Sports Analytics : Indian Premier League

Prajwal Kalpande
Winter In Data Science
Analytics Club,IITB
Mumbai, India
prajwalkalpande3@gmail.com

***Abstract*—Due to recent advances in data collection and management technology the scope of Sports Analytics has broadened significantly. Team owners, sponsors and managements make use of Machine Learning to get a competitive edge over the opponents. The Indian Premier League, which came in 2008 took T20 cricket to a whole new level. It is highly commercialized and predictive models are being used to make it more competitive and entertaining. In this paper, detailed data analysis of IPL from 2008-2020 has been done. Based on the insights obtained, different ML models have been discussed for score and winner prediction.**

***Index Terms*—Cricket; Indian Premier League; Machine Learning; Neural Networks; Data Science; Sports Analytics; Score Prediction; Winner Prediction**

I. Introduction

The practice of sports analytics has been around for decades, but recent advances in data collection and management technology have broadened its scope significantly. The use of analytics in sporting events helps various stakeholders, including sportsperson, associations, and fans, to gain in-depth insights on live in-game activity and past game events. Some of the main objectives include improving the overall team performance, maximizing winning chances and help rightsholders take decisions that would lead to higher growth and increased profitability. Apart from real-time data analytics which helps teams devise strategies during the game, predictive models are harnessed to determine the possible match outcomes that require significant number crunching and data science know-how, visualization tools and capability to include newer observations in the analysis.

With the rise of web-based betting industry and fantasy leagues in cricket in recent times, the demand for this field has risen very high. Another major factor boosting this sector is how streaming platforms and channels are making use of data analysis to make the audience involved in watching matches by showing stats, comparisons between teams and players and asking them to answer questions during live matches.Thus, commercialization has been a major driving factor in sports analytics.

A. About Cricket

Cricket is the most well-known game in a few nations around the globe. It might not have a similar prominence as football, however, this game is worshiped by a large number of fans. It is a bat-and-ball game played between two teams of eleven players each. Each team consists of bowlers,batsmen, all-rounders and one wicket-keeper. The goal of the batsman is to score as much runs as possible while bowlers try to restrict their score. All-rounders are players which are good at both bowling and batting and are considered a very important ingredient of a good team. The performance of a team depends on various factors such as the composition of the team, the opposition team, the venue where the match is being held, the environmental conditions during the gameplay, etc. The use of data analytics by sports management for improving the teams chances of winning and optimizing player performances come under sports analytics. It helps teams in deciding which players should they include and what strategy should they devise during the game, giving them a competitive edge against their opponent.

B. Indian Premier League(IPL)

The Indian Premier League (IPL) is an Indian professional Twenty20 cricket league which came in 2008 and completely revolutionized the cricket world. The craze of IPL has increased beyond bounds and it is now the the no. 1 cricket league in the world. It is more commercialized, with more entertainment, and more cricketer brand value. The Brand value of IPL is estimated at about $4.16 billion. Data collection and Data Analysis in IPL has breached the next level, because as IPL is spending lot of money on players, it has become necessary for IPL teams to find out which players to buy and how to utilize them to help them win the IPL trophy.

C. Scope and Overview

Section II includes the datasets used for analysis and prediction. These datasets were modified for better predictive power and the modified versions can be found on the Github Link. Section III summarizes the insights gained after performing Exploratory Data Analysis (EDA) on the datasets. In Section IV the results obtained after training various ML models have been discussed along with a brief overview of the models. Section V lists down the results of all the algorithms used. Finally, Section VI concludes the paper and discusses the future work which can be done. The main aim of the project was to develop forecasting model for teams to use during the match for score prediction as well as winner prediction. Based on the data available at any stage, predictions have been made by carefully selecting useful features. The seasons taken under consideration are 2008-2020 .

Since IPL 2020 season was held in the UAE instead of

India, it was another great challenge for the data analysts since most of the data is for matches played in India. The 2020 Season saw many changes inside teams and their strategies. the 2020 season was held in the UAE instead of India and provided a considerable challenge for the data analysts and team managements. The IPL 2022 season coming next year is going to be the biggest in history with it being a Mega Auction. In this season, two new teams will be joining and since all teams have been allowed to retain atmost four players, there is going to be cutthroat competition among the teams to buy players. Data analysis will be a key factor this year which will aid the teams in deciding which players to buy and at what price.

II. Datasets

The datasets used for analysis and prediction in this project were obtained from Kaggle [1], which included the entire data of IPL 2008-2020. Two datasets were used namely, deliveries (*IPL Ball-by-Ball 2008-2020.csv*) and matches(*IPL Matches 2008-2020.csv*). Both the datasets are linked by the ‘id’column(unique Match ID as per ESPNCricinfo) which represents the matches uniquely. Some of the useful features present in the dataset are year, venue, run(s) and wicket(if any) on every ball, toss decision, batsman and bowler, result of match with margin etc.

The deliveries dataset consists of 193k datapoints with a total of 18 features and the matches dataset contains data of 816 matches with 17 columns. There are some minor discrepancies in the data such as missing city name, result of the match missing and misspelled team names. But since they are very less in no. compared to the size of the data, it is not a big issue. Pre-processing was done on the data followed by feature engineering for better prediction results.

III. Exploratory Data Analysis

The main focus of the project was on building good predictive models to predict the projected score of the innings and the winner of the match. Since, IPL is a very competitive league this is a very difficult task. So, for getting an understanding of which features to use for predictions EDA was done. Comprehensive and in-depth analysis of the data available was done to gain useful insights about teams and players so as to extract important features and discover interesting facts.

A. Interesting Insights Drawn from the Datasets

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

1) Maximum number of wins in a season: As can be seen from Fig. 1, Mumbai Indians is the most consistent team, ending 5 seasons with maximum no. of wins in that season followed by CSK which had maximum no. of wins in 3 seasons. Rest of the teams have maximum no. of wins in atmost 1 season, showing the supremacy of MI and CSK in IPL

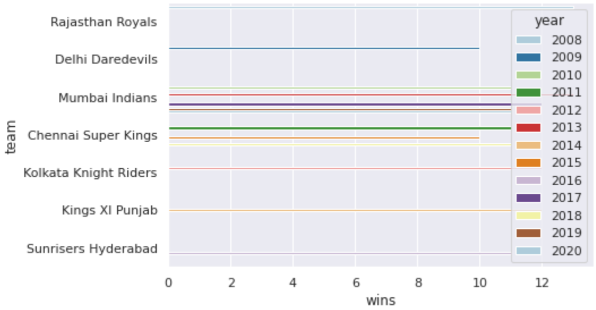

Fig. 1. Maximum number of wins per season

2) Toss Decisions : In Fig. 2, we can clearly see that most of the teams prefer to chase targets in IPL. About 5 out of 8 teams i.e. almost 62% teams chose to bowl after winning the toss instead of batting first.

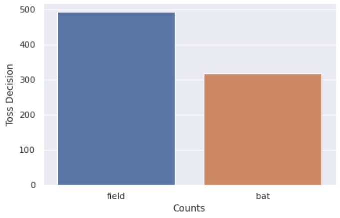

Fig. 2. Toss decisions taken by teams

3) Toss Influence: Fig. 3, shows the relation between winning the toss and winning the match. ‘Yes’denotes no. of matches in which the team who won the toss won the match as well and ‘No’shows the matches where team winning the toss had to face defeat. It is clearly visible that both have categories almost equal counts and hence it considered alone, the feature that which team won the toss is not so helpful in predicting the winner. However, when combined with other factors like venue, teams and playing conditions it may prove to be quite useful.

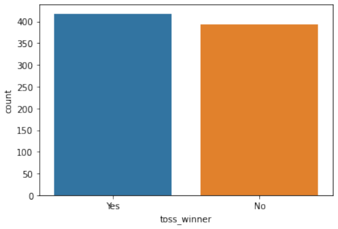

Fig. 3. Toss Influence on Match Result

4) Rivalry between Teams: Fig. 4 highlights the most famous rivalry of CSK and MI, in which both teams have performed equally well against each other. The other bar plot shows the rivalry between SRH and KKR in which KKR seems to have performed better.

5) Top 10 Greatest Victories: Fig. 5 depicts the largest win margins in the history of IPL. We can observe that Royal

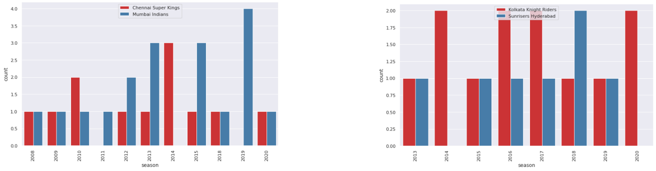

Fig. 4. Team Rivalries

Challengers Bangalore (RCB) appears 6 times in this list, out of which it has won 3 times. This shows that when the key players of RCB perform, RCB wins with ease. However, when they don't RCB performs poorly. Thus, we see that mere one-man shows aren't good enough to win big tournaments like IPL.

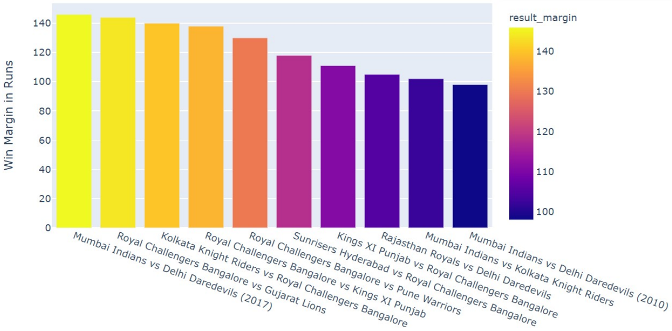

Fig. 5. Largest result margins in runs

6) Win Percentage : MI has the highest win percentage (nearly 60%) followed by CSK and SRH. Deccan Chargers has the least win percentage. Deccan Chargers has the least win percentage. As seen in Fig. 2 the win percentages of various teams are nearly equal, indicating that each team is equally probable to win i.e. it is hard to predict which team will win due to strong competition.

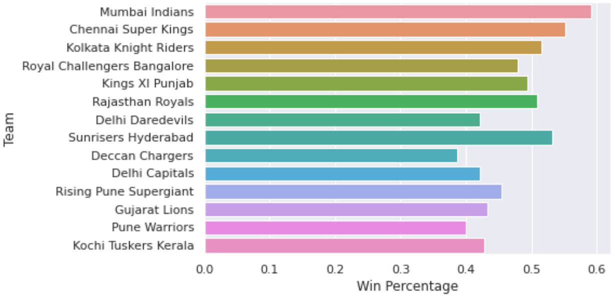

Fig. 6. Win Percentage of teams in IPL

7) Most Influential Player: Using Man of the Match (MoM) data available in the dataset, we can extract players with most MoM awards in IPL. Fig. 7 depicts the match winners who have won max MoM awards. If we look closely, we can see three RCB batmen in the top 10 list. Despite having good batsmen RCB hasn't been yet able to win any IPL season, indicating the fact that their bowling sector needs to improve to get better results.

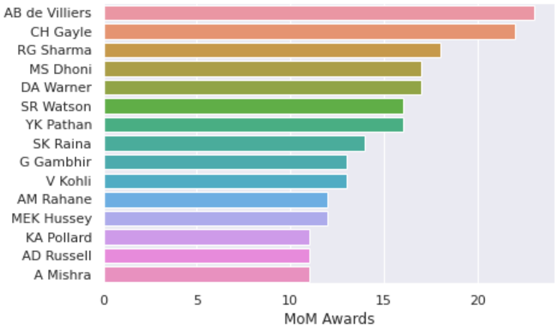

Fig. 7. Players who have won the most Man of the Match awards

8) Comparison between batsmen Stats: With the Mega Auction coming in 2022, teams will be focusing on choosing players based on skills with the intention of maximizing the team performance. Representing the stats of players as shown in Fig. 8 can help Team owners and managements to compare between different options available and take crucial decisions regarding team selection with ease.

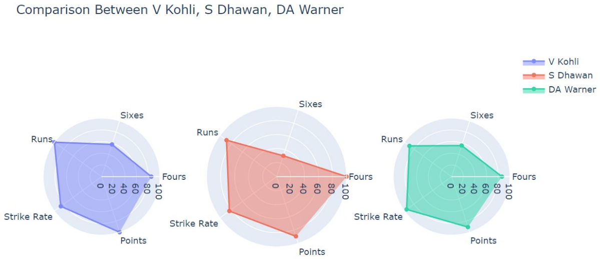

Fig. 8. Comparison between players

An interesting observation is that although Shikhar Dhawan deals more in fours than sixes, he still has a better strike rate and more runs than Virat Kohli and David Warner. This makes him a great choice while selecting a team, since not only does he score more but the chances of him getting out are lesser.

9) Top hosting cities: In Fig. 9, we can see Mumbai as the top city to host IPL matches, with about 17% of matches being held there. Despite only having one tournament there, Abu Dhabi in UAE also shows up in the top 10 list due to the limited number of grounds available in the country.

IV. Results

In this section, I have summarized the results obtained after using various Machine Learning models for final score prediction and winner prediction. In some sub-sections the *WorkFlow* has been discussed where the process of arriving at the model and the various problems faced have been discussed. For forecasting purposes, IPL 2020 data was considered, and the rest of the data (2008-2019) was used for training purposes.

Feature engineering was done to generate better features using existing ones. Particularly, current runs, current wickets,

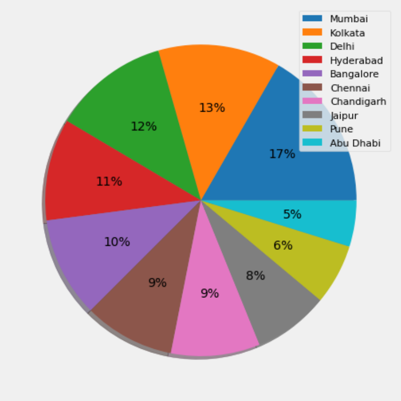

Fig. 9. Cities hosting IPL Matches

runs in last 5 overs and wickets in last 5 overs were engineered using the available data for better predictive power. As a result, the overs in the modified dataset, now began from 5 and went up to 20. This lead to reduction in the size of data used fro training.

A. Score Prediction

Custom accuracy was defined to consider the predicted score as correct if |Predicted score - Actual score| $\leq$ 10 runs and wherever accuracy is mentioned ahead refers to this custom accuracy unless mentioned otherwise.
The features considered for score prediction were as follows:

- Innings
- Over
- Venue
- Batsman
- Bowler
- Current Runs
- Team 1
- Team 2
- Team1_toss_win
- Team1_bat
- Runs in Last 5 overs
- Wickets in Last 5 overs

Team1_toss_win is 1 when Team 1 wins the toss and otherwise 0. Team1_bat is 1 if Team 1 is batting first and otherwise 0.

1) Linear Regression: Linear regression is a linear approach for modelling the relationship between the target response and one or more features which are being used for prediction. Ordinary Least Squares(OLS) Linear Regression was used.

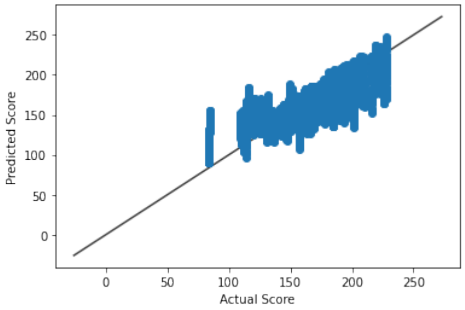

Fig. 10. Actual scores vs. Predicted Scores
*The straight line in the figure is y=x

That is, residual sum of squares between the actual score, and the predicted score was minimized to obtain the line that best fits the data.
Results: Custom accuracy of 47.20% was obtained and the Root Mean Squared Error(RMSE) was 19.38.

2) Lasso Regression: Lasso regression performs L1 regularization, which adds a penalty equal to the absolute value of the magnitude of coefficients to the ordinary least squares loss function. This type of regularization can result in sparse models with few coefficients as coefficients are allowed to be zero unlike L2 regularization.

After running Elastic Net Regression model it was found that for higher value of L1 ratio (almost 1), cross-validation score was about 0.51 as compared to the very low cross-validation score of 0.03 for L1 ratio = 0 i.e. complete L2 regularization. Hence, lasso regression model was trained as it has L1 ratio=1.
Results: The custom accuracy comes out to be 45.76% and RMSE was about 19.45. As expected, Linear Regression performs better than Lasso Regression. These models already have a high bias leading to higher variance and less tendency of over-fitting. Hence, applying regularization is not a good idea.

3) Support Vector Regression(SVR): SVR is a supervised Machine Learning algorithm based on the idea of Support vector machine that is popularly used for classification tasks. In SVR, the straight line that is required to fit the data is referred to as hyperplane.The objective of a support vector machine algorithm is to find a hyperplane in an n-dimensional space that distinctly classifies the data points. The data points on either side of the hyperplane that are closest to the hyperplane called Support Vectors influence the position and orientation of the hyperplane and thus help build the SVM.

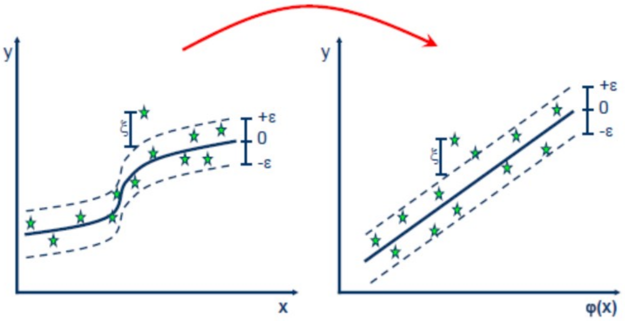

Fig. 11. Basic Idea of how SVR works
*Kernel functions transform the data thereby allowing linear separation

Although SVR allows us to model non-linear relationships unlike Linear Regression and also provides many hyperparameters to adjust, it has few disadvantages. First and the most important is that since it uses non-linear kernels with large training time-complexity (O($n^2_{samples} * n_{features}$). Hence, it takes lot of time to train and this limits us from doing hyperparameter tuning making it unsuitable for large datasets.

Results: RBF kernel was used for regression which gave a custom accuracy of 45.74% and RMSE equal to 19.29

4) Random Forest Regression: Random Forest(RF) Regression is a supervised learning algorithm that uses ensemble learning method for regression. Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model. RF uses multiple decision trees as base learning models and combines them in parallel leading to lower variance and higher accuracy. However, due to higher complexity RF model may overfit if the hyperparameters are not tuned well.

a) WorkFlow: The hyperparameters considered for tuning were n_estimators i.e. no. of trees in the forest and max_depth which is the maximum depth of a tree. Inititally, Grid Search was run on the following param_grid :
{”n_estimators”: [250,500], ”max_depth”: [50,100,250] }

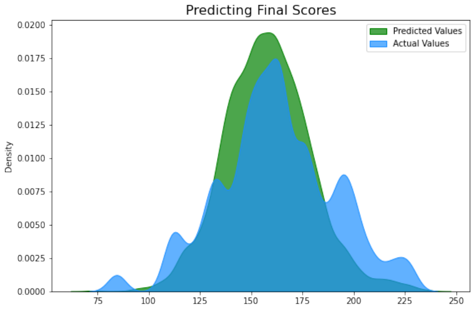

Fig. 12. Predictions made on Test Set
RF model {n_estimators = 500 and max_depth = 100}

The best parameters returned were n_estimators = 500 and max_depth = 100. However, when the model was used for prediction on test set, the accuracy returned was quite low 45.77% and RMSE was also higher than expected (20.90).
On investigating the cause, it was found that the RF model had overfit the training data as can be seen in Fig. 13.

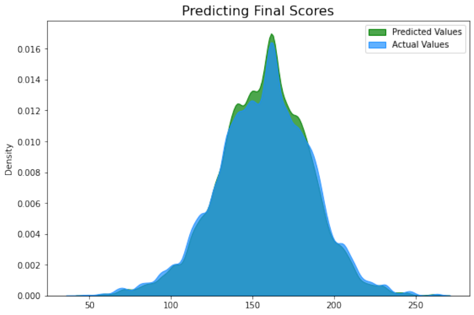

Fig. 13. Predictions made on Training Set
RF model {n_estimators = 500 and max_depth = 100}

Apart from this, there was one more additional factor which was leading to low accuracy. As mentioned earlier, feature engineering lead to reduction of data available, which increased the accuracy in case of Neural Networks and the three models discussed earlier. However, in case of Random Forest Regression better results were obtained when the original data was considered without feature engineering.

Thus, to conclude increasing the data size helped in reducing overfitting to some extent. But even after this the results were not satisfactory. So, I plotted the data distribution of IPL seasons from 2008-2019 and IPL season 2020 as shown in Fig. 14. As can be seen, there is considerable difference in the scores distribution, and if it were similar it would increase the accuracy. So, I removed the data of IPL from 2008-2012 because for predicting scores in a cricket match, recent data is more useful. Based on this, the modified training data distribution (2013-2019) was obtained as can be seen in Fig. 16.

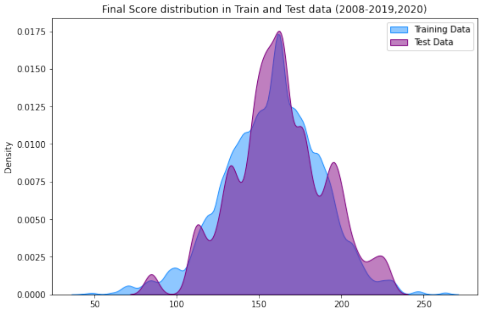

Fig. 14. Scores Distribution in IPL

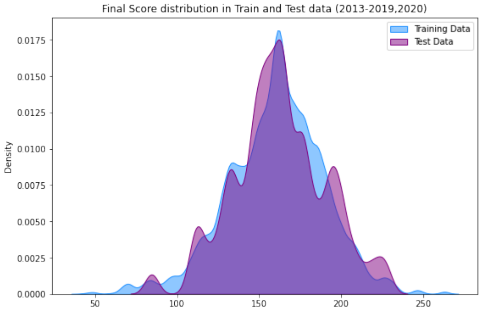

Fig. 15. Scores Distribution in IPL

Although there is only slight increase in overlap between the distributions after modifying, it makes more sense to not use very old data for prediction. After making these improvements, the final model was trained on this modified data and the results obtained are discussed below.

b) Results: After trying different hyperparameters, the final hyperparameters used for model training were as follows:
n_estimators=500 and max_depth = 250
The results were much better than expected, and the model fitted the training as well as test data very well this time.

- Custom Accuracy : 99.67
- Mean Absolute Error (MAE) : 1.37
- R_2 score : 0.994
- RMSE : 2.14

,

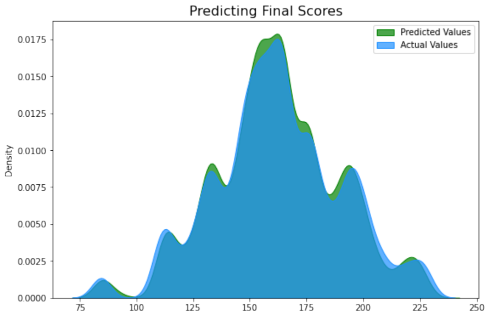

Fig. 16. Predictions done on IPL 2020 data
RF model {n_estimators = 500 and max_depth = 250}

5) XGBoost Regression: XGBoost stands for e**X**treme **G**radient **B**oosting. It is an ensemble method like Random Forest. XGBoost is an implementation of gradient boosted decision trees designed for speed and performance.
*Results :*Hyperparameter tuning was not done in this case. The parameters used were n_estimators=500, max_depth=5, subsample=0.8 and colsample_bytree=0.8. Here, 'subsample'is the fraction of the training samples (randomly selected) that are used to train each tree and 'colsample_by_tree'is the fraction of features (randomly selected) that are used to train each tree. The accuracy obtained by the XGB Regressor model was about 66.36% and RMSE of 11.43.

6) Neural Networks(NNs): Neural Networks draw inspiration from the human brain and try to mimic it. Neurons(Nodes) in the NN perform a dot product between the inputs and weights, add biases, apply an activation function, and give out the outputs. When a large number of neurons are present together to give out a large number of outputs, it forms a neural layer. Finally, multiple layers combine to form a neural network.

a) WorkFlow: NNs have lot of hyperparameters which need to be tuned in order to get good results. The hyperparameters taken under consideration were :

- Activation Function
- Dropout Rate
- Loss Function
- No. of Hidden Layers
- Batch Size
- Learning Rate

'Linear', 'ReLU', 'Leaky ReLU'were the different activation functions that were tried. It was found that 'Linear'gave much better results as compared to the other two. Increasing the Dropout Rate in the beginning increased the accuracy, however later it began to decrease. The optimal Dropout Rate chosen was about 0.3. The loss functions used were Mean Absolute Error (MAE) and Mean Squared Error (MSE). The accuracies obtained were 46.25% and 43.84% respectively. Thus, MAE is a better choice for loss function. On increasing the no. of hidden layers from 6 to 8 slight decrease in accuracy was found. The chosen batch size was 32 and on increasing/decreasing its value a drop in accuracy was observed. The final model obtained was trained at learning rates of 10^{-3} and 10^{-4}, out of which the latter one had slightly better results.

Thus, to sum up hyperparameter tuning helped in increasing the accuracy from 43.84% to 46.78%. The improvement is not too significant. [3]The NN architecture of the final NN used for training is as shown in Fig. 17

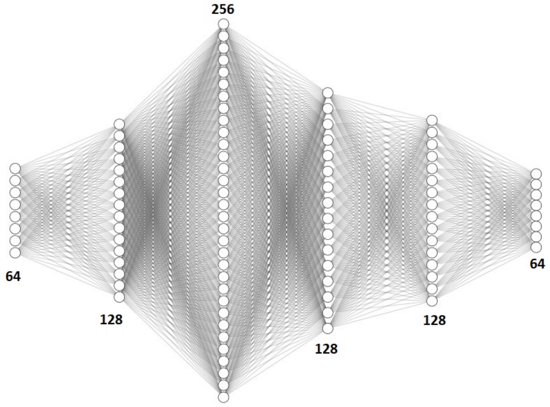

Fig. 17. Neural Network Architecture for Score Prediction

b) Results: Mean Absolute Error is used as a loss function for back-propagation. Since Early Stopping with a patience of 3 epochs was used, the final model was found to be trained for about 15 epochs. The results obtained are as follows => 'MAE': 14.45, 'RMSE': 19.17, 'R2score': 0.53 and 'Accuracy': 46.78 So, the accuracy is around 47% which is not at all usable for practical purposes. Also, it was observed that most of the models trained converged quickly and the validation loss and train loss lines became almost horizontal.The main reason behind NNs performing so bad seems to be the availability of less data.

B. Winner Prediction

For match winner prediction, 'Target Score'feature was engineered since the classification accuracy coming out earlier was quite low. The results obtained for winner prediction in IPL are not quite good. Primarily because of the nature of the league, in which almost every team is equally likely to win in any match. There are no weak teams as such. Moreover, the data available is also less, for just 812 matches.

1) Feature Selection using Extra Trees Classifier: The Extra Tree Classifier is an ensemble algorithm that seeds multiple tree models constructed randomly from the training dataset and sorts out the features that have been most voted for. It fits each decision tree on the whole dataset and random sampling of data is done while splitting at the nodes where it selects the best feature.

The final result is that it returns the features in descending order according to the Gini index (a criterion to measure information gain). Th results obtained can be seen in Fig. 19 The features shown in Fig. 19 above would have allowed

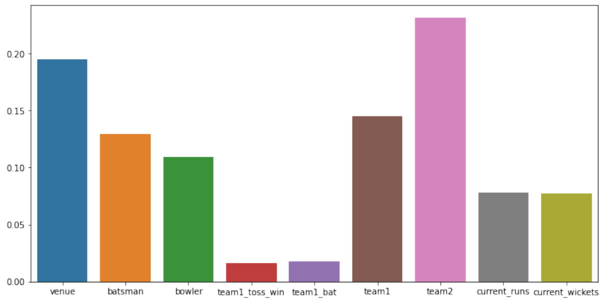

Fig. 18. Feature Importances

for more flexibility in prediction. However, on training models initially it was observed that the data available is not sufficient to get good results with these many features. So, for the sake of simplicity the features shown below were considered which gave slightly better results.

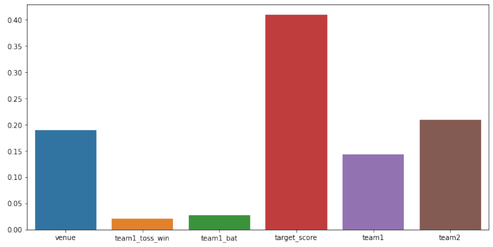

Fig. 19. Features for prediction after first innings

We can see that ‘Target score’has very high importance. As claimed earlier in Section III, toss decisions don’t play much significant role in who wins, which can be confirmed from Fig. 19

2) Logistic Regression: Logistic regression, despite its name, is a classification model rather than regression model. It is a very simple model like Linear Regression which can be used to get a decent base estimate of accuracy in binary classification problems. *Results:* The accuracy obtained was 46.67% with a F1-score of 0.63 and Area under ROC (Receiver Operating Characteristic) curve was 0.47.

3) Support Vector Machine (SVM): SVM has been discussed before in SVR. It does classification based on statistical approach instead of probabilistic approach (as in Logistic Regression). The downside is that it takes more time to train. I used a simple ‘Linear’kernel with C = 0.5 (where C quantifies the penalty for samples inside the margin).
Results: SVM gave about 50% accuracy on the test set by always predicting 1. This, tells us that Logistic Regression performs worse than if we kept predicting only 0 or 1. Hyperparameter tuning needs to be done to obtain better results using SVM.

4) Decision Tree Classifier: Decision Tree Classifier is a Supervised Machine Learning Algorithm that uses a set of rules to make decisions, similarly to how humans make decisions. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

When tuning of max_depth was being done, it was seen that the accuracy graph against max_depth was quite random and choosing any max_depth ahead of 10 gave similar results. *Results:* F1-score of 0.54 was observed. The accuracy varied each time prediction was done. On average it was about 55%. The best accuracy obtained was 58.33% and minimum being 53.67%. Thus, it outperformed the last two algorithms discussed before. The confusion matrix of the predictions made on test set is shown in Fig. 22

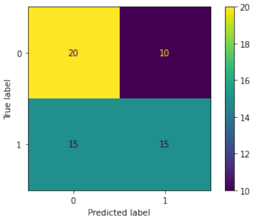

Fig. 20. Confusion Matrix *(Decision Tree Classifier)*

5) Random Forest Classifier: The Random Forest classifier is a collection of Decision Trees that are associated with a set of bootstrap samples that are generated from the original data set. It uses bagging and feature randomness when building each tree to create an uncorrelated forest of trees whose prediction is more accurate than that of any individual tree.

Randomized Grid Search was done to obtain the best set of parameters for the RF Classifier. The best parameters obtained for n_estimators $\geq$ 100 are as follows :
’bootstrap’: False, ’max_depth’: 500,
’max_features’: ’sqrt’, ’min_samples_leaf’: 1,
’min_samples_split’: 10, ’n_estimators’: 55

Results: The cross-validation score obtained was about 61% and unlike Decision Tree Classifier remained almost constant. F1-score was equal to 0.58 and Area under the ROC curve was 61.67. Thus, we see that Random Forest is better and more robust than Decision Trees.

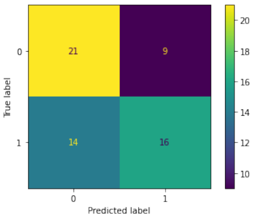

Fig. 21. Confusion Matrix *(Random Forest Classifier)*

6) Neural Networks: NNs were used to predict the winner at any time during the match unlike the models discussed before where prediction was done after the first innings. This has two advantages, one being that it is dynamic and can be used during a live match ,and the other is that it gives us more data to work with. Still, NNs need lot more data than what

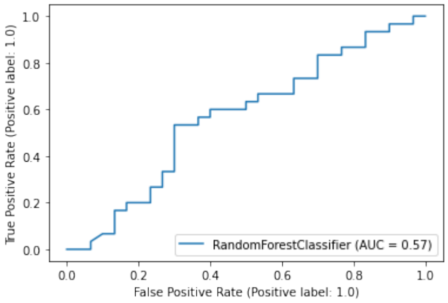

Fig. 22. AUC-ROC curve *(Random Forest Classifier)*

we have. As a result, the predictions are not very accurate but they are good enough considering the fact that the prediction task is more complex now.

Results: After trying many possible NNs, the final model was chosen which had accuracy of 53.84% and F1-score equal to 0.60. The accuracy plot for the same is given in Fig. 23

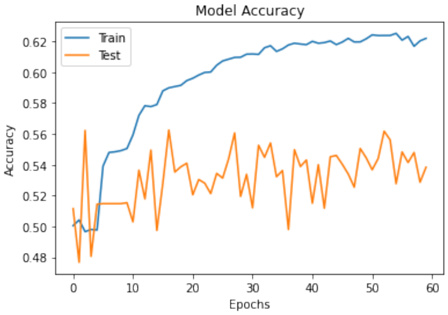

Fig. 23. Accuracy vs. No. of Epochs

V. Discussion

The performances of all the algorithms are summarized in the Table II

Algorithm	RMSE	Accuracy
Lasso Regression	19.45	45.76%
Linear Regression	19.38	47.20%
SVR	19.29	45.74%
Neural Network	19.17	46.78%
XGBoost Regression	11.43	66.36%
Random Forest	2.14	99.67%

TABLE I
SCORE PREDICTION RESULTS

Algorithm	F1 Score	Accuracy
Logistic Regression	0.63	46.67%
SVM	0.67	50.74%
Neural Network	0.60	53.84%
Decision Tree	0.54	55%
Random Forest	0.58	61.02%

TABLE II
WINNER PREDICTION RESULTS

VI. Conclusion and Future Scope

This paper provides useful insights from IPL dataset which can used for prediction and other purposes. Although both score prediction and winner prediction have been discussed, more focus is on building accurate score predictors. For winner prediction, more work needs to be done to improve the accuracy of the classifiers. Nevertheless, it does give good idea about how to proceed with the same. The claim that winner doesn't depend on toss decision has been proved subsequently while doing feature selection in Section IV B. Ways which can help team managements and owners in comparing and selecting different players have also been discussed.

I have discussed in depth the approach and various techniques to solve the problems which one may encounter while building ML models as well as how to increase the performance of the model. The conclusions drawn in Section III and Section IV can help in building better models with higher accuracy. The prediction of final score at any given moment of match is currently done with the help of Current Run Rate(CRR). However, it doesn't take into account a lot of factors like playing conditions, wickets fallen, etc.

The models proposed in this paper take these features into account to predict the final score given these features at any point in the game. Despite having limited amount of data, I have been able to achieve a custom accuracy of 99.67% i.e. the model almost always succeeds in predicting the final score within 10 runs bound of the actual score. The results for winner prediction aren't that good as data for that purpose is even lesser and predicting winner is much more complex task than score prediction.

Future work includes improving the classifiers used for winner prediction. Time series forecasting can be done to predict the future performance of players. This can be immensely useful since it can aid in Fantasy League Predictions and selecting Fantasy Teams. Finally, an easy to use user interface can be made for the different tasks covered above.

Acknowledgment

I would like to thank my mentor Prayas Jain for constantly guiding and helping me through the project. I would also like to thank Analytics Club, IIT Bombay for organizing Winter in Data Science and giving me an opportunity to explore the vast field of data science and work on such an interesting project.

References

[1] IPL Complete Dataset (2008-2020), Prateek Bhardwaj, 2020, https://www.kaggle.com/patrickb1912/ipl-complete-dataset-20082020

[2] Exploratory Data Analysis of IPL Matches-Part I, Bipin P., 2020, https://towardsdatascience.com/exploratory-data-analysis-of-ipl-matches-part-1-c3555b15edbb

[3] Predictive Data Analysis using Neural Networks, Prayas Jain, https://colab.research.google.com/drive/1o5-zSkT6aPIkN4BdJT72GcmHEcXwuiFS?usp=sharing

[4] Predictive Data Analysis of an IPL Match, Geet Pithadia, 2020, https://towardsdatascience.com/predicting-ipl-match-winner-fc9e89f583ce