Data Analysis and Winner Prediction of IPL (Indian Premier League)

Avichal Jain
Mechanical Engineering
IIT Bombay
200020035
200020035@iitb.ac.in

Divij Goyal
Mechanical Engineering
IIT Bombay
200020048
200020048@iitb.ac.in

Eknoor Singh
Mechanical Engineering
IIT Bombay
200020051
200020051@iitb.ac.in

***Abstract*— Cricket has been an integral part of the Indian culture for decades now, and the Indian Premier League (IPL) has been the torchbearer of this craze and culture for almost 14 years. This very craze motivates us for our project. The outcome of a particular match in the league depends on various variables like the playing 11 that day, previous performance, the venue, and other volatile variables like pitch and weather conditions. Our work is based on models of linear regression and Exploratory Data Analysis from the data collected from trusted websites like Kaggle. We also find out trends or relations among various variables in the data present to help analyze different matches. This analysis helps us to predict the outcome of any match provided that the variables mentioned are provided. This can help in further decisions like Fantasy Leagues and other spheres. The data collected is only for IPL, but the same process can be followed for similar data (if available) for different cricket tournaments across the globe.**

I. INTRODUCTION

Started in 2008, the Indian Premier League (IPL) is a professional Twenty20 cricket league, contested by ten teams based out of ten Indian cities. The Board of Control founded the league for Cricket in India (BCCI) in 2007. It is usually held between March and May of every year. The IPL is the most-attended cricket league globally and in 2014 was ranked 6th by average attendance among all sports leagues. The league began with eight teams and has had 14 seasons till now.

Data Science is extensively used in sports and cricket, for that matter, has a good amount of use of Data Science and Algorithms. Real-time data analytics can help gain insights even during the game for changing tactics by the team and by associated businesses for economic benefits and growth.

Out of the several projects we saw related to data science, we chose this as mentioned we felt IPL is an integral part of ourselves and the fact that it is not only limited to T-20 Cricket but also for other formats like ODIs and Test Matches. Moreover, a similar analysis can be performed for other sports like football, hockey and baseball, having somewhat similar parameters.

The data source we have primarily used for drawing conclusions is from the online sources like Kaggle. Primarily in the project we have done the following things,

- We have studied the data and carried out exploratory data analysis to identify factors which might play an important role in determining the winning team in a particular match.
- During EDA, we have also tried to identify trends in match results based on the fact that which team wins the toss, which team bats first, which teams bowls first
- We have further used a regression model to understand which factors play a role in what proportion in determining the match winner

With this done, our primary aim is to use the Plotly library in Python to render interpretation efficiently using graphs. Performance data using visual analysis help select players for future matches and provide additional information about the player and team profiles. The aim is to provide detailed insights numerically and graphically to understand the tournament's history and make data-driven decisions like predicting the winning side of a particular match in the future with an acceptable accuracy solely based on the parameters mentioned above. We realize that the winners are decided by the squad playing at that time, but this is a very volatile parameter and can be ignored for now.

II. Background and Prior Work

A basic understanding of Machine Learning using Python is sufficient to follow our solution report. One can familiarise oneself with the various libraries used in the code easily. The report also explains the various prediction models used, in the methodology section. While basic work in this sector exists, as can be seen in the references, we tried to experiment with a diverse variety of models and a lot of variables in order to get the best possible prediction with the available data.

III. Data and Methodology

The dataset we have used is from the online repositories from a trusted data repository website named Kaggle.

We have primarily used four datasets i.e. two for each part of the code.

1. IPL Ball-by-Ball 2008-2020.csv
   The data shows all the ball-by-ball outcome of a particular match ranging from the striker, to the outcome of the ball like a single, a six or a wicket.
2. IPL Matches 2008-2020.csv
   The data shows all the details of a particular match like the date, the venue, and the winning team, the man of the match and umpires of the match.
3. matches.csv
   The data provides the details as match result, venues, umpires, win margin (in terms of wickets or in terms of runs).
4. deliveries.csv
   The data shows various variables like the players batting per bowl, the bowler, the outcome of each ball, reason of dismissal (if) and type of extra run conceived (if).

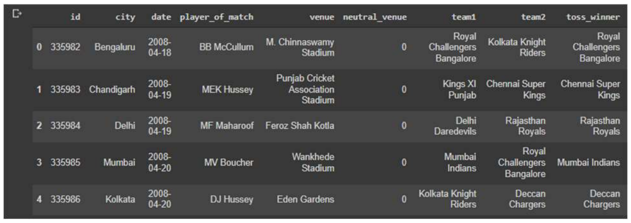
The data consists various variables whose name either has been used the same or has been changed according to the need. The number of teams has been a changing variable in these 14 years, with eight teams in the debut season. There has been a constant addition and deletion of various teams like

1. Deccan Chargers was active from 2008 to 2012.
2. Pune Warriors India was active from 2011 to 2013.
3. Kochi Tuskers Kerala only played one season in 2011.
4. Sunrisers Hyderabad was introduced in 2013 and is currently active.
5. Chennai Super Kings and Rajasthan Royals did not play in 2016 and 2017.
6. Gujarat Lions and Rising Pune Supergiant played only in 2016 and 2017.
7. Rising Pune Supergiant played as Rising Pune Supergiants in 2016 (i.e. deleted the last 's' in 2017)

As IPL is generally played between March and May of every year and this includes the disparity in 2020 (September- November) due to COVID-19 Pandemic. Many other tasks have been performed for Data Inspection and Cleaning like,

1. The name 'Bangalore' has been replaced with 'Bengaluru' everywhere in the dataset.
2. Missing values have been filled to prevent errors.

All this inspection and cleaning was done in all the four datasets to obtain a more usable form of data. For example, the final output for IPL Matches 2008-2020 dataset has **816** rows and **17** columns.



	id	city	date	player_of_match	venue	neutral_venue	team1	team2	toss_winner
0	335982	Bengaluru	2008-04-18	BB McCullum	M. Chinnaswamy Stadium	0	Royal Challengers Bangalore	Kolkata Knight Riders	Royal Challengers Bangalore
1	335983	Chandigarh	2008-04-19	MEK Hussey	Punjab Cricket Association Stadium	0	Kings XI Punjab	Chennai Super Kings	Chennai Super Kings
2	335984	Delhi	2008-04-19	MF Maharoof	Feroz Shah Kotla	0	Delhi Daredevils	Rajasthan Royals	Rajasthan Royals
3	335985	Mumbai	2008-04-20	MV Boucher	Wankhede Stadium	0	Mumbai Indians	Royal Challengers Bangalore	Mumbai Indians
4	335986	Kolkata	2008-04-20	DJ Hussey	Eden Gardens	0	Kolkata Knight Riders	Deccan Chargers	Deccan Chargers

Fig. 1. Snapshot of the final table for IPL Matches 2008-2020 dataset

NOTE: The table above is just a snapshot, hence does not have all the columns of the actual dataset.

METHODOLOGY FOR PREDICTION

We experimented with various models and predictor variable combinations. We narrowed our choices down to 'team1', 'team2', 'venue', 'toss_winner','city', 'toss_decision', based on the correlations existing. In various models, we experimented with the various combinations of features used for predicting, the results of which are tabulated below.

First, we used a logistical model. Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

Secondly, we experimented with the Random Forest Classifier. The Random forest or Random Decision Forest is a supervised Machine learning algorithm used for classification, regression, and other tasks using decision trees.

The Random forest classifier creates a set of decision trees from a randomly selected subset of the training set. It is basically a set of decision trees (DT) from a randomly selected subset of the training set and then It collects the votes from different decision trees to decide the final prediction.

Next we used a SVM Model. Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

Finally, we used a decision-tree classifier model. Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation.

IV. Experiments and Results

Since we have two final objectives here, analysing the different variables present in a match and then using them in the results section to make a good model to predict outcome of any future match given that all the data regarding the variables mentioned are provided. The analysis done can be seen as,

A. *Matches*

The date of the matches has been used to classify them into seasons (years). The following is a plot of match distribution.

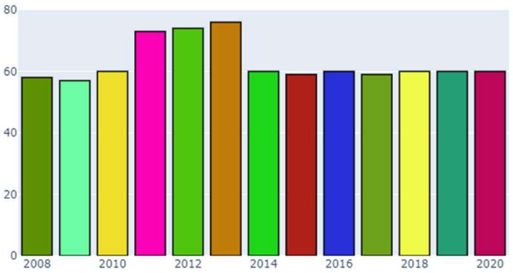


Fig. 2. Bar graph of season wise match distribution

As we see there has been a sudden increase in number of matches in seasons 2011 to 2013, due to more number of teams in the playing.

B. *Toss*

The following is the graphical representation of number of toss(es) won by the teams

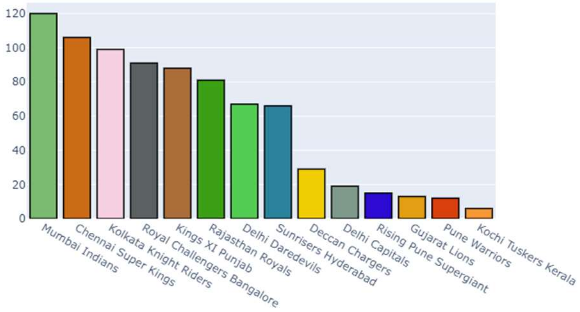


Fig. 3. Total number of wins by each team till 2020

We see that Mumbai Indians has won the most toss(es) and Kochi Tuskers Kerala, the least.

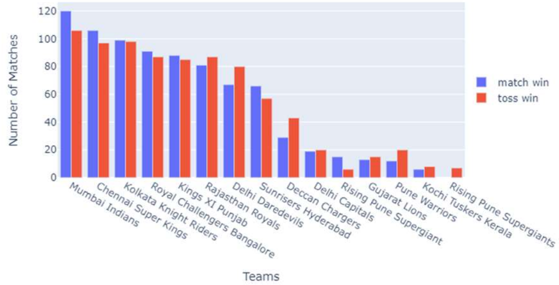


Fig. 4. Total number of wins v/s toss wins by each team till 2020

Now we see the comparison between the win of matches and win of toss(es) by each team. We can see a general trend of decline from Mumbai Indians to Delhi Capitals that Match Win is almost a bit higher than the Toss Win, i.e. if a team wins the toss, and hence has its will to chose to bat or to bowl first, has a better chance at winning. After Delhi Capitals, we see a drastic change in this trend for the following teams.

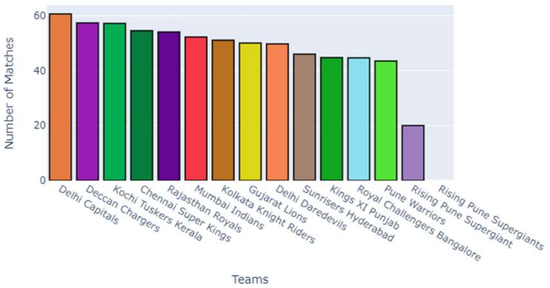


Fig. 5. Toss win success ratio

Here we have plotted Toss Win Success Ratio i.e. “Toss Won / Total Matches Played” and to our surprise we see Delhi Capitals leading it and Rising Pune Supergiant/-s in the last.

C. *Player of the Match*

Here we plot the number of Man of The Match titles won by various players in each match from to 2008 to 2020.

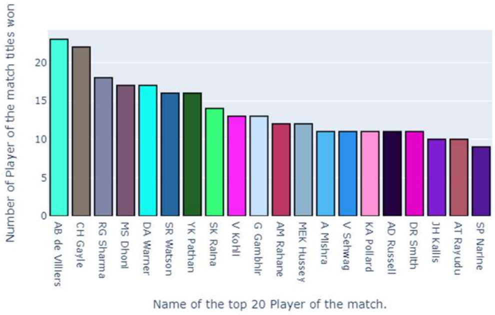


Fig. 6. Man of the Match statistics

We see that AB de Villiers is in the lead with most number of MotM titles.

D. *Success Rate of winning matches*

The following plot shows the variation of Success Rate of winning matches for each team.

The quantity is measured as “Matches Won / 100 Matches Played” and we see Chennai Super Kings being in the lead for this variable.

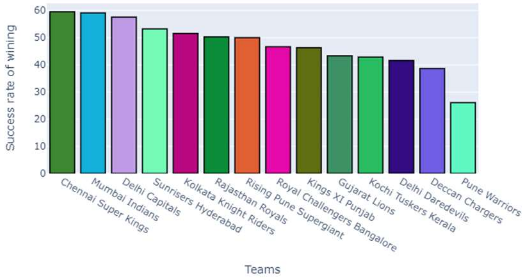


Fig. 7. Success rate of Teams

E. *Most IPL Titles*

This is a normal plot for Team v/s the number of titles that is seasons won by each team.

Here we see Mumbai Indian is in a whopping lead with 5 titles win.

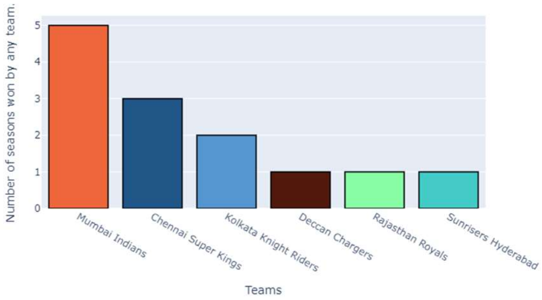


Fig. 8. Most Successful Teams

F. *Top Batsmen*

The following plot is the Runs v/s Player plot, showing total number of runs scored by each player in his complete IPL career.

Here we see Virat Kohli being in the lead.

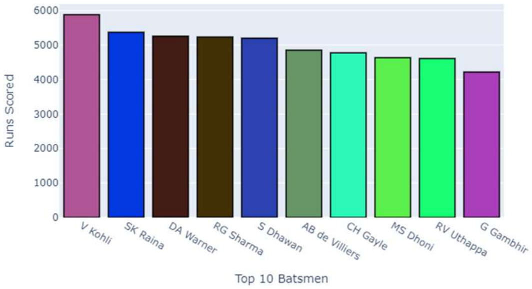


Fig. 9. Top Batsmen (in terms of Runs)

Now we plot the batting performances of various players using the formula “Strike Rate / Innings Score” and we see Chris Gayle leading the graph, due to his well known explosive style of batting.

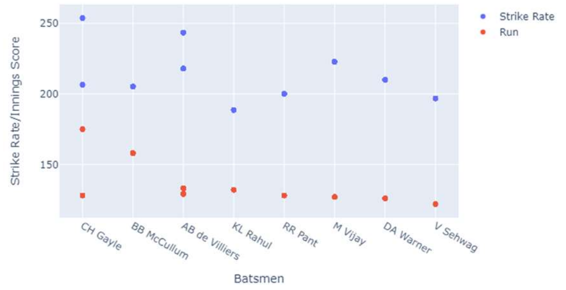


Fig. 10. Top Batsmen (performance in Match)

G. *Top Bowler(s)*

The following plot is the Wickets v/s Player plot, showing total number of wickets taken by each player in his entire IPL career.

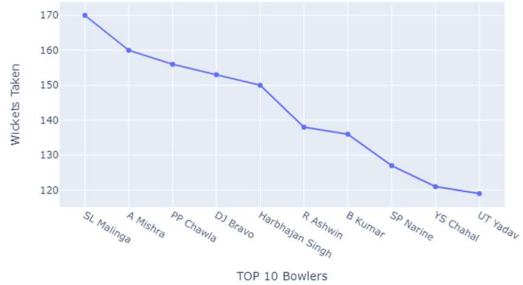


Fig. 11. Top Bowlers

Here we see Lasith Malinga being in the complete lead with almost 170 wickets.

Now we plot the bowling performances of various players using the formula “Runs per Wicket / Wickets” and we see A.S.Joseph leading the graph.

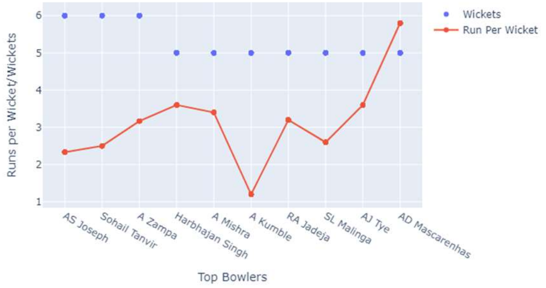


Fig. 12. Top Bowlers (performance in Match)

H. *Top Fielder(s)*

The following is a plot of the most successful fielders in the IPL and we see M.S.Dhoni leading the graph followed by various other wicket keepers as well as we realise that caught behind wickets, could be the most common type of dismissal.

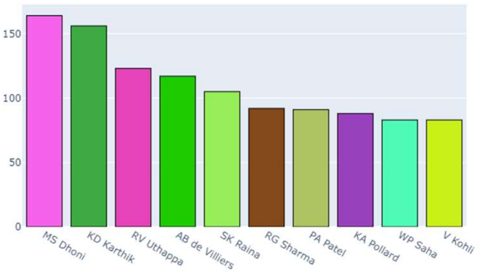


Fig. 13. Top Fielders

I. *Top Venue(s)*

The following is a pie chart of the most common venues for matches of IPL.

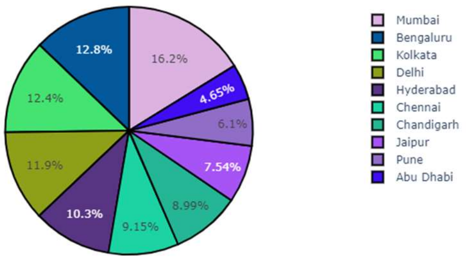


Fig. 14. Top IPL Venues

We see that Wankhede Stadium, Mumbai is the most common venue as most of the finals and other chart ending matches take place there.

J. *Top Umpires (based on number of matches)*

The following plot shows the umpires serving for most number of matches in the IPL, with S. Ravi being in the lead.

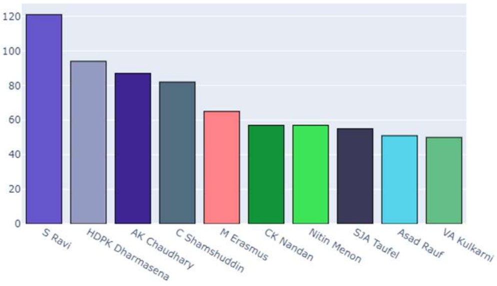


Fig. 15. Top IPL Umpires

K. *Results based on D/L Method*

The following donut chart shows that even less than 2.5% of the matches had the result predicted using Duckworth – Lewis Method, i.e. the matches which are not entirely completed due to some unforeseen weather circumstances like rain etc.

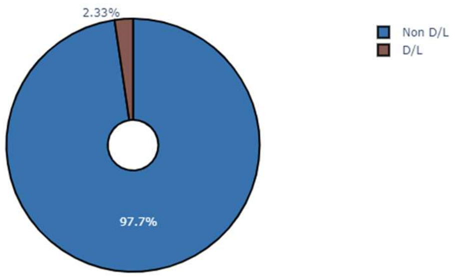


Fig. 16. Results based on Duckworth- Lewis Method

L. *Season wise summary of Matches won by Runs*

This plot shows the margin of runs with which a particular match is won by a team every season.

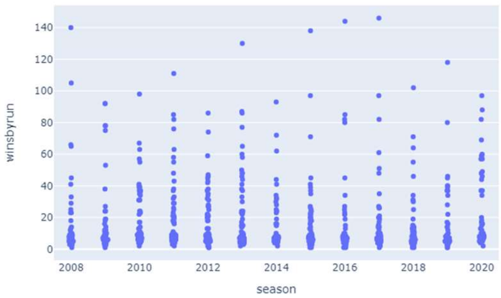


Fig. 17. Matches won by Runs

M. *Reasons of dismissals*

The following bar graph shows the top reasons of dismissals in IPL matches. We see a whopping lead in terms of number for 'being caught' followed by a great dip in dismissal due to being 'bowled out'.

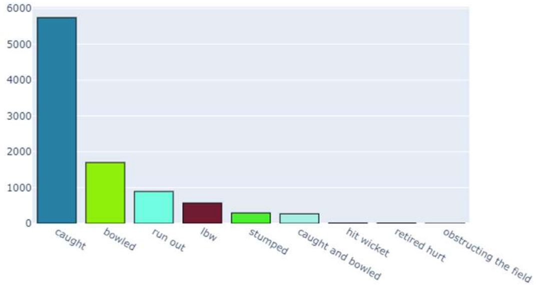


Fig. 18. Top Reasons of Dismissals

N. *Analysis of Over-Wise Runs and Wickets*

The following graph shows the total runs concieved over wise and we see that majority of runs are scored after 15th over, that is the death overs.

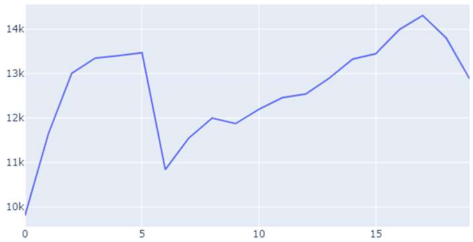


Fig. 19. Total Runs

The wicket – over plot also gives a similar result that most wicktes are taken in the death overs as well.

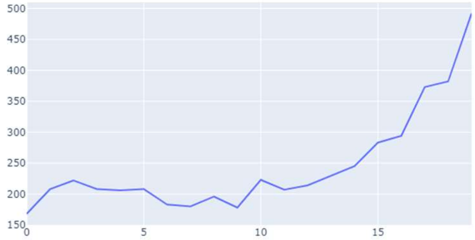


Fig. 20. Total Wickets

The total number of extra runs i.e. runs due to wide ball, leg-by, no ball etc., is a bit unpredictable variable, having a lot of sudden variations. Yet, we can say that the extra runs are minimum near the 8-12 overs bracket.

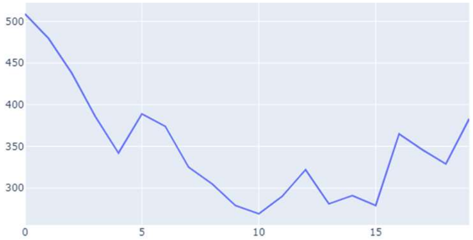
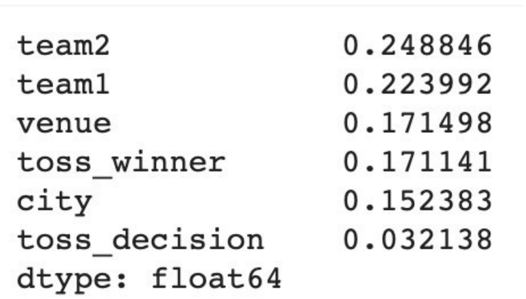


Fig. 21. Total Extra Runs

P. Prediction Part: Model and Variable Selection

Based on our domain knowledge of features, the graphical analysis done above and the feature importance scores we calculated (as below)



```
team2             0.248846
team1             0.223992
venue             0.171498
toss_winner       0.171141
city              0.152383
toss_decision     0.032138
dtype: float64
```

Fig. 22. Feature Importance Scores

RESULTS

A.
Model: Logistic Regression
Variables Used: 'team1','team2','toss_winner'
Accuracy: 24.686%
Cross-Validation Score: 22.482%

B.
Model: Random Forest Classifier
Variables Used: 'team1', 'team2', 'venue', 'toss_winner','city','toss_decision'
Accuracy: 89.151%
Cross-Validation Score: 48.594%

C.
Model: SVM
Variables Used: 'team1', 'team2', 'venue', 'toss_winner','city','toss_decision'
Accuracy: 40.723%
Cross-Validation Score: 34.425%

D.
Model: Decision Tree Classifier Model
Variables Used: 'team1', 'team2', 'venue','toss_winner'
Accuracy: 89.151%
Cross-Validation Score: 50.492%

V. LEARNING, CONCLUSIONS AND FUTURE WORK

Hence, it is evident that the final combination using Decision Tree classifier seems to give the best results and this model was used to predict the match outcomes of 2021 matches, both of which luckily came to be correct!

Even though the accuracy is not high enough to be extremely useful, owing to the limited domain of data available and a variety of factors IPL matches depend on, it gives a basic idea about the strategies and methodologies used in designing a solution to this Machine Learning problem.

Here we have calculated the accuracy of the training set and the cross validation (using K folds) using various different models- Logistic regression, Random Forest Classifier, Support Vector Machine and Decision Tree Classifier. It is very important to understand that accuracy and cross validation score are two completely different things and should not be confused. In Cross Validation score, one obtains an accuracy for each of the folds and average them. For each cross validation fold, the training and tests set are different; so, one obtains different accuracy values for each of them, and it enables to calculate standard deviation of the accuracies, which is enclosed in parentheses. Accuracy score calculates the accuracy based on the inputs prediction and the true value. For example, if one inputs the entire training set, one will get the accuracy of the entire training set, which is of course different than your cross validation score. For the cross validation using K Folds, we have set the value of n_splits to be 7. This indicates that about one-seventh of the total data is used for testing. Initially we were using and setting the value of k_folds but due to the newer version of Google Colab, it was not supporting and this was one of the challenges we faced during the making of this project. We tried to look for some other alternate on the internet and other source as well. It asked us to use stratified K fold but once again it showed the same. After numerous attempts and study, we learnt about n_splits which can be used with the proper syntax and successfully managed to run the code with the desired outcome. We learnt a lot about K folds, stratified k Folds and the correct usage of n_splits during the entire process.

Using the model of Logistic Regression and features 'team1', 'team2' and 'toss_winner', we obtained a very low accuracy score of 24.686% and comparatively low cross validation score as well i.e. 22.482%. This clearly indicates that that we have missed some of the features that play significant role in predicting the outcome of the match. In this logistic regression model of ours, we have not included some of the factors which have strong correlation with match winner and as a result of this we have got such a low accuracy score.

It is common and true in our case as well that the validation score is less than the training score This is because the model fits on training data and validation data is unseen by the model.

In our Random Forest Classifier model, we also included 'venue', 'city' and 'toss_decision' along with 'team1', 'team2' and 'toss_winner'. We saw a huge rise in the accuracy score of our model (89.151%) with comparatively high and good cross validation score of 48.277%. This is due to the inclusion of the features that play an important role in determining the winner of the match. The place in which the match is being played is the very key in deciding the match winner and same is for the decision taken by the team (whether batting or bowling) after winning the toss. Here as we have a categorical data, Random Forest Classifier was always a better option than Logistic Regression. Random Forest Classifier is more of Accuracy focused algorithm. Random selection in individual decision trees of RFC can capture more complex feature patterns to provide the best accuracy.

Once again it is clear that our accuracy score is higher than the cross validation score.

Similarly for SVM model we got a low accuracy and cross validation score whereas for the Decision Tree Classifier model we got the highest cross validation score and same accuracy score as that of RFC. This shows that DTC is the best model for us. Decision trees are much easier to interpret and understand. Since a Random Forest combines multiple decision trees, it becomes more difficult to interpret.

Our model passes through certain limitations, primarily the size of the dataset. The dataset consists of a large number of columns and rows as it is about the data of the past 13 years. the final number of columns were drastically reduced and only the ones having strong correlation with match winner were considered for our model.

These weaknesses also turn out to be the strengths of our work. The dataset is extremely extensive and covers a huge number of factors providing a good overall study of a particular match and thereby concluding the important features in determining the match winner and using only those features to build our model.

FUTURE WORK

Our method passes through certain limitations, primarily the size of the dataset. The dataset does not consist of various volatile variables like pitch condition, weather condition, playing eleven, recent performance of each player, etc.

There is scope of improvement in the model. More extensive surveying can take place to obtain complete datum about the above mentioned variables.

More collection of data and analysis will help in getting more concrete answers in future and if combined with the above mentioned variables in suitable proportion, i.e. some kind of algorithm if can be developed forming some relation of all the variables, this model can be used extensively in future IPL matches, even by the teams themselves to have a better insight about the matches and changes they should make.

This model, can further be applied to various other formats of cricket like ODI(s) and Test Matches. Also, to other sports like football, hockey and other sports having such similar variables.

Moreover, this model if completed with the said variables can also be used by general audience in many other spheres like Fantasy Leagues, an industry worth more than $ 1 billion just for IPL, and much more billions for other cricket formats and sports mentioned above.

CONTRIBUTION

The team contributed actively in the project because of the inclination and interest in Data Science and liking of cricket and IPL. The topic was mutually decided upon. The coding work and the reports, both written and explanation video was done equally by extensive collaboration of all the three team members. Contribution towards logical aspect had equal participation by the complete team.

ACKNOWLEDGEMENT

We would like to thank our professors, Prof. Amit Sethi and Prof. Manjesh K. Hanawal for teaching us this course, without which we would not have been able to complete this project. We would also like to thank all the TA’s involved with this course, who helped us a lot in doing the various assignments, and completing this course. We would also like to thank people on stack overflow - for answering whatever doubts we had during the course of this project.

APPENDIX

- Main Colab Notebook: https://colab.research.google.com/drive/1YAjdDgixKCAjd6_nveJ9SzzT9TZ71xfK
- Video Report: https://drive.google.com/file/d/1Z1wcr-ZVEFoBEydZddW1_BuKW9ltN2_M/view?usp=sharing
- Datasets: https://drive.google.com/drive/folders/1_FNLvsyoHhla7snFVYLvx_S7w_RzqgnH?usp=sharing

REFERENCES

[1] https://en.wikipedia.org/wiki/Indian_Premier_League

[2] https://www.kaggle.com/patrickb1912/ipl-complete-dataset-20082020/download

[3] https://thecleverprogrammer.com/2020/12/23/ipl-analysis-with-python/