Data Analysis of IPL & Finding Out How Much A Toss Decision Matters in Cricket

Arnav Jain and Arham Choraria
Informatics 248
University of Massachusetts, Amherst

11th May 2022

Data Source

Dataset: https://www.kaggle.com/datasets/nowke9/ipldata?select=matches.csv (https://www.kaggle.com/datasets/nowke9/ipldata?select=matches.csv) This is a dataset of all the Indian Premier League (IPL) matches, statistics, and results by team and by player from 2008 to 2019. The dataset contains around 750 rows and 18 columns. It contains the statistics of every team that has played in the IPL from 2008 to 2019. The information we will be focusing on are which teams were facing each other, which team had won the toss out of them, what the toss decision was (i.e whether they chose to field first or bat first), and which team won the match. Moreover, we will also be looking at who the player of the match was and how many runs and wickets a team won the match by which will help us visualize and understand the history of the kind of games that the league has witnessed.

The league, since 2008, has gone through a lot of changes when it comes to the teams or the players that have taken part. In 2019, the team that represented the Indian Capital, Delhi, had changed its name from Delhi Daredevils to Delhi Capitals hence, the dataset had both the names included. However, for better understanding and for simplicity, we assigned all of Delhi's data to one identifier called Delhi Capitals as this would help us get better results while outputting a team's summary for its statistics. Furthermore, there have been a few teams that have played less than 3 seasons of the tournament like Gujarat Lions and Rising Pune Supergiants which means that such teams will have very less data compared to the teams that have played many more seasons in the tournament. We have chosen not to remove data about these teams as even though they have less data, they give us more information about the rest of the teams and that can help us achieve better conclusions and results in general. Although, there are a few columns that contain information that we do not require for any sort of modeling or for any visualization purposes and we feel that we could work the same without the data. This includes columns that contain information about the umpires in the matches, whether the result used the Duckward-Lewis method or not when a play was stopped due to rain, and the ID column which we feel will not make any difference to our results or conclusion. For this reason, we have decided to remove such columns from our dataset and refine it further to suit our needs for the guiding question.

At the very beginning, we imported all the libraries we would require during the course of this project. The dataset we found was in the form of a csv file named matches.csv. Since we had recently learnt SQL, we thought it would be a good learning experience to implement it in our project. We created a database where we imported this csv file and used that db to query in the dataset. While querying we also created two new columns: "Winner_decision" and "Team_won_toss". The former tells us whether the team that had won the match Batted/Fielded first and the latter gives us information about whether the winning team had won the toss or not. The need to create these columns arose because these two columns have information that we would be using vastly in finding whether and if there is any correlation between winning the match and fielding first as well as winning the toss and winning the match. Hence these columns, ultimately serve as our predicted or predictor variables as described in the guiding question. Once we had our dataframe imported in, we then removed the unwanted columns (as mentioned above) and NA values from the dataset. Lastly, we converted the categorical columns into factors so that we can use those columns in our various modeling techniques and finally we disconnected the db from the rmd file.

Guiding Question

The domain of the project is a Sports Analysis on Cricket.

Cricket is a popular and widely recognized sport. It is a game where you don't just play for yourself but for your entire team and even your nation. The game has evolved through the decades to become faster, more engaging and more entertaining. Following this, there have been several leagues and formats introduced to catch a larger audience. The Indian Premier League (IPL) is a professional men's Twenty20 cricket league, contested by teams based out of Indian cities. A few teams our dataset consists of are Chennai Super Kings, Delhi Capitals, Kolkata Knight Riders, Mumbai Indians, Kings XI Punjab, Rajasthan Royals, Royal Challengers Bangalore and Sunrisers Hyderabad.

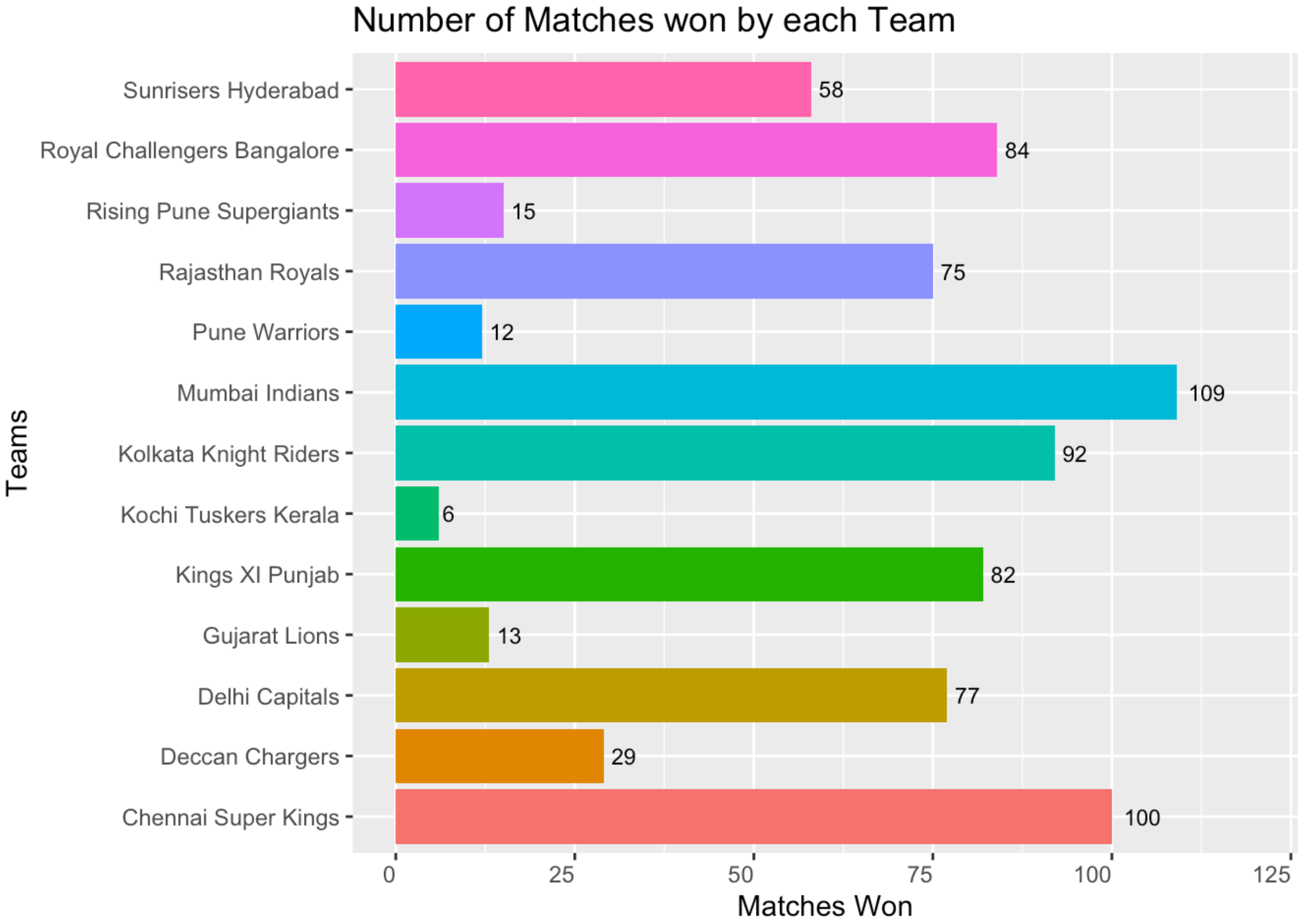
IPL is one of the most-attended cricket leagues in the world. The league consists of some of the most world-renowned players and is known for its nail biting match finishes. Our motive is to analyze the game of Cricket through this league in particular. Cricket is a game where several strategies are imposed. Whether you bat or field, the tactics that your team uses can change the entire game within the span of an over. However, there is one factor that decides how a team will start playing the match; the Toss. It is simply a coin flip that decides whether a team will bat or field first. Several journalists and sport enthusiasts say that winning the toss is like getting a head start for winning the game. It is quite ironic how a flip of a coin, that has a probability of 50% for its outcome, can decide the team's fate for the entire match. But we are interested in finding out whether winning the toss and deciding what to start with in specific (batting or fielding) makes a difference in the outcome of a match or not. As the IPL takes place annually and has players being involved from all around the world, we think that we will get up to date information and analyzing its data will provide us with reliable results. Furthermore, we will be analyzing the IPL in general by visualizing its journey so far, by showing the most successful teams/franchises, players along with their performance in general and other statistics. We will also be focusing on two questions which are whether winning the toss increases your chances of victory or not and whether a particular toss decision results in higher chances of winning the game or not.

Exploring and Visualizing the Data Set

After checking the summary of the data, a lot of variables and their statistics were shown in a concise manner. The dataset consisted of an 'id' column that acted as an identifier for each row. It also consisted of a 'Season' and a 'Date' column which indicated when the matches were played and what edition or season they were played in (the first edition was in 2008). Variables 'City' and 'Venue' showed which cities the matches took place in and which stadium they were played in as well which gave us demographic information about the matches and the tournament in general. Following that, there were 'Bat_first', 'Field_first', 'Toss_winner_team', and 'Toss_decision' variables which showed the names of the teams that batted first, fielded first, won the toss, and what their decision of the toss was (bat or field). There was also information about which team won the match which was identified by the ‘Winner_team’ column and how many runs or wickets they won by which was under the ‘win_by_runs’ and ‘win_by_wickets’ columns which were numerical. The ‘player_of_match’ column identified which player got the player of the match award in the match for their individual brilliance in the game. These were most of the columns in the dataset and we will be using several of them in the project for visualization and modeling purposes.

Which Team has won the most number of Matches?

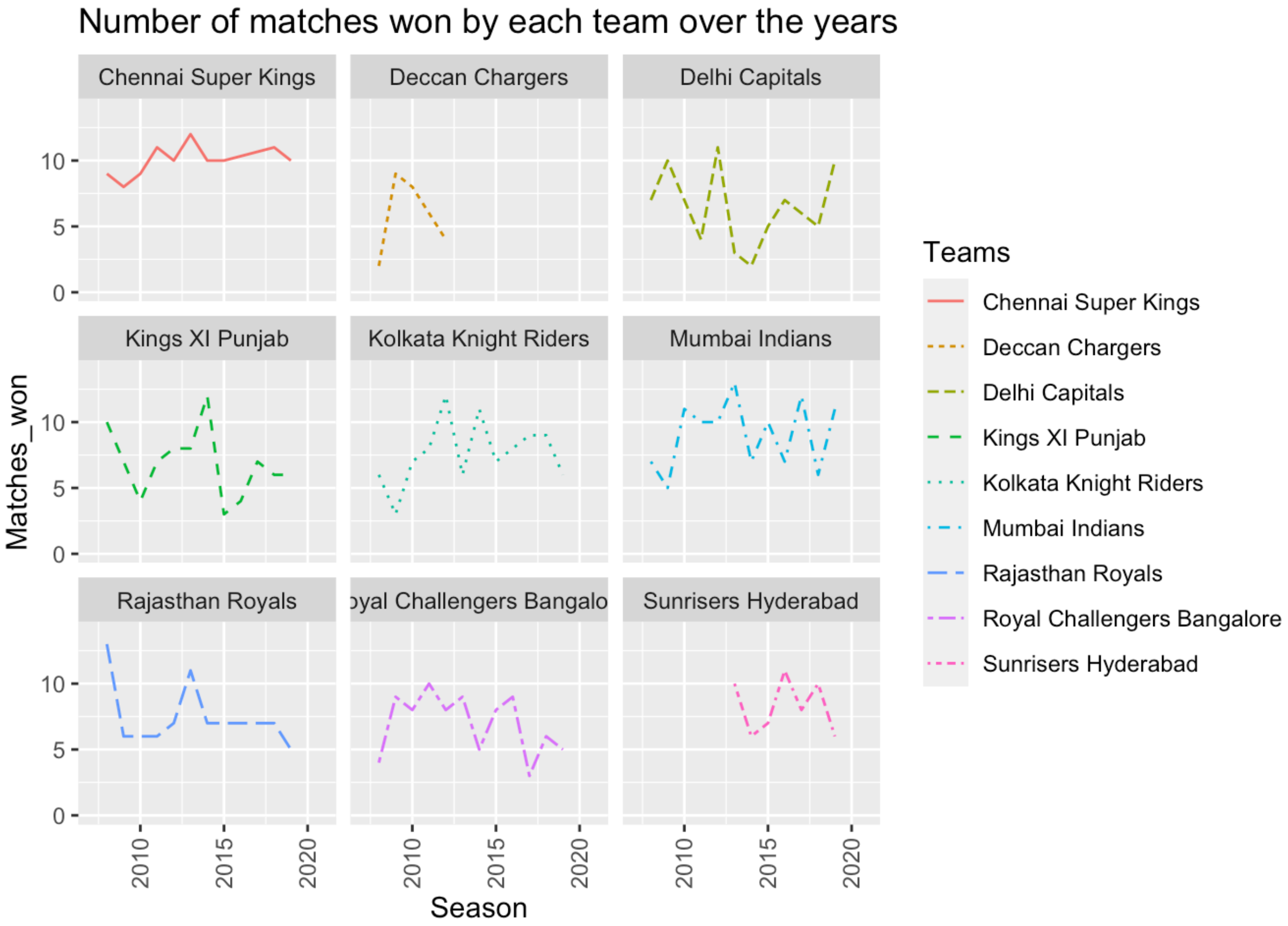
Let us find out the most successful team in the IPL through the number of total match wins each team has had in the history of the tournament from 2008 to 2019. In order to do this, we found out the number of matches won by each team from the main dataframe and stored it in a separate dataframe which we then used for our visualization. To do this, we used the ‘sqldf’ library and found out the distinct teams and their number of occurences from the ‘Winner_team’ column by querying the main dataframe with SQL. The result of this query was finally used for the required visualization. We have used a similar process to compute the required values which we have used to visualize throughout the project.



Here, we can see that Mumbai Indians have won the most number of matches followed by Chennai Super Kings and then Kolkata Knight Riders. Not so surprisingly, Mumbai Indians is also the team that has won the most number of IPL titles which is 4 from 2008 to 2019. This is followed by Chennai winning 3 IPL titles and then Kolkata winning 2. Therefore, the total number of wins is a good representation of the most successful team in the tournament.

Which team has been more consistent over the years?

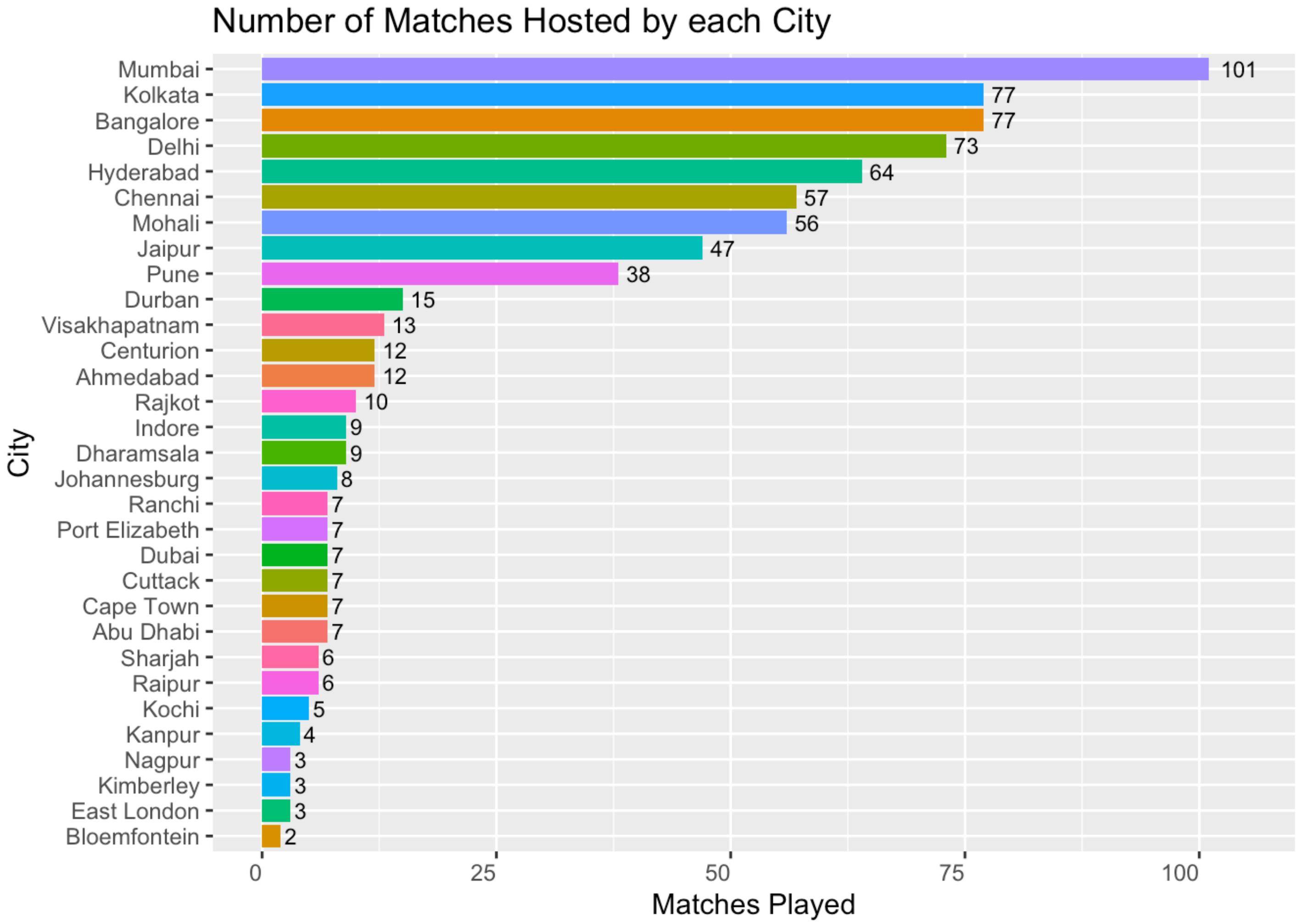
Now that we know the most successful teams of the tournament, it would be great to visualize their performance in each season. Since a few teams have played only 2-3 seasons and have less data comparatively, we have decided not to include them in the graph.



Here we can see how each of the teams have performed in the tournament through the years. The graphs of most of the teams fluctuate between more wins in a few seasons and less in the other. However, we clearly see that Chennai Super Kings have been the most consistent team in terms of winning matches over the years after analyzing their graph. This reiterates why they are one of the most successful teams in the entire league, as we had seen in the previous graph, and have multiple trophies to their name.

Which cities have hosted the highest number of IPL Matches?

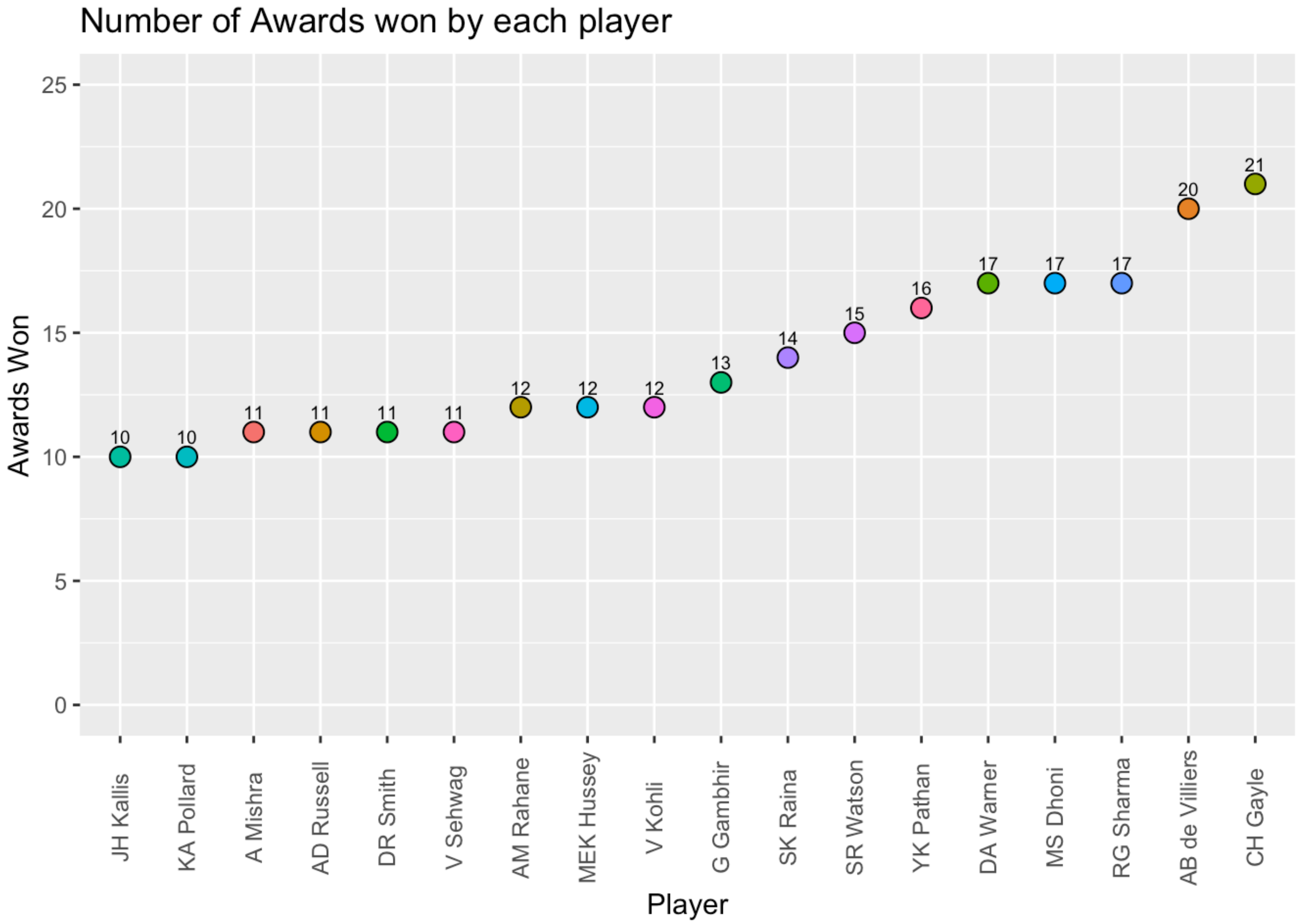
Home and away advantage plays a big role in major sport tournaments so it would be interesting to see which cities or venues have hosted the highest number of matches in the IPL.



By looking at the barplot above, we notice that Mumbai has hosted the most number of matches followed by Kolkata and then Bangalore. Mumbai Indians is also the team with the most number of wins and Kolkata Knight Riders is third on the list followed by Royal Challengers Bangalore being fourth as we saw in the first bar plot. This could suggest that home advantage does play a role in the IPL as well because the city that has hosted the most number of matches has the most successful team. One thing to notice here is that even though Chennai Super Kings is the second most successful team in the IPL, Chennai city does not lie in the top 3 when it comes to the number of matches hosted. A good reason for this could be that matches weren't played in Chennai due to domestic problems in the city itself and Chennai Super Kings had to play their matches elsewhere.

Which players have got the most Player of The Match awards?

IPL is an excellent tournament for players to showcase their skills and individual brilliance in order to play for their national teams so this is a tournament where several players would want to stand out and win matches for their teams. There have been numerous cases where players have changed the entire game and made their team win. Of course, their efforts are recognized and players are awarded with a player of the match award at the end of every match. Let's find out which individual players have been the most successful for their team in the league.

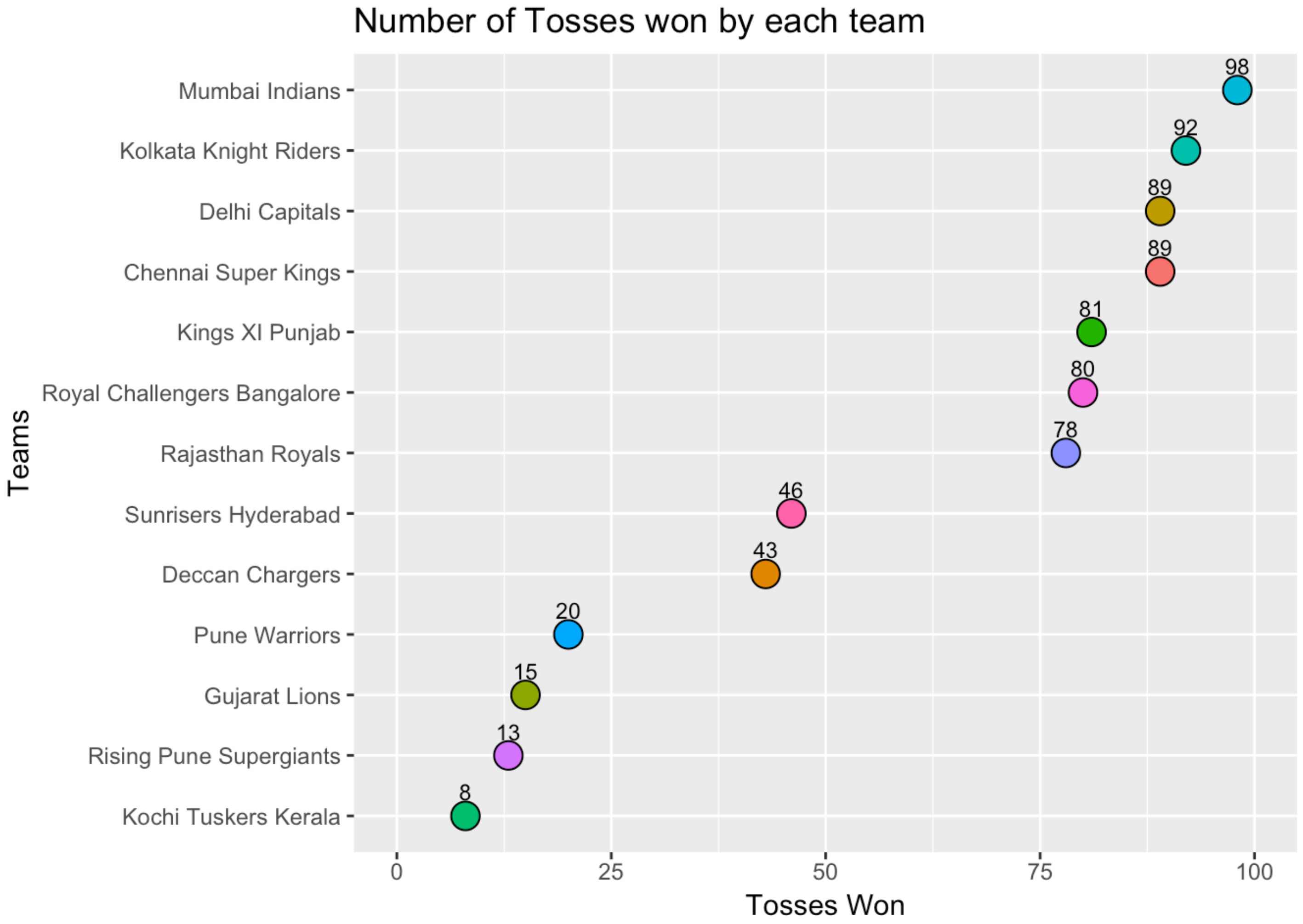


With the help of the dotplot above, we can see that CH Gayle has won the most number of player of the match awards (21) followed by AB de Villiers with 20. The list is then followed by MS Dhoni, RG Sharma and DA Warner having 17 awards each. The interesting fact here is that both CH Gayle and AB de Villiers have played for the Royal Challengers Bangalore, however, Bangalore has never won the winner's trophy in the tournament and is the team with the 4th most number of wins. Even though this seems surprising, this suggests and supports the fact that cricket, at the end of the day, is a team sport and even though individual players may win you several matches, you need the entire team to help you win the tournament and the trophies.

Now that we have seen the history of the tournament based on which teams have been successful and which players have been commendable, let's shift the focus to our guiding questions. We are interested in finding out whether winning the toss increases the chances of winning the match and whether a specific toss decision (batting or fielding) increases the chances of victory or not.

Which Teams have won the most number of Tosses in the Tournament?

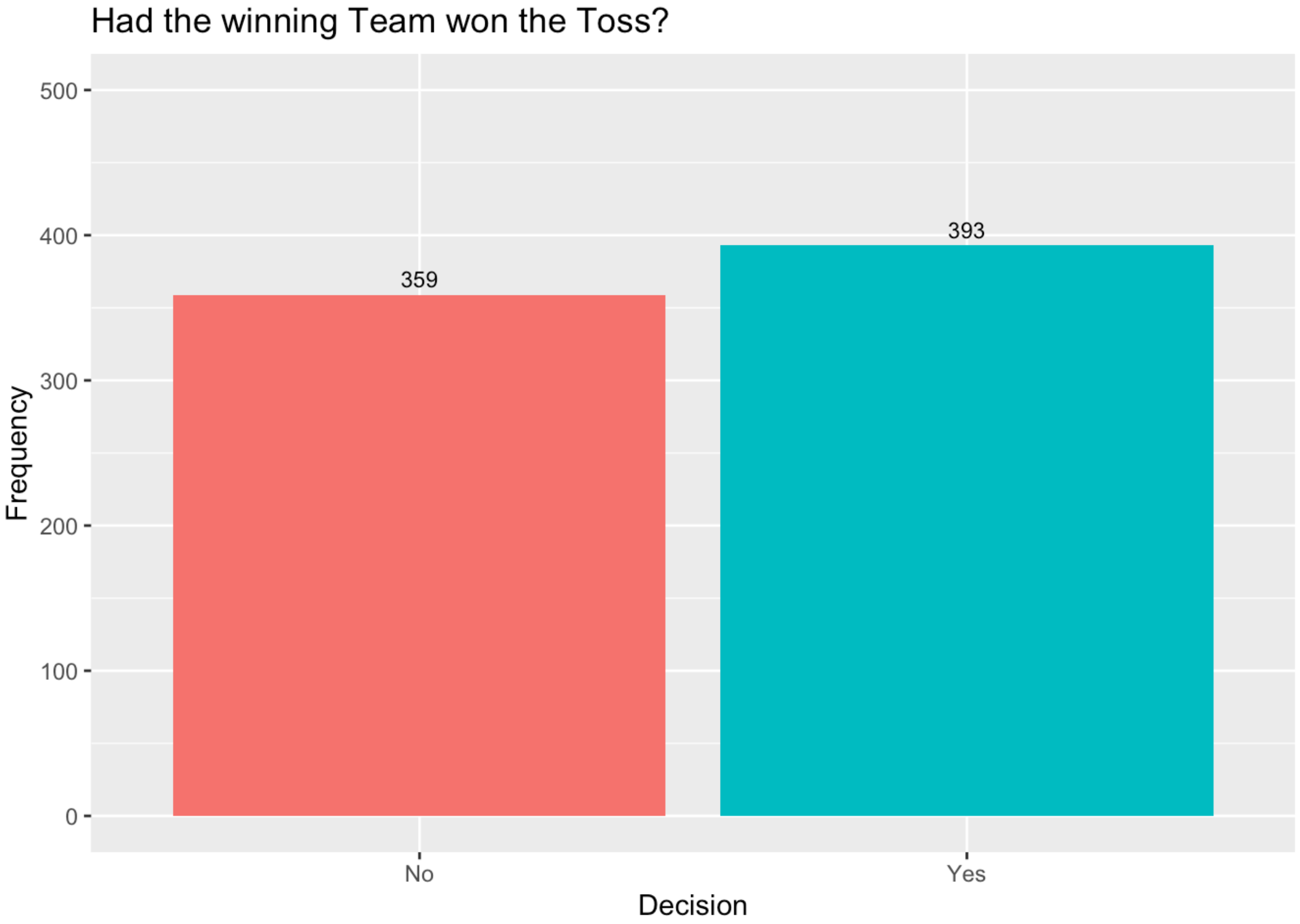
Let's check the number of times the teams have won the toss throughout the years in the tournament.



The dotplot shows that Mumbai Indians have won the maximum number of tosses compared to the other teams in the tournament. Kolkata Knight Riders rank second with Delhi Capitals ranking third followed by Chennai Super Kings and Royal Challengers Bangalore. Mumbai has been the most successful team in the tournament and has also won the most number of matches along with winning the most number of tosses. This goes with the other 3 teams as well because overall, the top 4 teams (Mumbai, Chennai, Kolkata, and Bangalore) that have won the most number of tosses are also the teams that have won the most number of matches. This suggests that a relation may exist between winning the toss and winning the match.

Let’s look at this problem from a broader perspective by seeing how many times teams have won games after winning the toss.

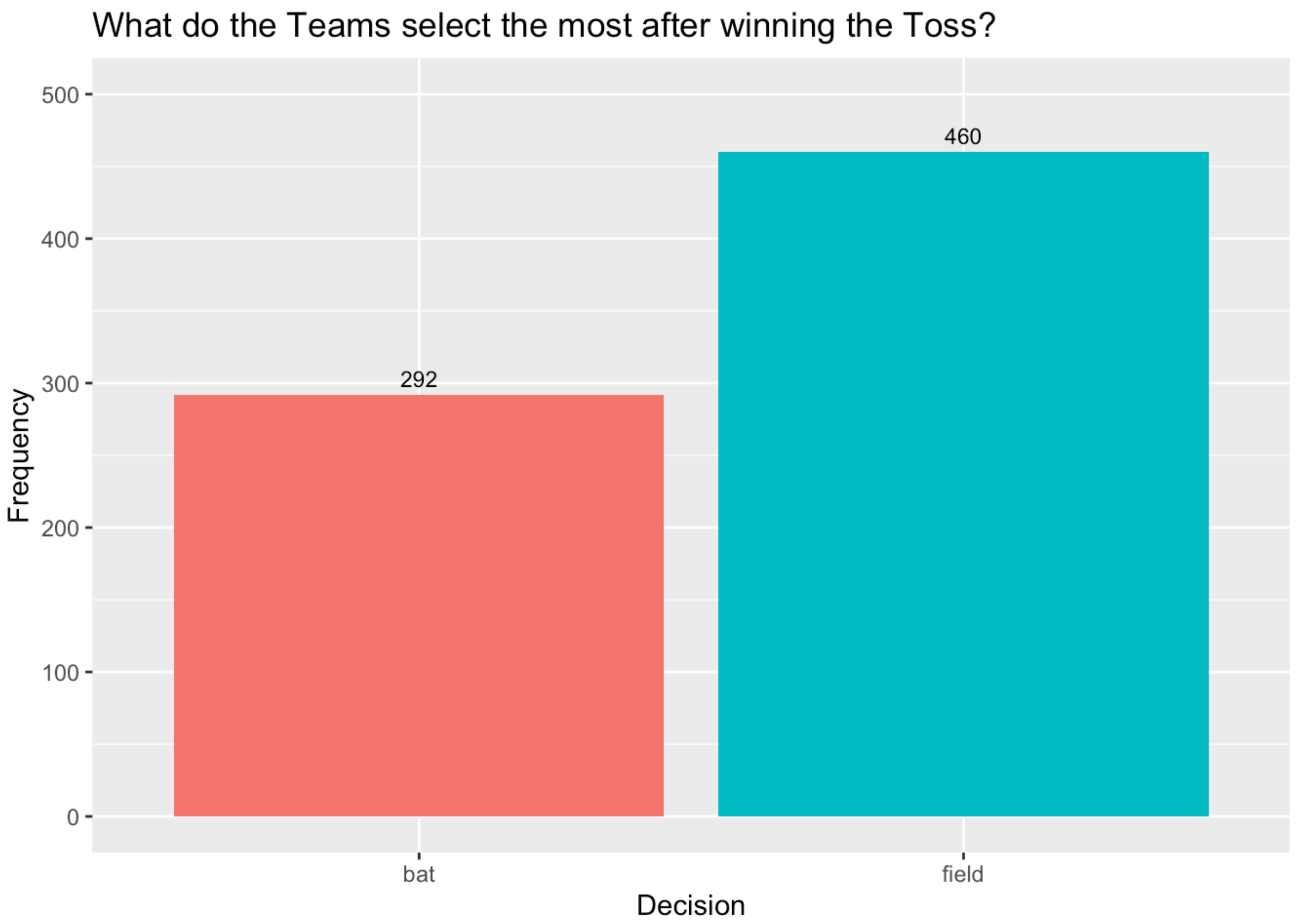
How many times did the teams, that won the match, win the toss?



The bar graph above shows how many times the teams have won or lost the matches after winning the toss. Out of 752 matches played, teams have won the match 393 times after winning the toss which suggests that there is a 52.3% chance of winning the match after winning the toss according to these results. We can see that there is a slight advantage in winning the toss when it comes to winning matches. However, this advantage doesn't seem to be enough as teams have lost the matches 359 times despite winning the toss as seen in the bar graph. Hence, it wouldn't be correct to conclude that winning the toss results in winning the match but yes, there is a very slight advantage in winning the toss when it comes to winning the match.

What do the Teams select the most after winning the Toss? Batting or Fielding?

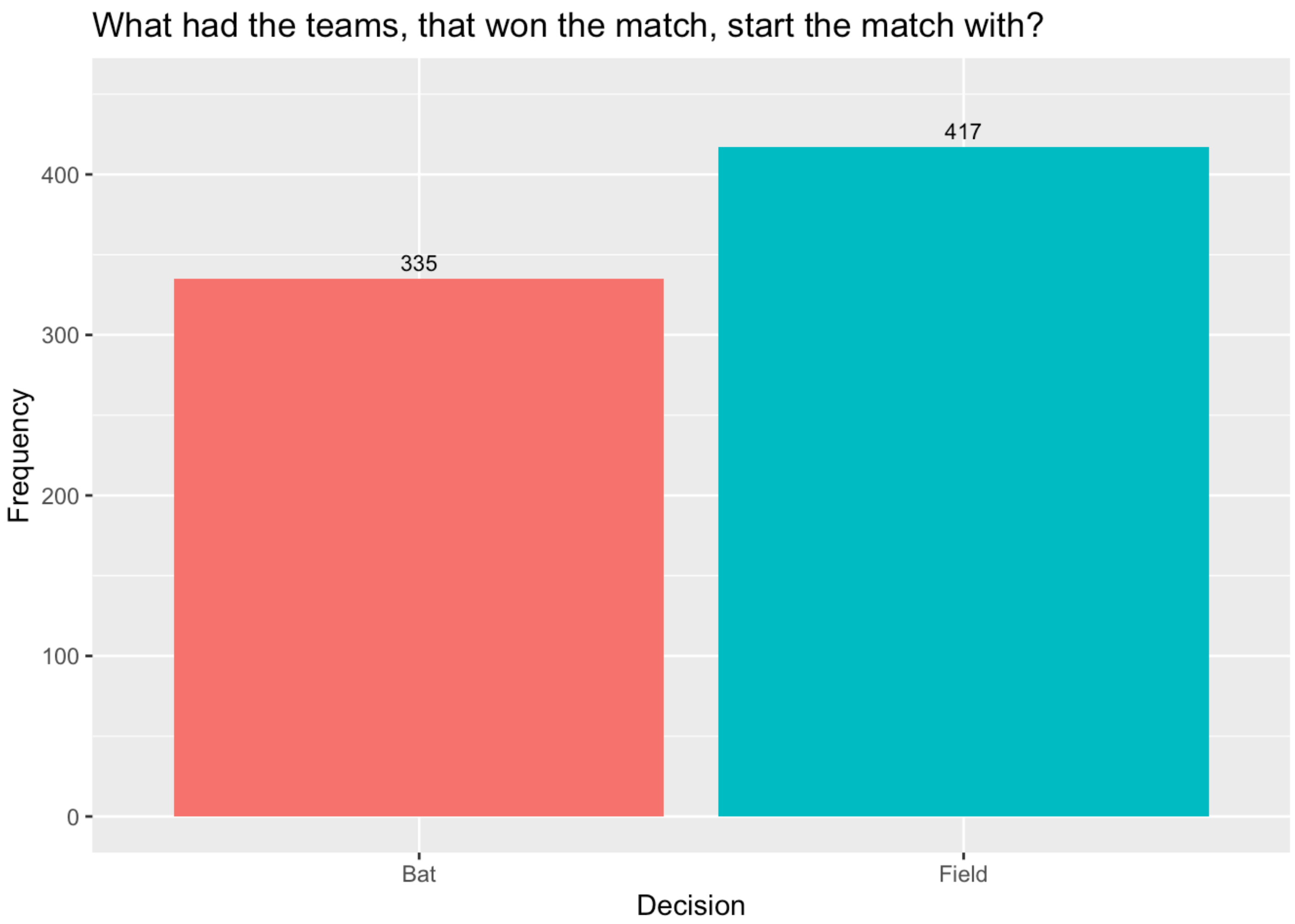
Moving on, let us see how a toss decision affects the outcome of a match. Several captains in the game of cricket tend to favor fielding first as it lets a team know what the target is and hence, it gets easier to play the game with a calculated requirement. We are interested in finding out whether a specific toss decision results in winning more games or not. Therefore, let's check what the teams decide to choose the most after winning the toss.



As seen above, teams in the IPL through the years have chosen to field first more than bat first after winning the toss. Teams have chosen to field first 460 times out of the 752 matches in the dataset. This indicates that teams choose fielding first 61.2% of the time after winning the toss which is a higher percentage than choosing to bat which was done 292 times by the teams. This shows that teams favor to field and chase first more than bat first and set a target for the opponent.

What had the teams, that won the match, start the match with? Batting or Fielding?

Let’s take a look at what the outcome of fielding or batting is when it comes to the result of the game. Do teams win more if they field first or do they win more when they bat first? Let’s visualize the decisions of the winning teams via a bar graph to see whether they fielded or batted first.



The above bar graph indicates that the teams that won the match had fielded first 417 times on one hand and had batted first 335 times on the other hand. This means that the teams that won matches had fielded first 55.5 % of the time which is more than the percentage of winning the game when batting first. Even though this is not that big of a difference, we can say that fielding first in the game, especially in the IPL, has had a slightly higher success rate in being victorious. Moreover, compiling this with the other visualizations that we observed above, we can deduce that the teams prefer to field and chase a given target more because that slightly increases the team's chances of winning the match and being more successful overall.

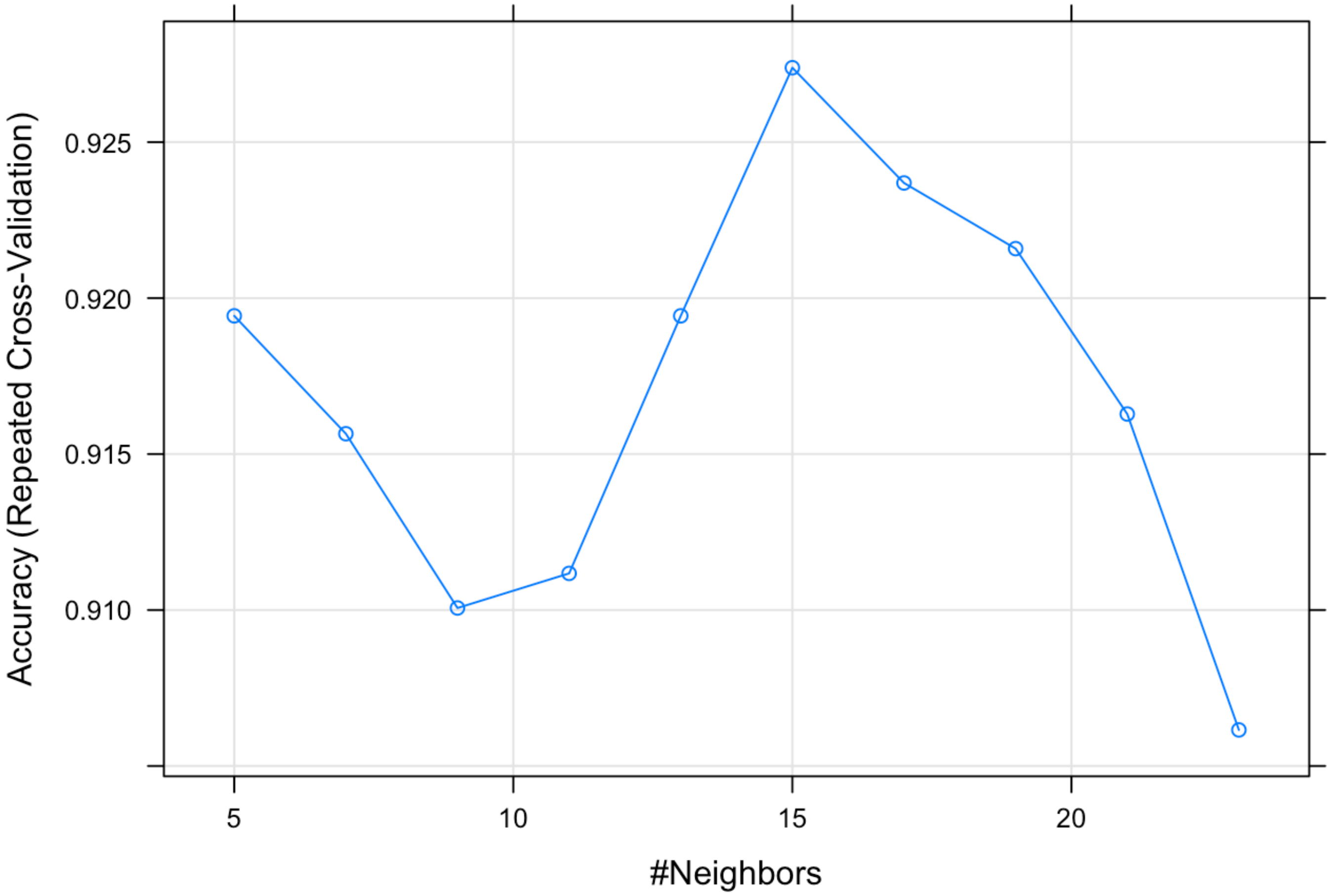
Different conditions affect how a match is played by two teams. Few of these conditions are the type of the pitch that the game is being played on, the ground and the city the match is being played in, and the fact that a team is bowling or batting first based on the conditions of the ground and the pitch. We have seen in several cases how a team has won more matches by winning the toss and how more teams have won by bowling first compared to batting. A toss and its decision matter because it gives a team an advantage regarding how they want to start the game. We wanted to classify toss decisions based on a few features of our dataset like the city/ground the match was played on, the magnitude by which a team won the match by wickets or by runs, and the previous toss decisions the teams had taken. We also wanted to find out whether the teams that had won the match had batted first or fielded first. Moreover, finding out whether the team that had won the match had won the toss as well or not would also be very helpful in determining the toss decisions. We made these variables and visualized them above for better understanding. As a result, we used these variables to classify and predict the toss decisions that would be the best to take in order to increase the chances of victory in the match.

Modelling and Results

First, we split the data into 60% for the training set and 40% for the testing set. We performed this split as we had a small dataset and we wanted a good amount of data to test our model on so that we could get better and more reliable results so choosing 40% for the testing set seemed to be a good fit to understand how effective our models would be. Second, we decided on using a few models on our data like the K-Nearest Neighbors model, the Decision Tree model, and the Random Forest model to train and test our data. After performing these models on our dataset and computing metric values like the accuracy, precision, and a confusion matrix, we concluded our results which can be seen towards the end of the project.

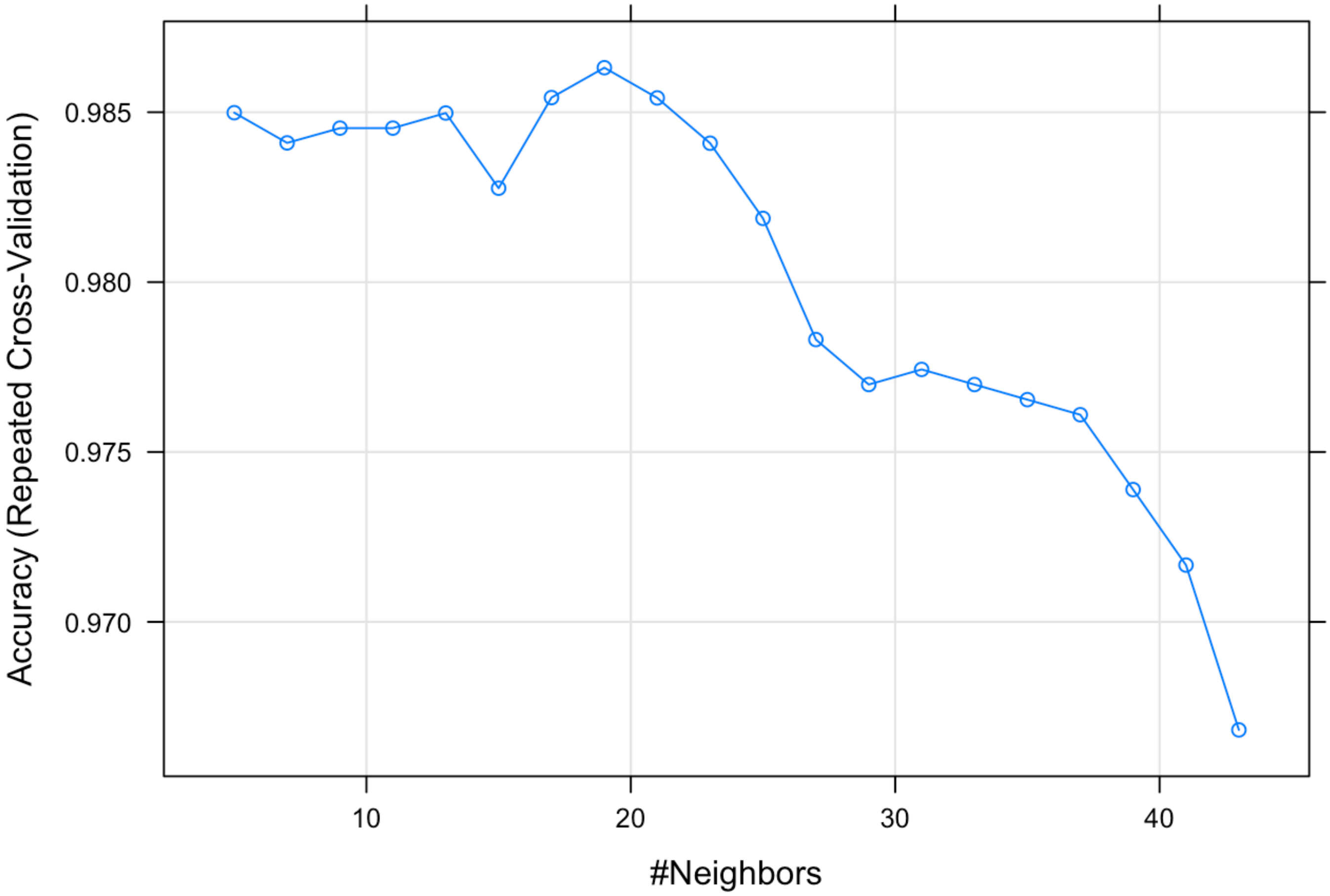
KNN

K-Nearest Neighbors was selected for its classification purposes as a simple and standard supervised classification machine learning model. We fit the model on the entire dataset first to check what kind of results we could be expecting. Our predictor variable was the ‘Winner_decision’ which indicated whether the team that had won the match had fielded first or batted in all the matches. As the outcome of this variable was either ‘Bat’ or ‘Field’, we could make it a binary classifier by making this variable a factor as this would make our modeling easier. Our predictor variables consisted of the ‘Toss_decision’ which showed what the team had chosen (batting or fielding) after winning the toss, ‘Team_won_toss’ which indicated whether the team that had won the match had won the toss or not, ‘City’ that showed which city and hence which ground the match was played on, ‘win_by_runs’ which showed how many runs the team had won by in the match and finally ‘win_by_wickets’ which showed how many wickets the team had won by in all the matches. As discussed above, we felt these variables are what make a difference in real-life toss decisions in cricket matches and we wanted our model to learn from the history of these variables as well and give us accurate results. With running the KNN model, we also ran the K-fold cross-validation with five repeats and ten folds to determine what the best number of K would be on our entire data set. After running the model, we received some really high accuracy scores, mostly around 92%. K = 15 gave us the highest accuracy score as shown in the following graph and it was chosen to be the optimal model for the dataset with an accuracy of 92.74%.

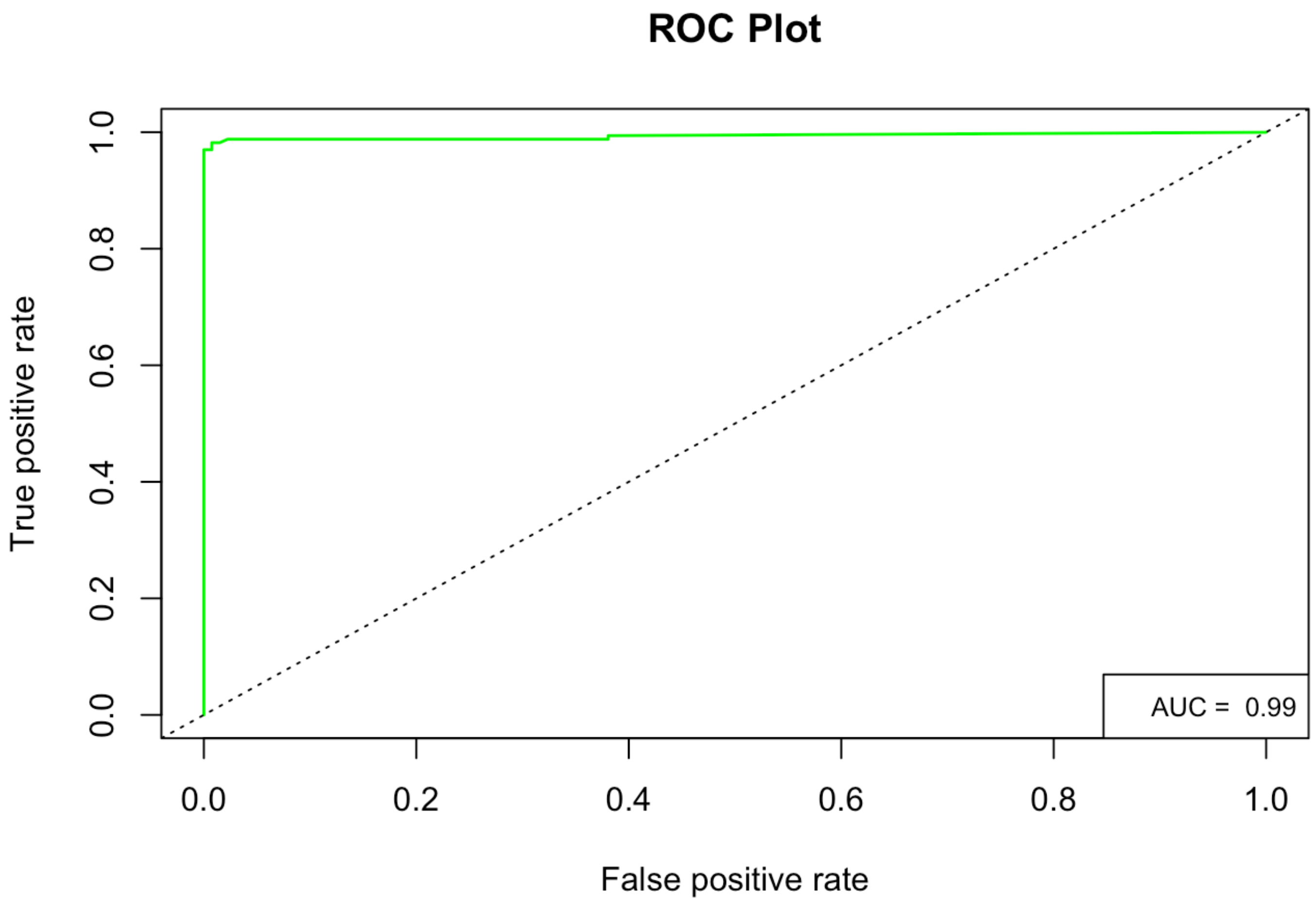


Confusion Matrix and Accuracy Plot of the Model run after the data split

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Bat Field
##      Bat   134     5
##      Field   0   161
##
##                Accuracy : 0.9833
##                  95% CI : (0.9615, 0.9946)
##     No Information Rate : 0.5533
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.9664
##
##  Mcnemar's Test P-Value : 0.07364
##
##             Sensitivity : 1.0000
##             Specificity : 0.9699
##          Pos Pred Value : 0.9640
##          Neg Pred Value : 1.0000
##              Prevalence : 0.4467
##          Detection Rate : 0.4467
##    Detection Prevalence : 0.4633
##       Balanced Accuracy : 0.9849
##
##        'Positive' Class : Bat
##
```



ROC Plot and AUC

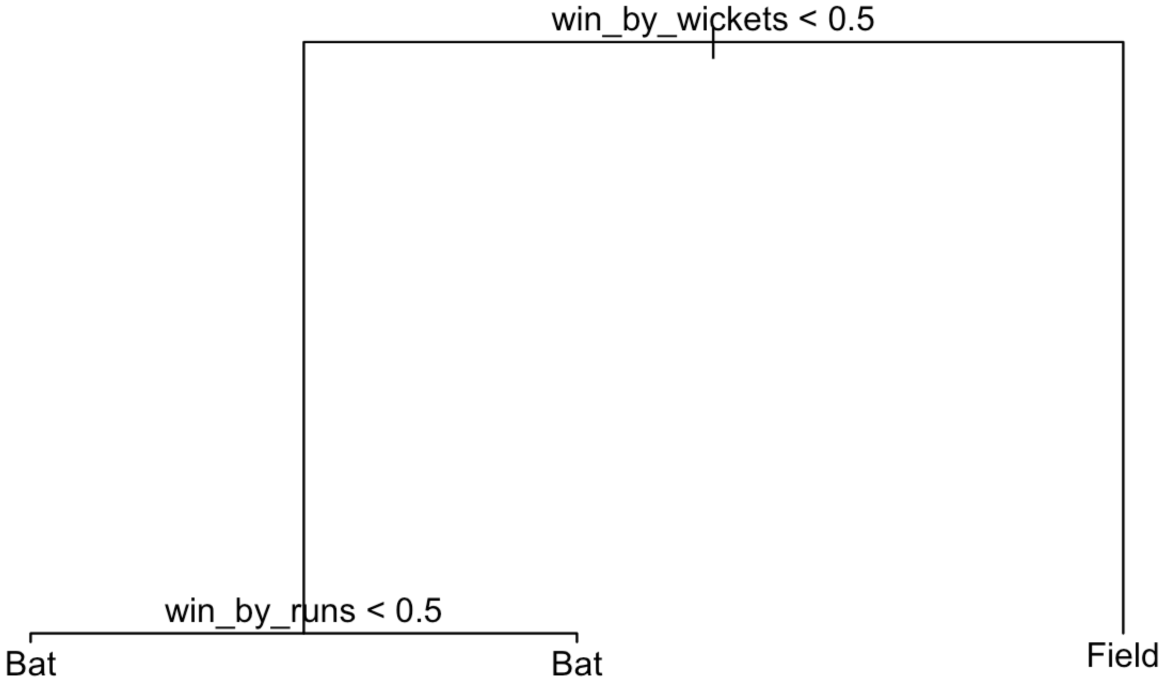


After trying and testing the model on the dataset and receiving acceptable results, we decided to use the model on the training and testing sets. By using the same predicted variable and predictor variables, we modeled another variable with knn but using the training dataset this time. We again ran K-Fold cross-validation as well five times with ten folds in order to determine the best number of neighbors and the optimal model. As the training set was smaller and 60% of the entire dataset, we received even higher accuracy compared to the previous results falling around 98%. The best k was computed to be 19 as that model provided us with the highest accuracy of 98.63%. This can be seen in the accuracy plot above. The model was then used to make predictions on the testing dataset which was 40% of the entire dataset. A confusion matrix and a few other metrics were output to judge the model and its results. The model had an overall accuracy of 0.9833 which was really high. The model produced a classification rate with just five misclassifications and the rest were correctly classified which was great and can be seen through the confusion matrix above. The model also had a Sensitivity of 1.0 and a Specificity of 0.9699 which showed that it performed well. Moreover, an AUROC was also plotted for the results and the curve had an area of 0.99 which further supported our results and showed how the model was able to learn well from the predictors and predict what was the best decision to take, fielding or batting, in order to increase the chances of victory. This could help teams decide which decision to go ahead with in order to maximize the chances of winning as even if another team won the toss and chose to field when the model predicted to bat first, it could help the team be satisfied with getting the decision to bat first as that would have a higher chance of winning them the match.

Decision Tree

The next model chosen was Decision Trees in order to gain an insight into which features of the dataset may be of more importance in splitting the data. Since decision trees split the data on binary nodes of the tree, it can easily be visualized how the model does feature importance ranking and prioritization, enabling the ability to clearly comprehend which features lead to certain classification outcomes. The tree was modeled using the same variables that were used in the KNN model.

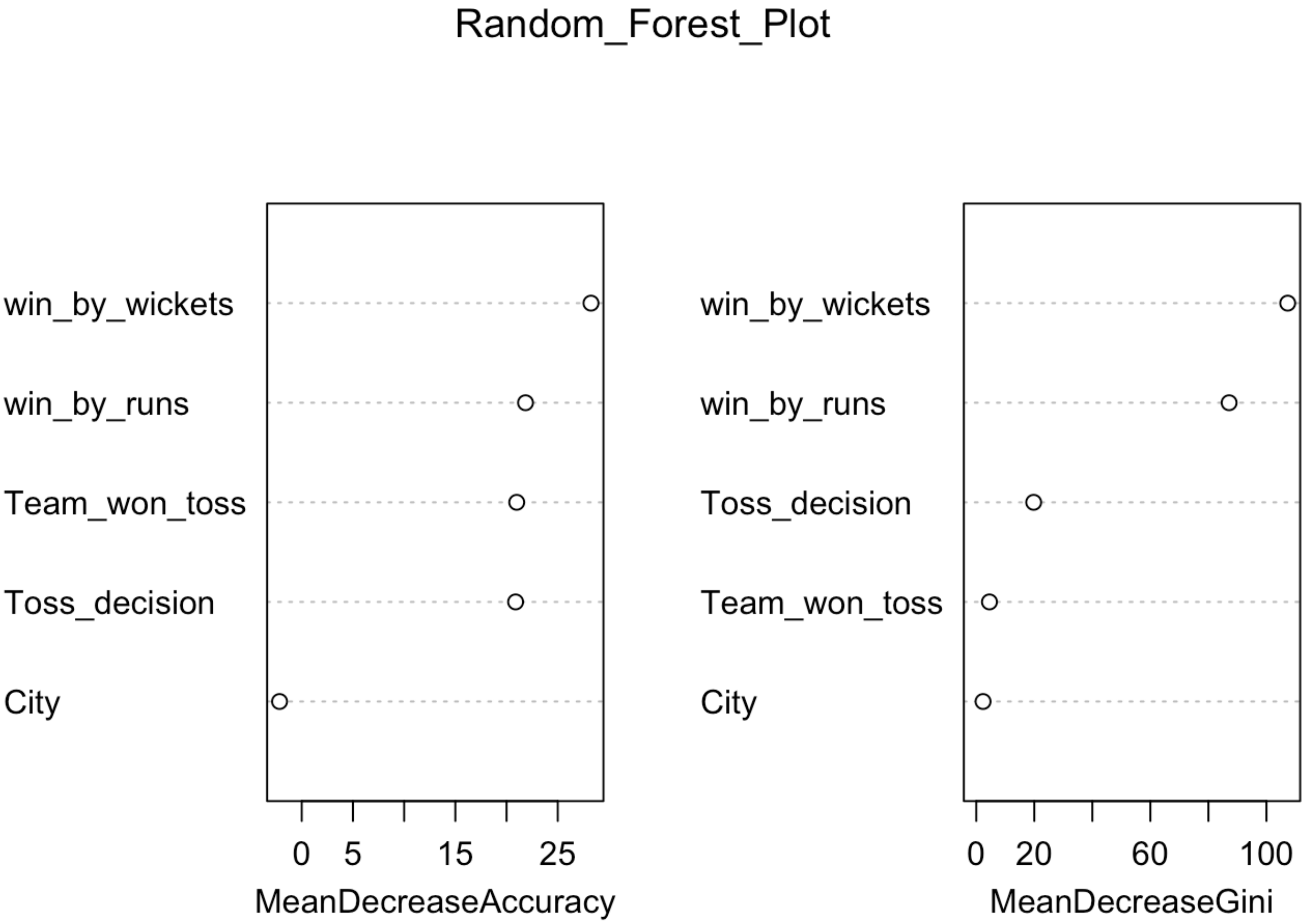
```
##
## Classification tree:
## tree(formula = Winner_decision ~ Toss_decision + Team_won_toss +
##     City + win_by_runs + win_by_wickets, data = train.set)
## Variables actually used in tree construction:
## [1] "win_by_wickets" "win_by_runs"
## Number of terminal nodes:  3
## Residual mean deviance:  0.1025 = 46.03 / 449
## Misclassification error rate: 0.01327 = 6 / 452
```



A simple tree model was output with two splits performed on two variables. The first variable used to split the dataset was ‘win_by_wickets’ and the second variable was ‘win_by_runs’. If ‘win_by_wickets’ was greater than 0.5 (which means 1 or more in the sense of the game as wickets can only be in whole numbers) then the model would recommend choosing to field, else the model would check the win_by_runs and recommend choosing to bat instead. The misclassification rate of this model was 0.01327 and the residual mean deviance was 0.1025. However, this model did not perform the way it was expected to. The model learned and only used ‘win_by_wickets’ and ‘win_by_runs’ from the training set and did not rank the other variables as important, which we felt was not right as other variables would matter more in the real world. Judging the classification from the runs or wickets won by would not give us very reliable results if a match was going to be played on new ground as that ground would have no history of matches being played. The city the match is played in and the previous toss decisions or whether the team that had won had fielded or batted first should ideally hold a lot of weight while making the decision but the model did not weigh them. This summarizes how even though our model ran and split the data correctly, the outcomes weren’t great and they were not what was expected. Hence, the results from the decision tree model were not very helpful.

Random Forest

The final model chosen was the Random Forest model. The random forest implements an ensemble of multiple decision trees in order to avoid the possible overfitting or bias of a single decision tree. This would enable us to compare the outputs of the decision tree with those of the random forest model. The Random Forest was also modeled using the same variables that were used in the KNN and the Decision Tree model with ‘Winner_decision’ being the predicted variable and ‘Toss_decision’, ‘Team_won_toss’, ‘City’, ‘win_by_runs’, and ‘win_by_wickets’ being the predictors. The model was fit on the training set and the predictions were given on the testing set. A confusion matrix along with a feature importance table was output to judge the model and its results.



The confusion matrix showed that the model gave seven incorrect classifications, however, the rest were all correctly classified. In the feature importance table, ‘win_by_wickets’ was given the most importance in the model which was followed by ‘win_by_runs’. The model ranked ‘Team_won_toss’ and ‘Toss_decision’ next respectively and showed that these variables carried importance too. However, the model ranked the ‘City’ variable last and showed that it carried very little or no importance while predicting the variable. The random forest model again ranked ‘win_by_runs’ and ‘win_by_wickets’ as the two most important variables like the Decision Tree model did which again showed us how the model wasn’t very great and did not perform as expected. Nevertheless, the model showed how variables like the previous toss decisions and the winner team’s decision to bat or field carried importance which could help us with our conclusion and interpretation of the results. We expected the latter variables to play a larger role in the predictions but that was not the case after looking at our model. Even though the model mostly classified the variables correctly with high accuracy, it would have been better if the model used the ‘Toss_decision’, ‘Team_won_toss’, and ‘City’ variables more which would give us a much better conclusion compared to a real world scenario.

Discussion and Conclusion of the Results

Overall, all our models gave us a high accuracy with very less misclassifications which was great. However, the variables used by the models weren’t the ones which we thought would get more importance as we wanted to base our conclusions more on whether it would be better to bat first or field first for a particular match against an opponent based on the ground history, previous toss decisions and match outcomes more than the history of winning by runs and wickets in the tournament as that would realistically be more helpful.

This raised a rather profound question regarding what would the model perform like if the two variables were removed from the dataset. The answer is that the model would perform but be less accurate especially compared to the current performance. There would be more incorrect classifications and the accuracy would be affected as well. The reason why our model performed the way it performed, ranking the win by runs or wickets higher than the other variables was because the data set was small and had less information about the ground in general. We only had the ground and city name as the information and the toss decisions and outcomes which we felt could not predict the best decision to choose in a match very accurately. If we had more data regarding specific ground conditions, pitch conditions, and its type, we could probably come up with better conclusions and base our predictions more on such variables than the runs and wickets won by. We could analyze how the ground size, pitch texture, weather type, amount of dew on the ground and the wind would affect matches and that would assist us to come up with more realistic predictions and conclusions. However, with the dataset we were given, this is the best the models could perform. The Random Forest model gave slightly more misclassifications than the KNN model which suggested that the KNN model performed better and was more accurate. On the other hand, the Random Forest model ranked 'Toss_decision' and 'Team_won_toss' important as well along with the 'win_by_runs' and 'win_by_wickets' variables. This is something the Decision Tree did not do as it only used 'win_by_runs' and 'win_by_wickets' to split the dataset. The random forest model considered variables like the previous toss decisions, outcomes, and the winning team's decision to bat or field to carry weight and importance along with the other variables which made us conclude that the Random Forest model made better use of variables than the Decision Tree and was more useful for our conclusion and results than the Decision Tree. At last, we feel that as the KNN model performed with the highest amount of accuracy and had the lowest amount of misclassifications, we would choose that model to predict what would be the best decision for a team to go ahead with, batting or fielding. We felt that the results from the KNN model were the most helpful for our conclusion.

Lastly, as we saw numerous visuals along with the information provided above, we realized how the successful teams throughout the tournament are the ones who obviously won the most number of matches but also the most number of tosses. The tournament also hosted the most number of matches in the cities that these teams belonged to which showed us a lot of reasons why a team might be very successful in the tournament. However, when it came to whether winning tosses helps teams in winning matches, we saw how teams that had won the match had won the toss 52.3% of the time. Even though this number was slightly higher, we could not conclude whether winning tosses definitely helped in winning matches as the numbers were not very different. On the other hand, our second question was to find out which specific toss decision helped win matches. Through the plots, we saw how out of the teams that had won the matches, they had fielded first (55.5% of the time) more than they batted which showed how fielding first did result in being more victorious in the game of cricket in the IPL. Furthermore, the plots showed through the years how the teams favored choosing to field first more than batting as they thought they would have a higher chance of winning the game by chasing the target. This could help us form an opinion about how even though winning the toss did not change the outcome of winning the match by a lot, fielding first in the game did help win games slightly more. Although, the decision of choosing fielding or batting in a particular match depends on several factors as discussed before like the ground condition, pitch type, dew factor, the history of wins and losses, and their margins on the pitch to name a few.

References

1. Sports, T. O. (2022, March 25). Which is the Best Team in IPL History? - Decoded the Data. Times of Sports News Today, Latest Headlines Updates, Live Match Score. https://www.timesofsports.com/cricket/ipl/which-is-the-best-team/ (https://www.timesofsports.com/cricket/ipl/which-is-the-best-team/)
2. Sports, T. O. (2022, March 25). Which is the Best Team in IPL History? - Decoded the Data. Times of Sports News Today, Latest Headlines Updates, Live Match Score. https://www.timesofsports.com/cricket/ipl/which-is-the-best-team/ (https://www.timesofsports.com/cricket/ipl/which-is-the-best-team/)
3. P., B. (2021, December 16). Exploratory Data Analysis of IPL Matches-Part I - Towards Data Science. Towards Data Science. https://towardsdatascience.com/exploratory-data-analysis-of-ipl-matches-part-1-c3555b15edbb (https://towardsdatascience.com/exploratory-data-analysis-of-ipl-matches-part-1-c3555b15edbb)
4. Wikipedia contributors. (2022, April 20). List of Indian Premier League seasons and results. Wikipedia. https://en.wikipedia.org/wiki/List_of_Indian_Premier_League_seasons_and_results (https://en.wikipedia.org/wiki/List_of_Indian_Premier_League_seasons_and_results)
5. Nguyen, C. (2022, January 5). Guide To Data Visualization With ggplot2 - Towards Data Science. Towards Data Science. https://towardsdatascience.com/guide-to-data-visualization-with-ggplot2-in-a-hour-634c7e3bc9dd (https://towardsdatascience.com/guide-to-data-visualization-with-ggplot2-in-a-hour-634c7e3bc9dd)
6. Wickham, H. D. N. (n.d.). 15 Scales and guides | ggplot2. Elegant Graphics for Data Analysis. Retrieved May 11, 2022, from https://ggplot2-book.org/scales-guides.html (https://ggplot2-book.org/scales-guides.html)