total wins by all teams till 2020

7) *Most Influential Players:* Using Man of the Match (MoM) data available in the dataset, we can extract players with most MoM awards in IPL. Fig. 7 shows us that despite RCB not winning no IPL yet, three of it's players feature in top 10 list with AB de Villiers and CH Gayle at number 1 and 2 respectively. This highlights the incompetence of RCB's bowling which hampers their title-winning chances.

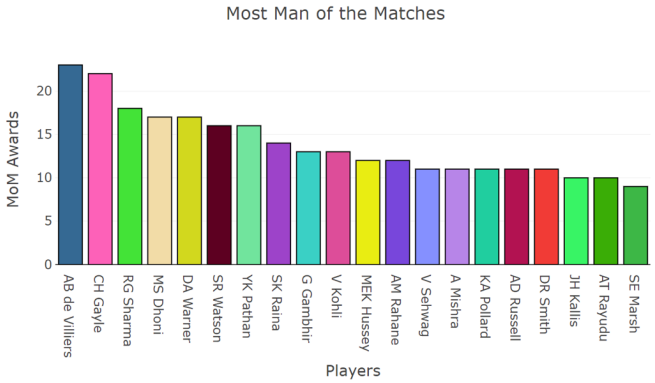

Fig. 7. Players with most Man of the Match awards till 2020

8) *Top hosting cities:* In Fig. 8, we can see Mumbai as the top city to host IPL matches, with about 16% of matches being held there. Despite only having one tournament there, one city of UAE also shows up in the top 10 list due to limited number of grounds available in the country.

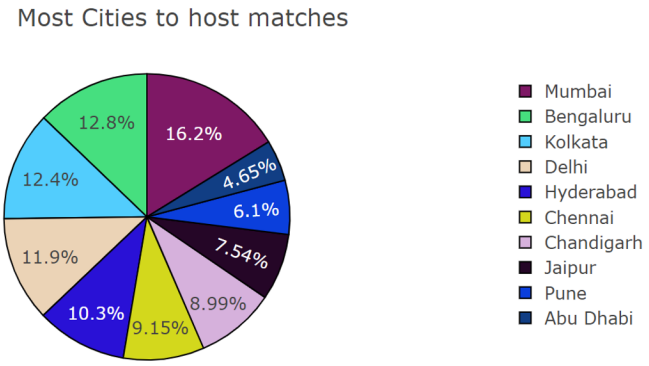

Fig. 8. Top 10 cities to host IPL matches

V. Results

In this section, the paper presents a method to forecast the final score of the team which is presently batting, based on the current score of the team. The 2020 season games are being considered for forecasting and the data available of IPL matches played before that is taken into account as well for prediction.

The exact features which are used for prediction are listed as follows:

- Current runs
- Current wickets
- Overs completed
- Striker's runs
- Non-striker's runs

Current runs and wickets are most influential in determining what will be the final score of the inning as more wickets fallen would mean the team won't be able to muster up more runs and more runs at any position would automatically mean higher score possibility due to runs being added cumulatively to the final score. Current score alone won't be of much help if we don't know current overs as combining these two the network can know current net run rate. Striker and non-striker's current score would also be helpful as set batsmen would be crucial to elevate the teams' score.

A. Prediction

1) *Neural Networks (NNs):* Neural networks are the modern times way-to-go prediction models. Neural networks are based on how the human nervous system works, neural units are basic processing junctions of a neural network. Each unit receives an input, applies an activation function to it and sends processed output to next layers' units. Several such layers of units are stacked together and the neural network learns intricate details of data itself such as to achieve satisfactory results on target labels.
Fig. 9 shows the architecture of the NN used for prediction. Batchnorm is applied after each linear operation before applying leaky relu as activation function. Batchnorm is used to prevent exploding of units' activations and consequently the gradients. Leaky relu prevents 'dead neurons' as it always has a slope for gradient computation. Adam optimizer is used for backpropagation and regularization purposes. Mini-batches of 128 size are used for mini-batch gradient descent. The training is performed on Google Colab with GPU runtime and the model is built in Pytorch, a deep learning framework. The total number of learnable parameters in the model are 100k and it is run for 200 epochs with 0.001 as the learning rate. These hyperparameters for learning are chosen after running many experiments.

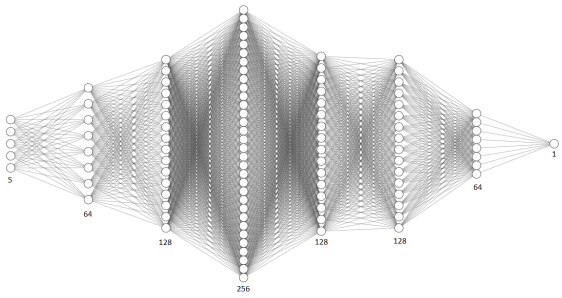

Fig. 9. Neural network architecture with hidden layer sizes

Metrics used: Mean squared error is used as a loss function for backpropagation. R^2 score and custom accuracy (predicted score being in margin of 10 of actual final score) is used for evaluating the model.
Results: After 200 epochs, the results obtained are as follows: Train Loss: 0.0030 — Val Loss: 0.0031 — Train R2: 0.5263 — Val R^2: 0.4813 — Val Acc: 59.8028%. So, the accuracy is around 60% which is not usable for practical purposes. Limited amount of data is the primary reason why neural networks are not giving

decent results even after training for around 1 hour. But, this is not entirely useless as can be seen in the training graphs as shown in Fig. 101112.

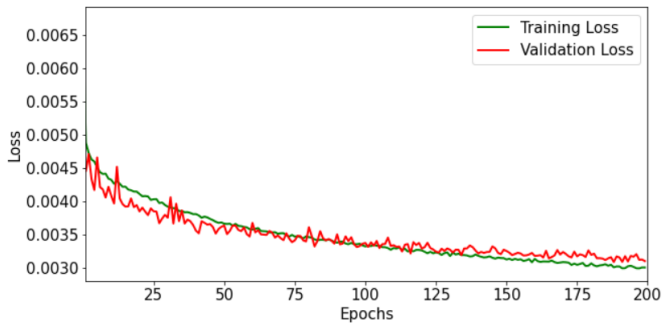

Fig. 10. Training and Validation losses are decreasing with epochs with stabilizing in end due to limited data

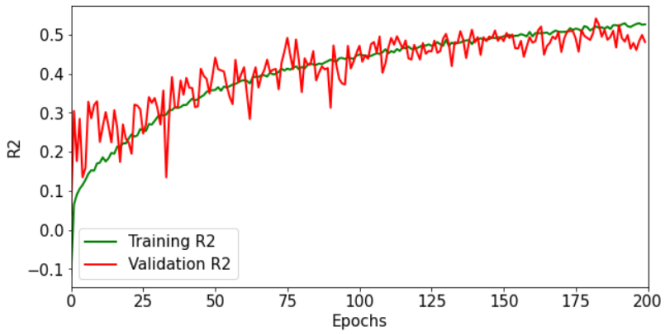

Fig. 11. R^2 score is also increasing for both training and validation data

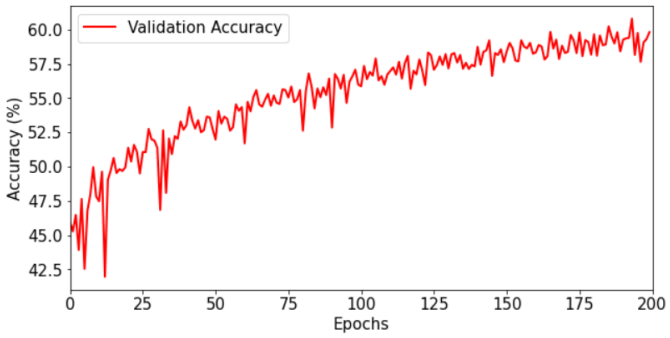

Fig. 12. Validation accuracy as a function of epochs

2) *Gradient Boosting Regressor:* 'Boosting' refers to one by one adding weak learner sub-models or more specifically decision trees (weak learner denoting a learner performing slightly better than chance). Gradient descent is applied after calculating loss and the next tree is added such as to take maximum descent towards optimum value (the gradient direction).
Hyperparameter Tuning: There are several parameters that can be tuned for the gradient boosting regressor, the most crucial one being the number of estimators or trees, also learning rate and maximum depth of a tree are also crucial tunable hyperparameters. Maximum features to take is also another parameter which is set to 'full' as we already have only 5 features for our regression. After performing hyperparameter tuning on the dataset with 5-fold cross-validation strategy, we found increasing the number of estimators continuously increases the accuracy due to limited data available . Best learning rate and maximum depth of trees are found to be 0.5 and 14 respectively.
Results: Choosing the number of estimators to be 5000, learning rate as 0.5 and maximum depth as 14, we get the accuracy as 64% and R^2 score as 0.6 on testing data.

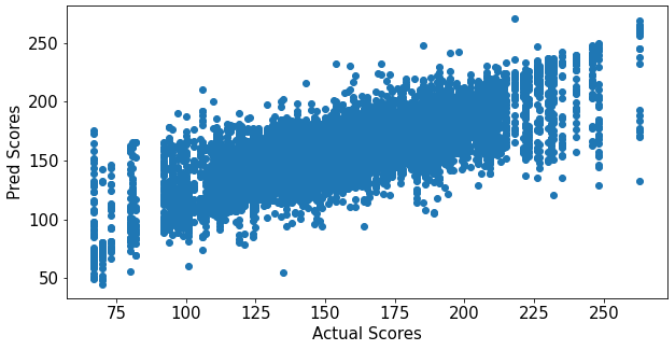

Fig. 13. For XGBoost, comparison of actual scores and predicted scores

3) *Random Forest Regressor:* The tree growing in Random Forests happens in parallel which is a key difference between AdaBoost and Random Forests. Random Forests achieve a reduction in overfitting by combining many weak learners that underfit because they only utilize a subset of all training samples.
Hyperparameter Tuning: Number of estimators or trees and maximum depth of a tree are the hyperparameters chosen to tune. Maximum features are again set as 'full' as before. Hyperparameter tuning is performed on the dataset with 5-fold cross-validation strategy. Best number of estimators and maximum depth of trees are found to be 300 and 50 respectively.
Results: Choosing the number of estimators to be 5000 and maximum depth as 14, we get the accuracy as 65.52% and R^2 score as 0.68 on testing data.
4) *Extra Trees Regressor:* Random forest uses bootstrap replicas, that is to say, it subsamples the input data with replacement, whereas Extra Trees use the whole original sample. This reduces bias. Another difference is the selection of cut points in order to split nodes. Random Forest chooses the optimum split while Extra Trees chooses it randomly. This reduces variance.
Hyperparameter Tuning: Number of estimators or trees and maximum depth of a tree are the hyperparameters chosen to tune. Maximum features are again set as 'full' as before. Hyperparameter tuning is performed on the dataset with 5-fold cross-validation strategy. Best

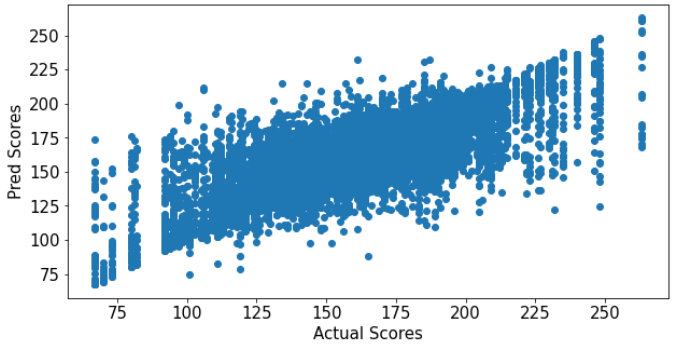

Fig. 14. For RF Regressor, comparison of actual scores and predicted scores

number of estimators and maximum depth of trees are found to be 700 and 60 respectively.

*Results:*Choosing the number of estimators to be 5000 and maximum depth as 14, we get the accuracy as 68.25% and R^2 score as 0.67 on testing data.

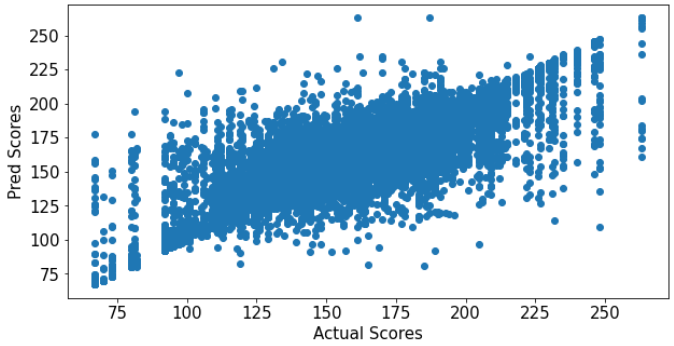

Fig. 15. For Extra Trees Regressor, comparison of actual scores and predicted scores

VI. Discussion

As seen in Fig. 16, the Random Trees Regressor and the Extra Trees Regressor offer the best performance and are the best ML frameworks to use for predicting the final score of the batting team.

Algorithm	R2	Accuracy (%)
Neural Networks	0.48	59.80
XG Boost Regressor	0.60	64.36
Random Forest Regressor	0.68	65.52
Extra Trees Regressor	0.67	68.25

Fig. 16. Summary of results obtained from different models

VII. Conclusion and Future Scope

This paper provides useful insights from IPL dataset about what are the best performing teams and players. Toss decisions and their importance in winning matches prove the overall winning toss has more or less no influence on winning chances. Best performing players of IPL can be listed with the most MoM awards analysis. Sponsors can focus on which cities host the IPL matches most to analyze the audience in those areas specifically and make their plans accordingly. The prediction of final score at any given moment of match is currently done with the help of Current Run Rate(CRR), while it is one of the useful features, it doesn't take into account what are the remaining overs and scores of the batsmen at crease. The models proposed in the work take these features into account to predict the final score given these features at any point in the game. Due to limited data, the best model is 70% accurate on an error margin of ±10 runs. Future Work can be pre-training the neural network models on an ODI or T20 international datasets and then fine tuning them for ipl predictions as direct training with datasets is not possible due to different formats and playing conditions.

Acknowledgment

We would like to express our very great appreciation to Prof. Amit Sethi, Prof. Manjesh Hanawal, Prof. Sunita Sarawagi, and Prof. S. Sudarshan for this opportunity to explore the application of data science to cricket.

References

[1] H. Barot, A. Kothari, P. Bide, B. Ahir and R. Kankaria, "Analysis and Prediction for the Indian Premier League," 2020 International Conference for Emerging Technology (INCET), Belgaum, India, 2020, pp. 1-7, doi: 10.1109/INCET49848.2020.9153972.

[2] Passi, Kalpdrum & Pandey, Niravkumar. (2018). Increased Prediction Accuracy in the Game of Cricket Using Machine Learning. International Journal of Data Mining & Knowledge Management Process. 8. 19-36. 10.5121/ijdkp.2018.8203.

[3] Lamsal, Rabindra & Choudhary, Ayesha. (2018). Predicting Outcome of Indian Premier League (IPL) Matches Using Machine Learning.

[4] Deep Prakash, Chellapilla & Patvardhan, C. & Vasantha, C.. (2016). Data Analytics based Deep Mayo Predictor for IPL-9. International Journal of Computer Applications. 152. 6-11. 10.5120/ijca2016911875.

[5] Priyanka, Sachi. (2020). Prediction of Indian Premier League-IPL 2020 using Data Mining Algorithms. International Journal for Research in Applied Science and Engineering Technology. 8. 790-795. 10.22214/ijraset.2020.2121.

[6] Thenmozhi, D. & Palaniappan, Mirualini & Sakthi, S.M.Jai & Vasudevan, Srivatsan & Kannan, V & Sadiq, S. (2019). MoneyBall - Data Mining on Cricket Dataset. 1-5. 10.1109/ICCIDS.2019.8862065.

[7] @miscWinNT, author = Prateek Bhardwaj, title = IPL Complete Dataset (2008-2020), year = 2020, url = https://www.kaggle.com/patrickb1912/ipl-complete-dataset-20082020, urldate = 2020-11-23

[8] Cricsheet, url = https://cricsheet.org/downloads/, urldate = 2020-11-30

[9] Exploratory Data Analysis of IPL Matches-Part I, url = https://towardsdatascience.com/exploratory-data-analysis-of-ipl-matches-part-1-c3555b15edbb, urldate = 2019-10-16

[10] Predictive Analysis of an IPL Match, url = https://towardsdatascience.com/predicting-ipl-match-winner-fc9e89f583ce, urldate = 2020-03-06