Data Analysis of the Indian Premier League

Introduction

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

About Cricket:

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

against their opponent. These days, data science techniques are being made use of by every team that competes in the sport professionally. When used correctly, it can help teams bridge the gap in skill by formulating an effective strategy to counter their opponents.

About the Indian Premier League:

The Indian Premier League (IPL) is the world's biggest domestic cricket tournament. It is a 20-over format of the game that makes for short, fast-paced games which is one of the reasons for its massive fanbase. It is an annual tournament and has seen 14 such tournaments conducted so far. There are 9 teams involved in the tournament and the teams themselves consist of players from all around the world. The tournament generates a large revenue and has many stakeholders heavily invested in it. So teams will do everything they can to get an edge over their opponents in a game. Data Analysis is now heavily used by all teams to try and gain this edge.

Datasets

Source: https://www.kaggle.com/nowke9/ipldata?select=matches.csv

Content:

- Data till Season 11 (2008-2019)
- matches.csv - Match by match data
- deliveries.csv - Ball by ball data

Both the datasets are linked by the 'id' column of matches.csv and 'match_id' column of deliveries.csv which represent the matches uniquely.

Columns of matches.csv:

'id', 'season', 'city', 'date', 'team1', 'team2', 'toss_winner', 'toss_decision', 'result', 'dl_applied', 'winner', 'win_by_runs', 'win_by_wickets', 'player_of_match', 'venue', 'umpire1', 'umpire2', 'umpire3'.

Columns of deliveries.csv:

'match_id', 'inning', 'batting_team', 'bowling_team', 'over', 'ball', 'batsman', 'non_striker', 'bowler', 'is_super_over', 'wide_runs', 'bye_runs', 'legbye_runs', 'noball_runs', 'penalty_runs', 'batsman_runs', 'extra_runs', 'total_runs', 'player_dismissed', 'dismissal_kind', 'fielder'.

These two datasets are used for all the tasks. After Data Pre-processing, EDA is performed on the datasets separately. For predictive tasks, the datasets are combined. Further, to perform various tasks more efficiently, copies of the datasets are created and modified accordingly.

Data Pre-processing and EDA

Data Pre-processing:

The datasets matches.csv and deliveries.csv can not be used directly for predictive analysis. There are some problems like missing values, duplicate team names, insignificant columns etc. Also, for predictive analysis, the data must contain only numerical values for the models to work. Hence, some steps for data pre-processing are needed.

First, the null values of the 'city' column of matches.csv are filled based on the values of the 'venue' column. The 'city' values were empty where 'venue' was 'Dubai International Cricket Stadium', so the 'city' values are filled as 'Dubai'.

The datasets contain duplicate names for two teams. One team has names 'Rising Pune Supergiants' and 'Rising Pune Supergiant' and another has 'Delhi Daredevils' and 'Delhi Capitals'. The names for the teams are changed to 'Rising Pune Supergiant' and 'Delhi Capitals' respectively in the columns 'team1', 'team2', 'toss_winner', 'winner' of matches.csv and 'batting_team' and 'bowling_team' of deliveries.csv.

The datatype of the 'date' column of matches.csv is changed to 'DateTime' from 'object'.

The columns 'umpire3' of matches.csv and 'player_dismissed', 'dismissal_kind' and 'fielder' of deliveries.csv have a large number of missing values. Hence, they are removed.

All the rows of matches.csv where 'winner' values are missing are removed.

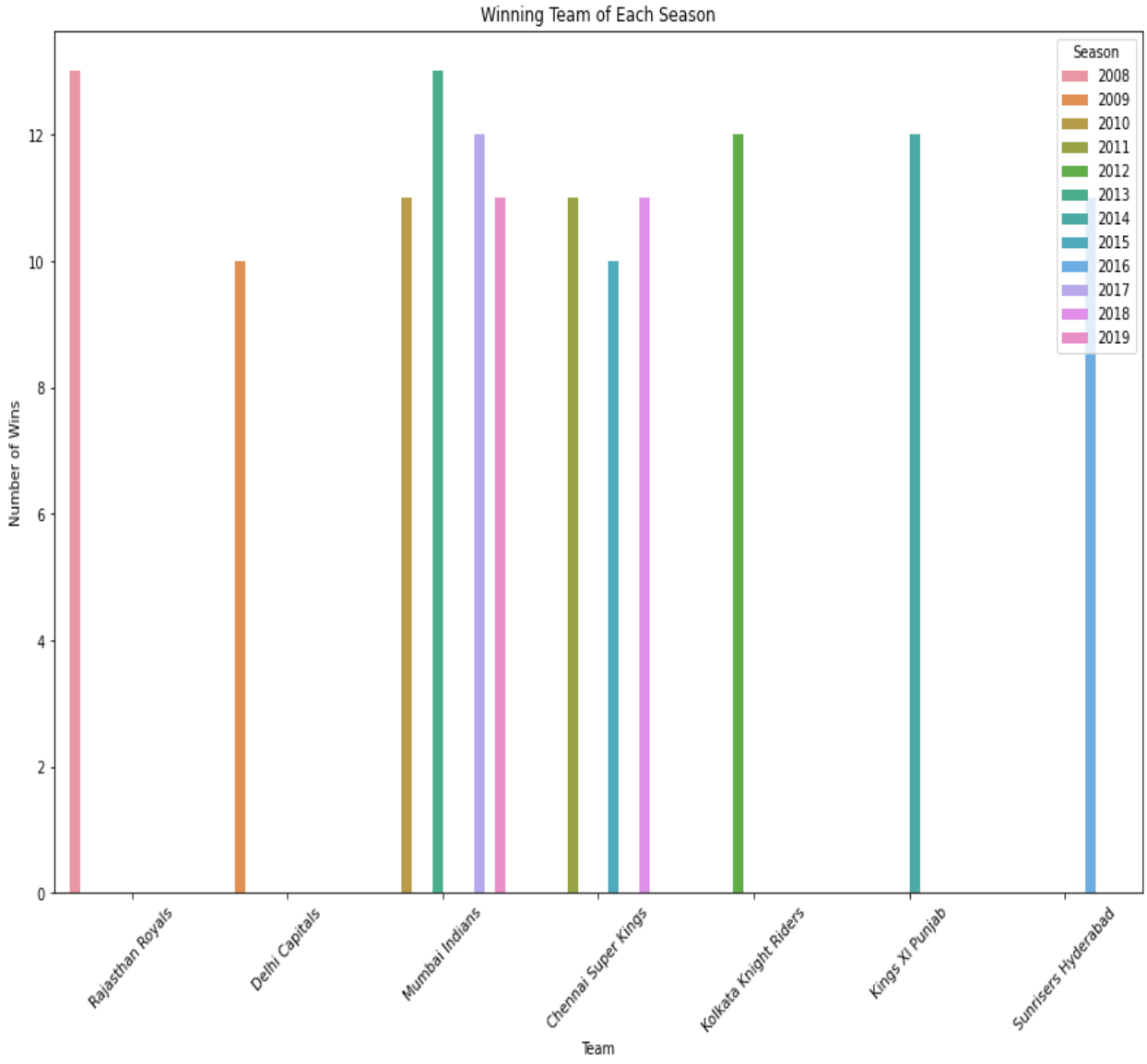
There are still some null values in a few columns of matches.csv. These are filled with '.' as the columns are of 'object' data type.

Before performing the predictive tasks, the columns with 'object' data type are changed to numerical using label encoding. To bring the data to scale, StandardScaler is used.

Inferences from EDA:

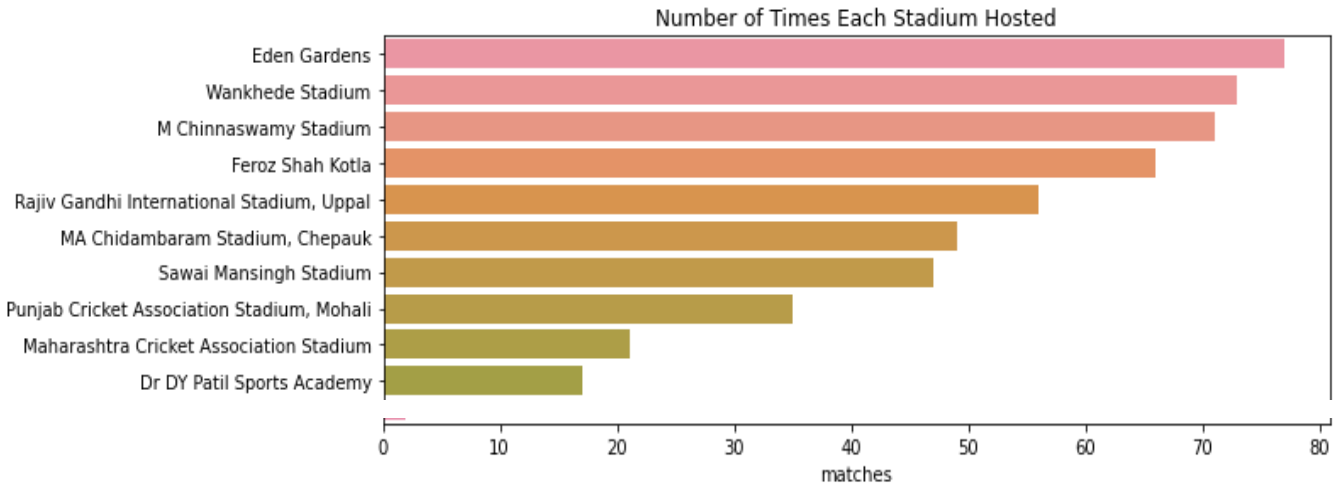
Various manipulations are performed on the available datasets to extract some insightful information from them.

The team with most number of wins per season:



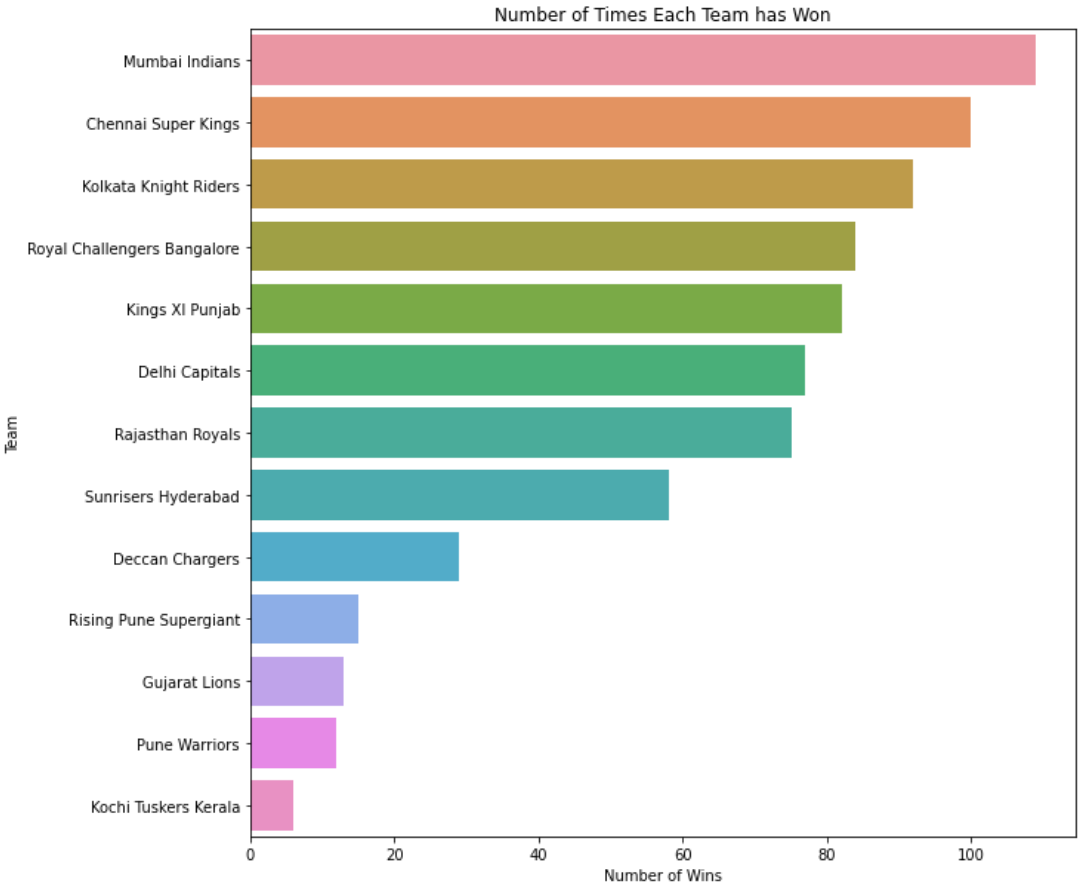
The team Mumbai Indians has secured the most wins, in four seasons (2010, 2013, 2017 and 2019) followed by the team Chennai Super Kings with wins in three seasons (2011, 2015 and 2018).

Top hosting venues:

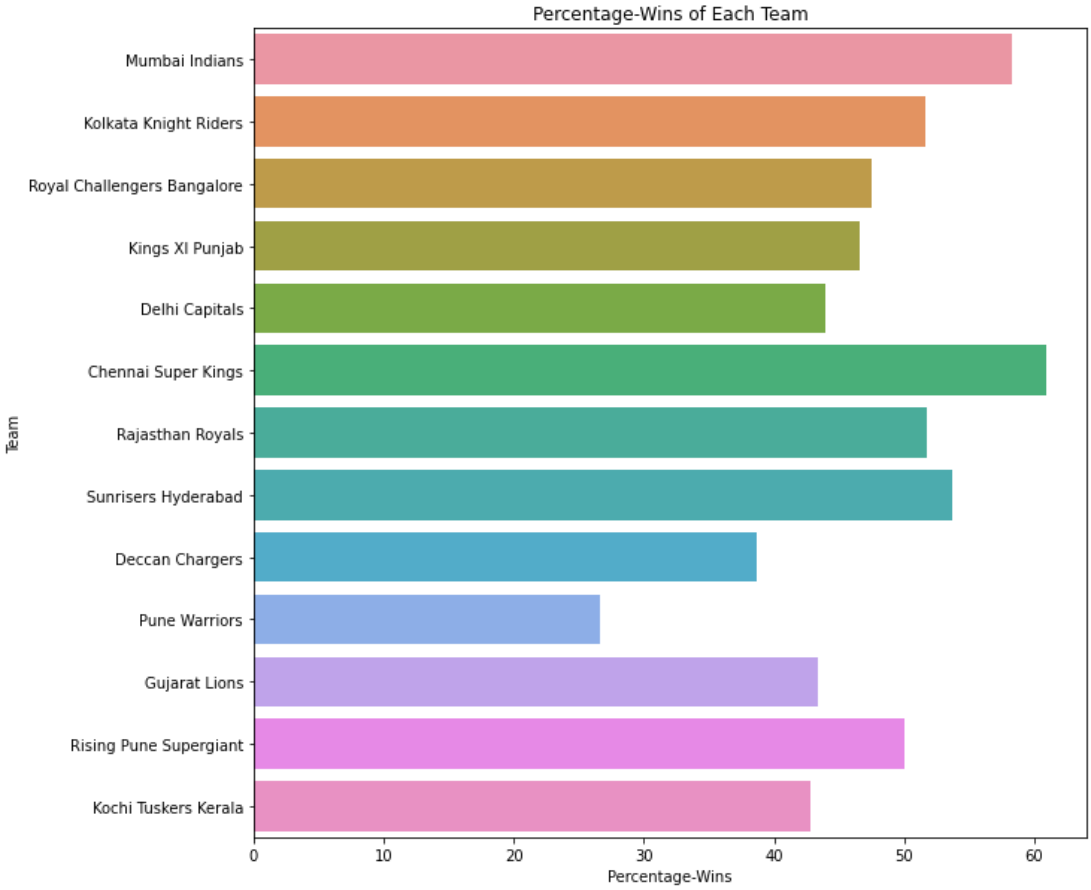


The venue Eden Gardens has hosted the most number of matches, 77, followed by Wankhede Stadium that hosted 73 matches.

The Most Successful IPL Team:

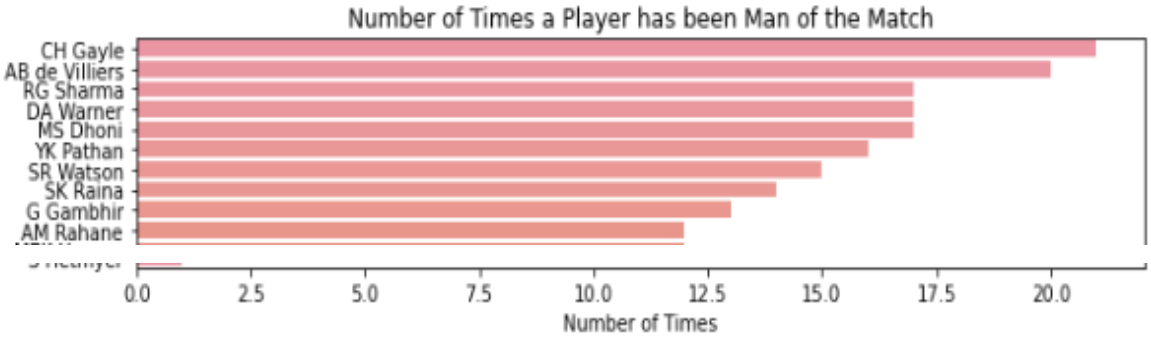


According to this plot, the team Mumbai Indians is the most successful team of IPL followed by Chennai Super Kings. However, not all teams have played every year. Hence, a better way to find the most successful team is to compare the percentage wins of each team, i.e., the percentage of the number of wins against the total number of matches played by the team.



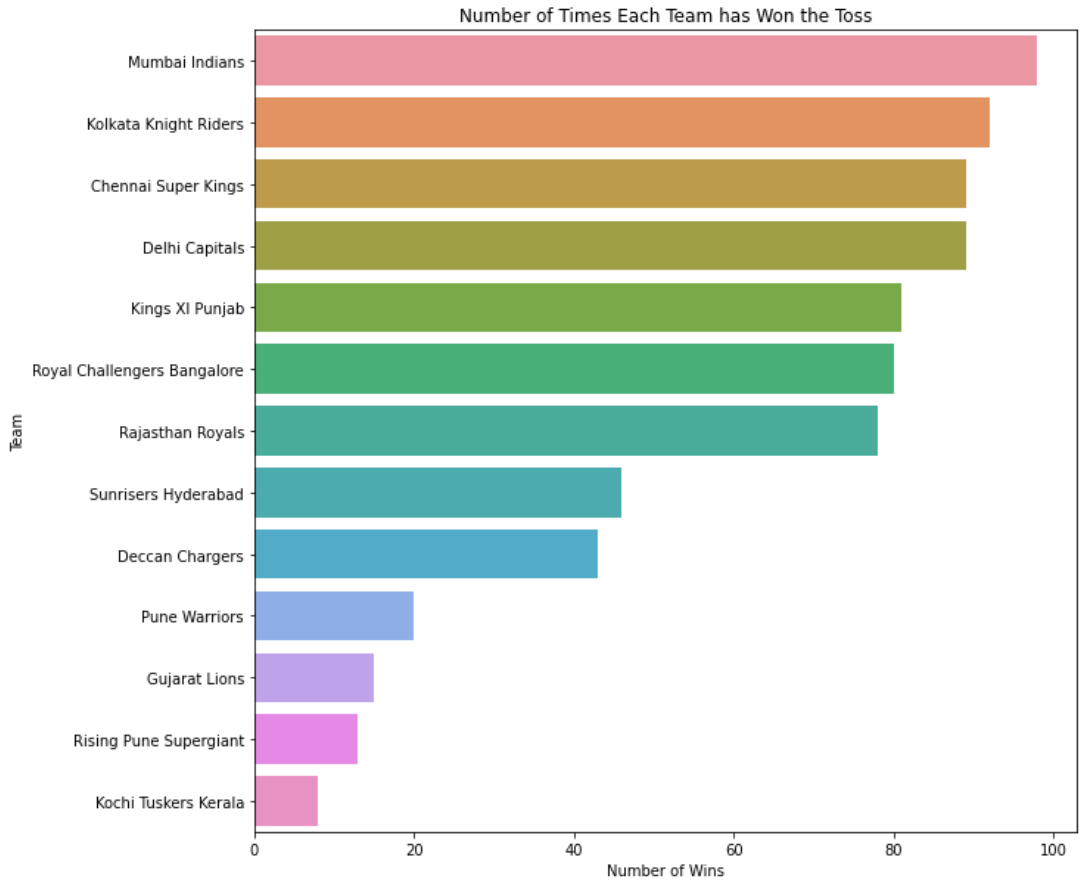
According to this plot, the most successful IPL team is Chennai Super Kings followed by Mumbai Indians.

Most Valuable Player:



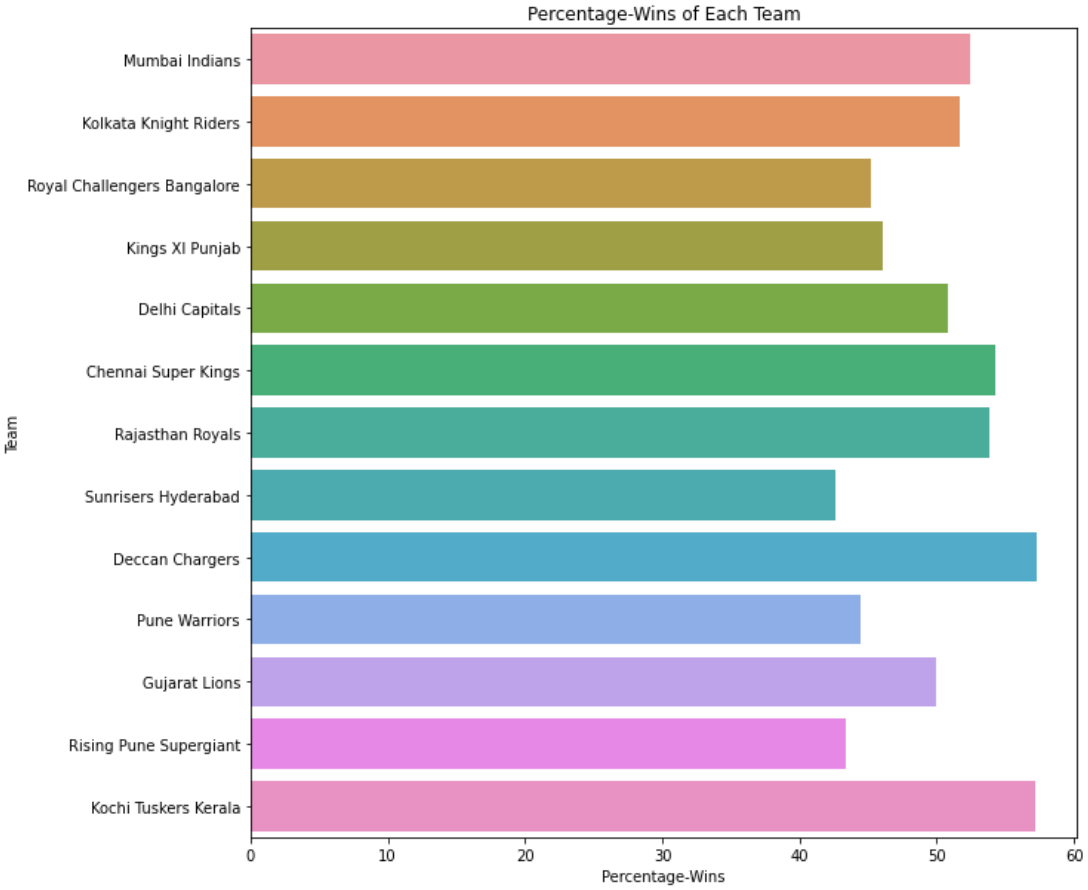
From the plot, we infer that the most valuable player of IPL is CH Gayle, who has been the player of the match 21 times followed by AB de Villiers, who has been the player of the match 20 times.

The Luckiest IPL Team:



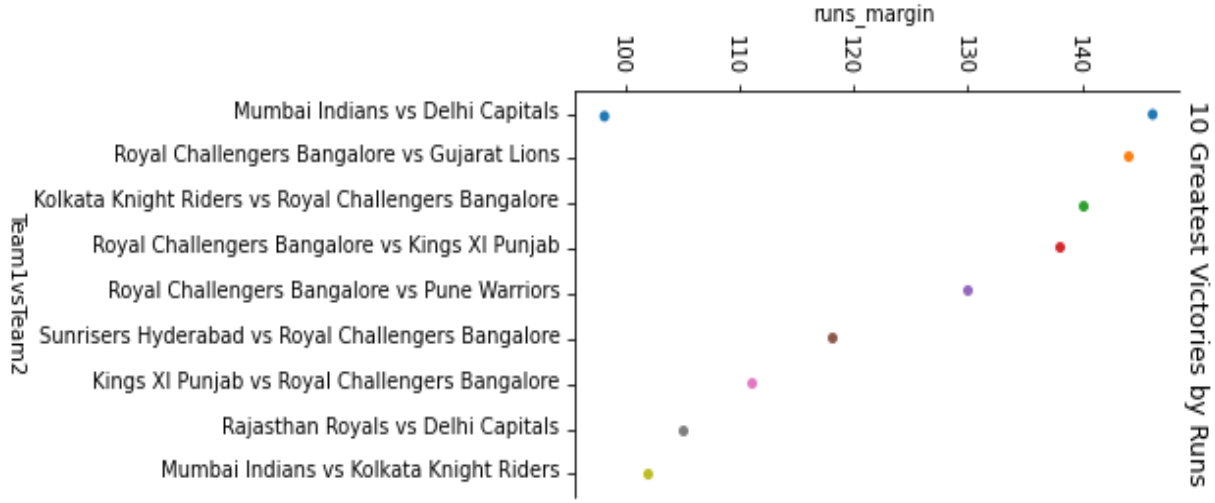
From the plot we infer that the luckiest IPL team is Mumbai Indians, followed by Kolkata Knight Riders. However, not all teams have played every year. Hence, a better way to find the luckiest team is to compare the percentage toss wins of each team, i.e., the percentage of the number of toss wins against the total number of matches played by the team.

From the second plot we infer that the luckiest IPL team is Deccan Chargers, followed by Kochi Tuskers Kerala.

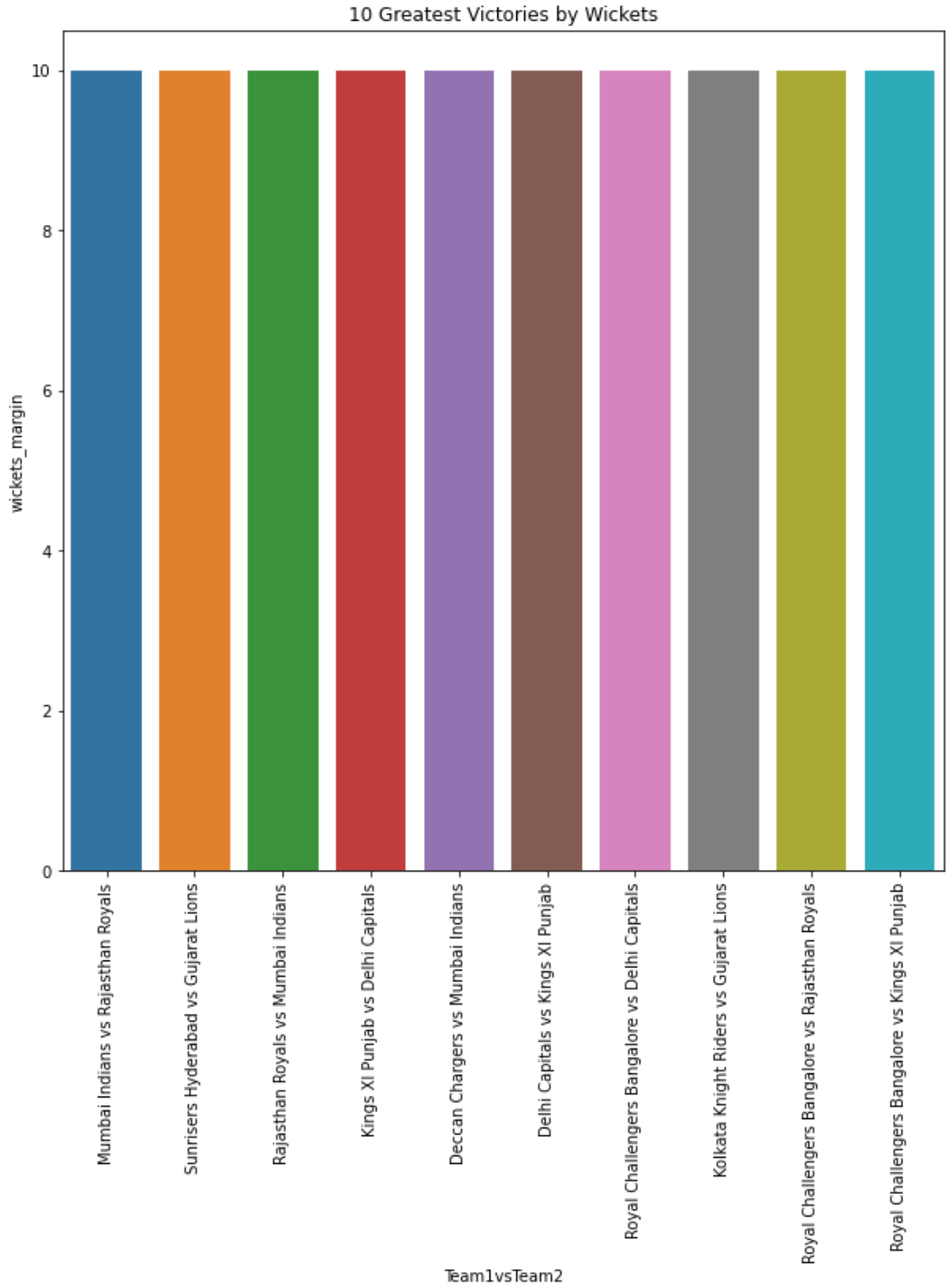


The Top 10 Greatest Victories:

The following graph shows the top 10 greatest victories by runs
Team 1 is the winning team

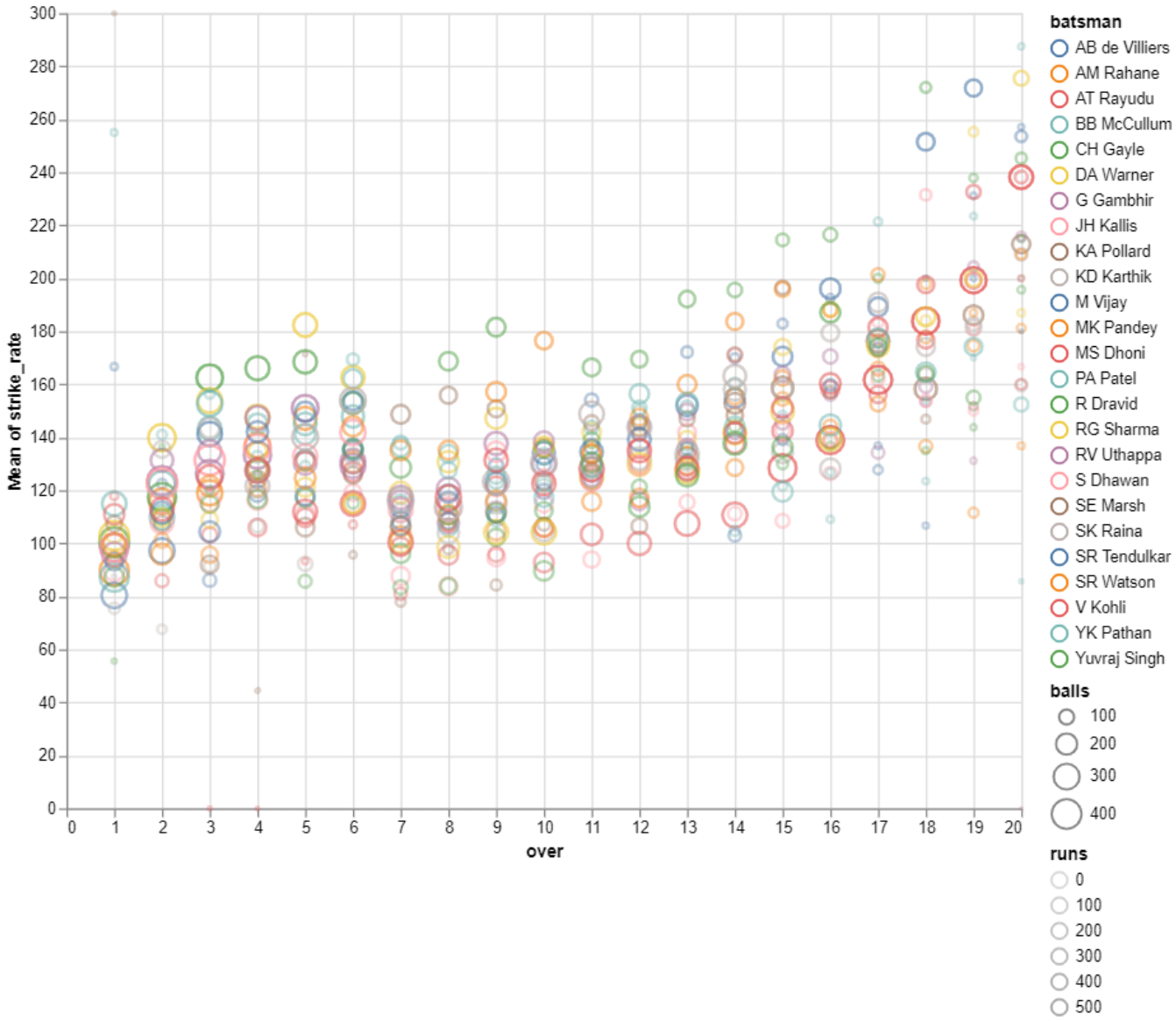


The following graph shows the top 10 greatest victories by wickets with the wickets range in the y-axis and the team name in the x-axis



Scatter Plot of Batsmen:

The following plot shows information about the top 25 batsmen. The y-axis has the mean strike rate and the x-axis has the over. The balls are coloured according to the batsmen, the size of the balls shows the number of balls faced by them and the intensity of the colour of the balls shows the runs.



Predictive Tasks

Predicting the Winner:

The task of predicting the winner is actually a classification task where the model has to classify the winner as team 1 or team 2.

For this purpose three new columns 'team1_toss_win', 'team1_bat' and 'team1_win' are made. 'team1_toss_win' gets the value 1 if 'team1' is the toss winner, else it gets the value 0. 'team1_bat' gets the value 1 if 'team1' is the batting team, else it gets the value 0. 'team1_win' gets the value 1 if 'team1' is the winner, else it gets the value 0.

Upon correlation analysis, it is found that 'team1_bat' and 'team1_toss_win' are highly correlated. The reason is that the dataset is constructed in such a way that the toss winning team mostly choses to bat. Hence, 'team1_bat' is dropped.

The final dataset for prediction contains the features 'team1', 'team2', 'team1_toss_win' and 'venue' and the target variable 'team1_win'.

Models:

Random Forest Classifier:

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression. One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems.

Result:

Accuracy Score = 0.877232
Precision Score = 0.855448
Recall Score = 0.880138
F1 Score = 0.867617
ROC AUC Score = 0.877462
Average Precision Score = 0.807700

Support Vector Classifier:

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.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called support vectors, and hence the algorithm is termed as Support Vector Machine.

Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and the classifier used is called Linear SVM classifier.

Result:

Accuracy Score = 0.542909	Precision Score = 0.000000
Recall Score = 0.000000	F1 Score = 0.000000
ROC AUC Score = 0.500000	Average Precision Score = 0.457091

Extra Trees Classifier:

Extremely Randomized Trees, or Extra Trees for short, is an ensemble machine learning algorithm.

Specifically, it is an ensemble of decision trees and is related to other ensembles of decision trees algorithms such as bootstrap aggregation (bagging) and random forest.

The Extra Trees algorithm works by creating a large number of unpruned decision trees from the training dataset. Predictions are made by averaging the prediction of the decision trees in the case of regression or using majority voting in the case of classification.

Unlike bagging and random forest that develop each decision tree from a bootstrap sample of the training dataset, the Extra Trees algorithm fits each decision tree on the whole training dataset.

Like a random forest, the Extra Trees algorithm will randomly sample the features at each split point of a decision tree. Unlike random forest, which uses a greedy algorithm to select an optimal split point, the Extra Trees algorithm selects a split point at random.

Result:

Accuracy Score = 0.877385	Precision Score = 0.857687
Recall Score = 0.877319	F1 Score = 0.867392
ROC AUC Score = 0.877379	Average Precision Score = 0.808541

Gradient Boosting Classifier:

Gradient boosting classifiers are a group of machine learning algorithms that combine many weak learning models together to create a strong predictive model. Decision trees are usually used when doing gradient boosting. In the case of Gradient Boosting Machines, every time a new weak learner is added to the model, the weights of the previous learners are frozen or cemented in place, left unchanged as the new layers are introduced. This is distinct from the approaches used in AdaBoosting where the values are adjusted when new learners are added. The power of gradient boosting machines comes from the fact that they can be used on more than binary classification problems, they can be used on multi-class classification problems and even regression problems. Taking random subsamples of the training data set, a technique referred to as stochastic gradient boosting, can also help prevent overfitting. This technique essentially reduces the strength of the correlation between trees.

Result:

Accuracy Score = 0.655113	Precision Score = 0.702466
Recall Score = 0.425842	F1 Score = 0.530245
ROC AUC Score = 0.636993	Average Precision Score = 0.561582

Predicting the Final Score:

The task of predicting the final score is a regression task.

For this purpose, three new columns ‘total_over_runs’, ‘current_score’ and ‘final_score’ are made. ‘total_over_runs’ contains the value of the total runs scored in that over, ‘current_score’ contains the current score of the inning and ‘final_score’ contains the final score of that inning.

The final dataset for prediction contains the features ‘venue’, ‘inning’, ‘batting_team’, ‘bowling_team’, ‘batsman’, ‘bowler’, ‘total_over_runs’ and ‘current_score’ and the target variable ‘final_score’.

Models:

Random Forest Regressor:

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression. One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems.

Result:

Custom Accuracy = 88.785586 Mean Absolute Error = 4.249554

Mean Squared Error = 58.520111 Root Mean Squared Error = 7.649844

Mean Squared Log Error = 0.002893 Root Mean Squared Log Error = 0.053782

Extra Trees Regressor:

Extremely Randomized Trees, or Extra Trees for short, is an ensemble machine learning algorithm.

Specifically, it is an ensemble of decision trees and is related to other ensembles of decision trees algorithms such as bootstrap aggregation (bagging) and random forest.

The Extra Trees algorithm works by creating a large number of unpruned decision trees from the training dataset. Predictions are made by averaging the prediction of the decision trees in the case of regression or using majority voting in the case of classification.

Unlike bagging and random forest that develop each decision tree from a bootstrap sample of the training dataset, the Extra Trees algorithm fits each decision tree on the whole training dataset.

Like a random forest, the Extra Trees algorithm will randomly sample the features at each split point of a decision tree. Unlike random forest, which uses a greedy algorithm to select an optimal split point, the Extra Trees algorithm selects a split point at random.

Result:

Custom Accuracy = 93.201792 Mean Absolute Error = 2.618441
Mean Squared Error = 35.379998 Root Mean Squared Error = 5.948109
Mean Squared Log Error = 0.001672 Root Mean Squared Log Error = 0.040885

Gradient Boosting Regressor:

Gradient Boosting algorithm is used to generate an ensemble model by combining the weak learners or weak predictive models. Gradient boosting algorithm can be used to train models for both regression and classification problems. Gradient Boosting Regression algorithm is used to fit the model which predicts the continuous value.

Result:

Custom Accuracy = 36.598690 Mean Absolute Error = 17.942199
Mean Squared Error = 546.012511 Root Mean Squared Error = 23.366911
Mean Squared Log Error = 0.026843 Root Mean Squared Log Error = 0.163837

Support Vector Regressor:

In machine learning, Support Vector Machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. In Support Vector Regression, 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 are called Support Vectors. These influence the position and orientation of the hyperplane and thus help build the SVM.

Support Vector Regression is a supervised learning algorithm that is used to predict discrete values. Support Vector Regression uses the same principle as the SVMs. The basic idea behind SVR is to find the best fit line. In SVR, the best fit line is the hyperplane that has the maximum number of points.

Unlike other Regression models that try to minimize the error between the real and predicted value, the SVR tries to fit the best line within a threshold value.

Result:

Custom Accuracy = 33.178379 Mean Absolute Error = 20.559682
Mean Squared Error = 719.027079 Root Mean Squared Error = 26.814680
Mean Squared Log Error = 0.034748 Root Mean Squared Log Error = 0.186409

Neural Network:

A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature.

Neural networks can adapt to changing input; so the network generates the best possible result without needing to redesign the output criteria. The concept of neural networks, which has its roots in artificial intelligence, is swiftly gaining popularity in the development of trading systems. A neural network works similarly to the human brain's neural network. A "neuron" in a neural network is a mathematical function that collects and classifies information according to a specific architecture. The network bears a strong resemblance to statistical methods such as curve fitting and regression analysis. A neural network contains layers of interconnected nodes. Each node is known as a perceptron and is similar to a multiple linear regression. The perceptron feeds the signal produced by a multiple linear regression into an activation function that may be nonlinear.

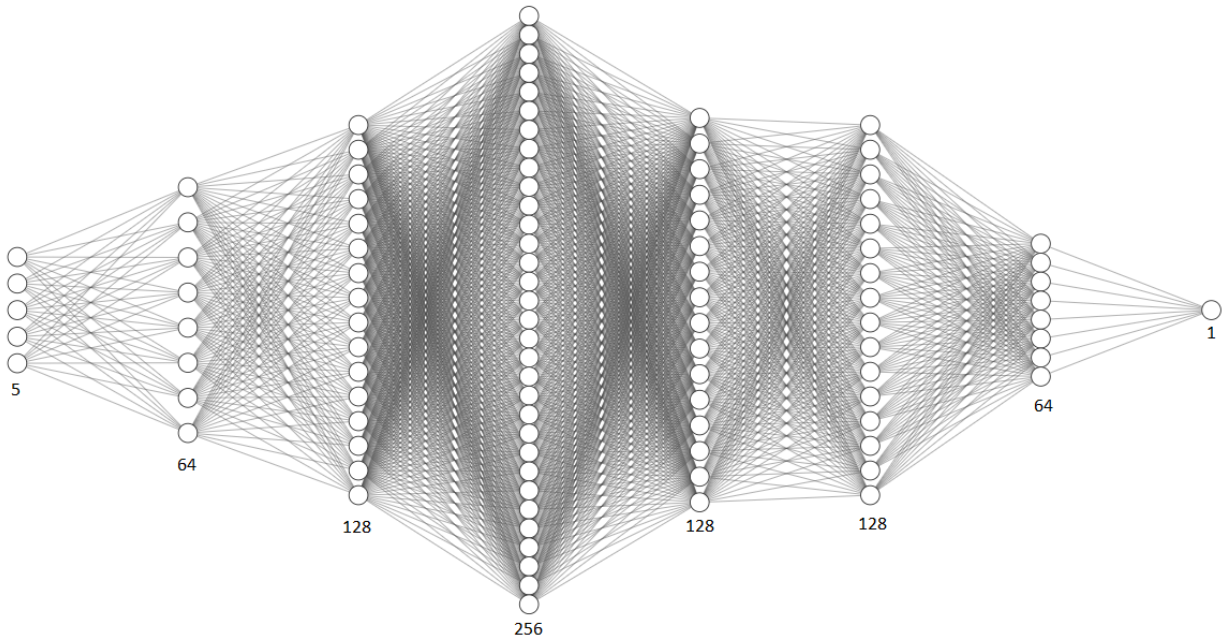


Figure shows the architecture of the Neural Network used for prediction. Batchnorm is applied after each linear operation before applying leaky relu as activation function. Batchnorm is used to prevent exploding of units' activations and consequently the gradients. Leaky relu prevents 'dead neurons' as it always has a slope for gradient computation. Adam optimizer is used for backpropagation and regularization purposes. Mini-batches of 128 size are used for mini-batch gradient descent. The training is performed on Google Colab with GPU runtime and the model is built in Pytorch, a deep learning framework. The total number of learnable parameters in the model are 100k and it is run for 200 epochs with 0.001 as the learning rate. These hyperparameters for learning are chosen after running many experiments.

Metrics used:

Mean squared error is used as a loss function for backpropagation. R2 score and custom accuracy (predicted score being in margin of 10 of actual final score) is used for evaluating the model.

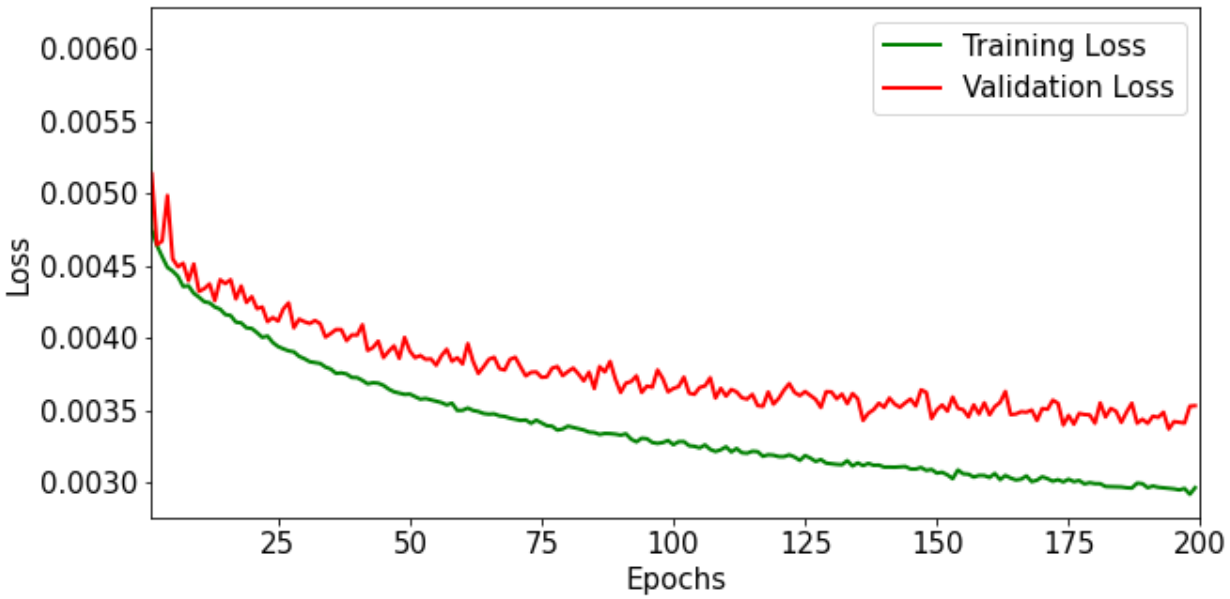
Result:

After 200 epochs, the results obtained are as follows:

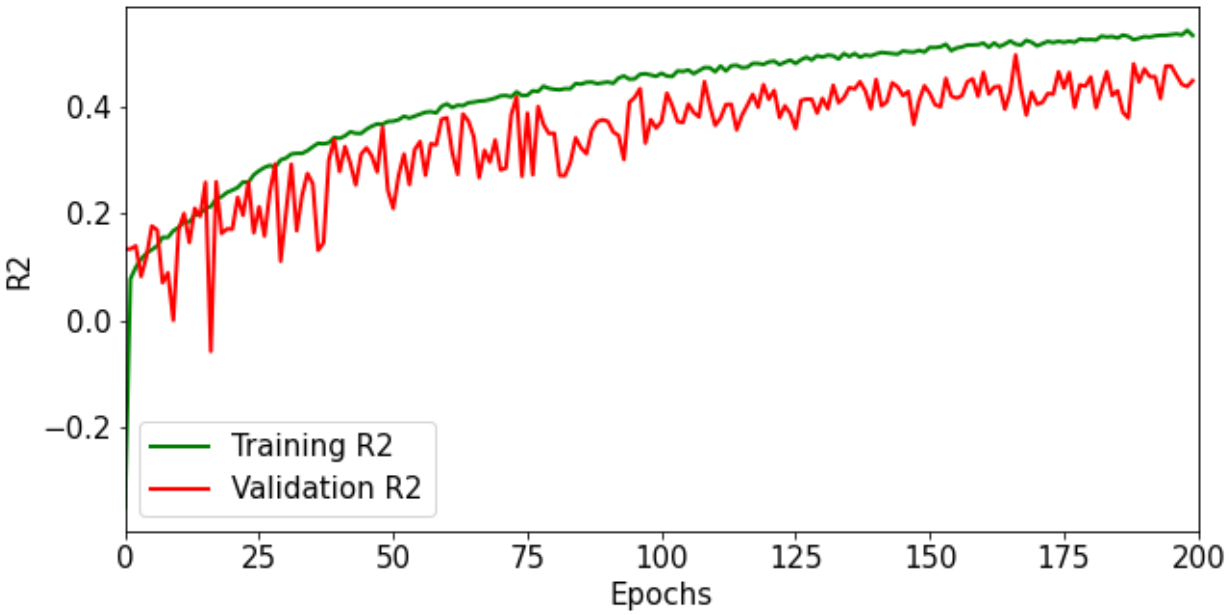
Train Loss: 0.0030 | Val Loss: 0.0035 | Train R2: 0.5330 | Val R2: 0.4484 | Val Acc: 45.0934 %.

So, the accuracy is around 45% which is not usable for practical purposes. Limited amount of data is the primary reason why neural networks are not giving decent results even after training for around 1 hour. But, this is not entirely useless as can be seen in the training graphs as shown in figures.

Test Acc: 54.7843 % | Test R2: 0.0003268



Training and Validation losses are decreasing with epochs



R2 score is also increasing for both training and validation data

Conclusion and Future Scope

This project provides useful insights from the IPL dataset about what are the best performing teams and players. Sponsors can focus on which cities host the IPL matches most to analyze the audience in those areas specifically and make their plans accordingly. Best performing players of IPL can be listed with the most MoM awards analysis. Toss decisions have more or less no influence on winning.

For the prediction of the winner, the models proposed in the work take into account the teams that are playing, the probability of toss winning of the teams and the venue. Due to limited data, the best model is 87% accurate.

The prediction of final score at any given moment of match is currently done with the help of Current Run Rate(CRR), while it is one of the useful features, it doesn't take into account what are the remaining overs and scores of the batsmen at the crease. The models proposed in the work take these features into account to predict the final score given these features at any point in the game. Due to limited data, the best model is 93% accurate.

Future Work can be pre-training the neural network models on an ODI or T20 international datasets and then fine tuning them for ipl predictions as direct training with datasets is not possible due to different formats and playing conditions.

Further, it is also possible to scrap Time Series data for any Batsman/ Bowler/ Team and forecast future performance of that batsman/ bowler/ team using a Time Series model.

We can build a Deep Learning model to simulate a whole T20 match ball-by-ball with historic data.

We can even train a convolutional neural network model on static images of cricketing shots. Then, using historic commentary data we can build a model that works on live cricket video and generates live commentary using a trained model.