IPL Data Analysis

Winter in Data Science

Aziz Shameem

20d070020

Mentor : Prayas Jain

Contents

1 Introduction

In this project, I have attempted to perform Data Analysis on available data on the Indian Premier League, a yearly cricket tournament that started way back in year 2008.
I have performed Exploratory data analysis, followed by employment of ml and dl models to predict final score reached by a side in an inning, as well as winner of the match based on some pre-determined factors.

1.1 A Brief History

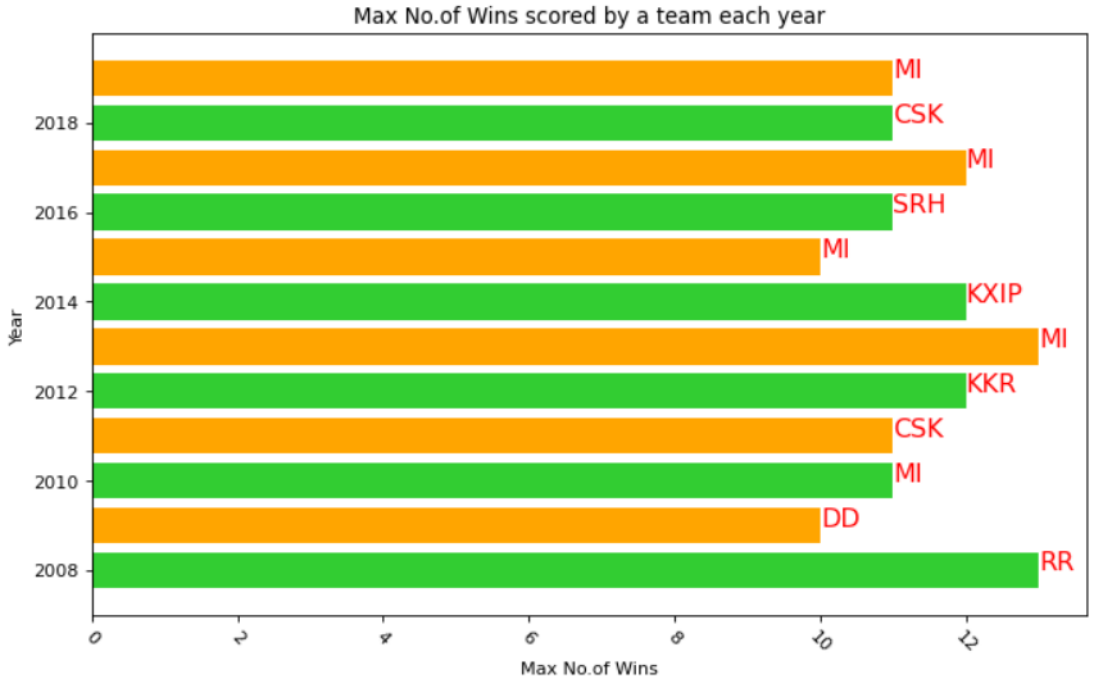
The Indian Premier League (IPL) is a professional men's Twenty20 cricket league, contested by ten (or less) teams based out of Indian cities. The league was founded by the Board of Control for Cricket in India (BCCI) in 2007. It is usually held between March and May of every year and has an exclusive window in the ICC Future Tours Programme.

2 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is an approach in analyzing data sets to summarize their main characteristics, often using statistical graphics and other data visualization methods.
EDA assists Data science professionals in various ways:-

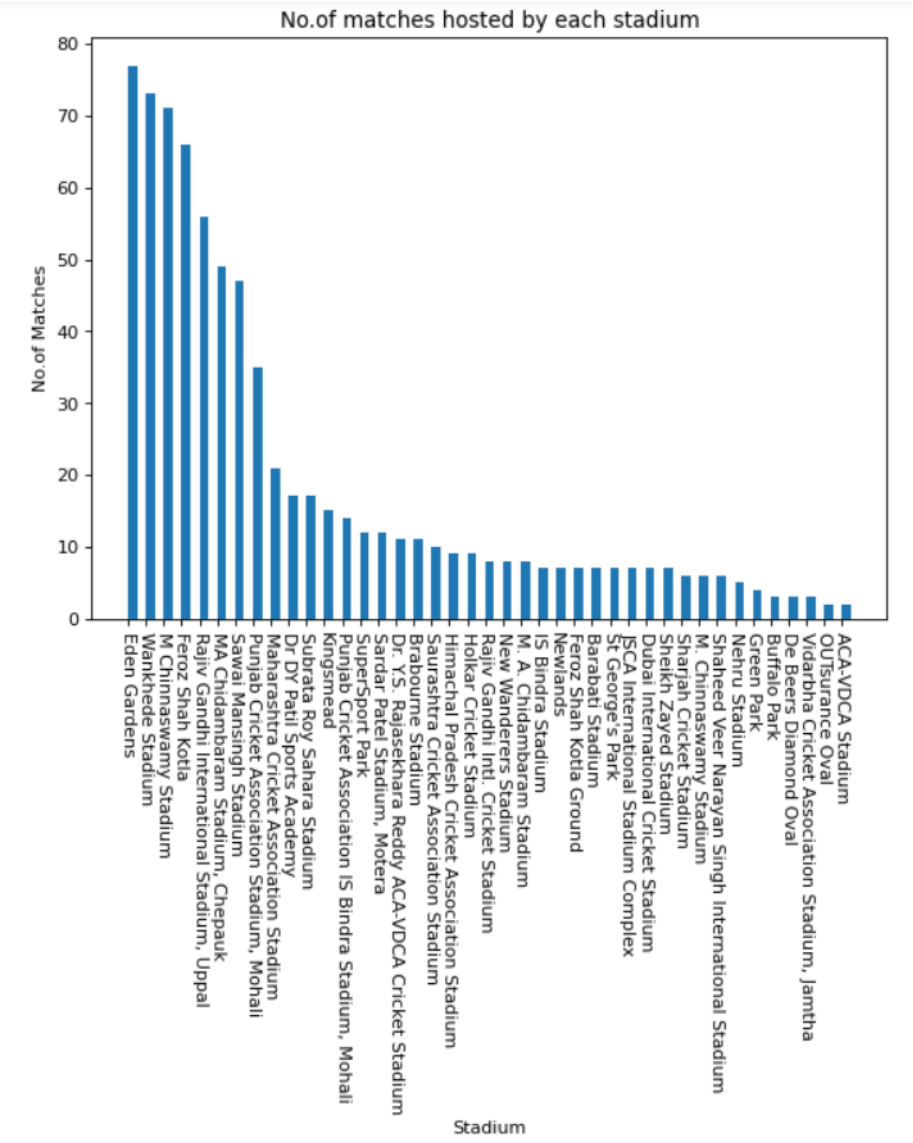
1. Getting a better understanding of data
2. Identifying various data patterns
3. Getting a better understanding of the problem statement

2.1 Team with max wins each season



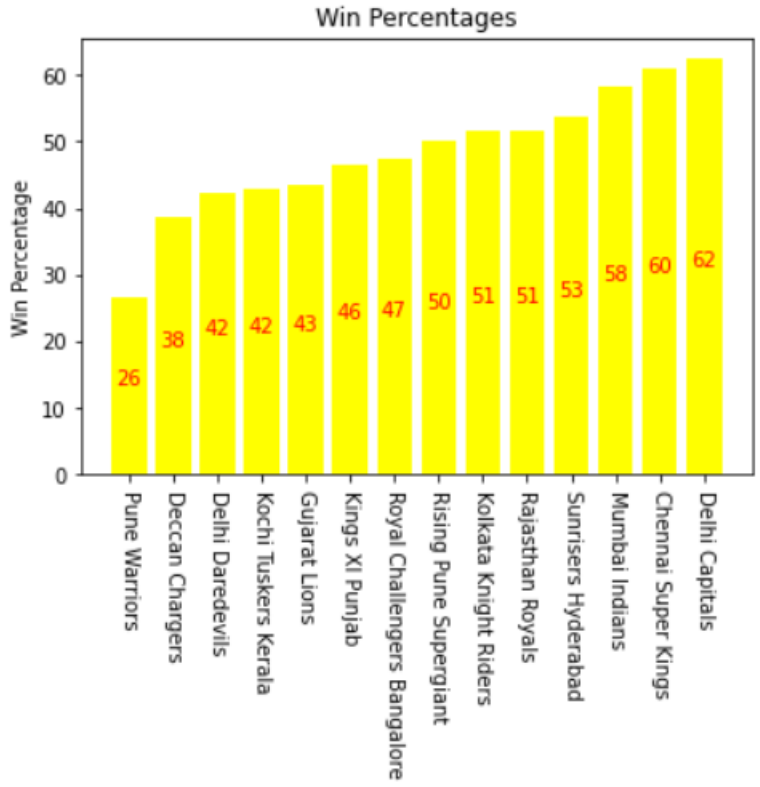
As is clear from the plot, only seven teams have been able to get maximum number of wins in a season, out of which Mumbai Indians (MI) have been the most successfull, getting the top spot 5 times : In 2010, 2013, 2015, 2017 and 2019.

2.2 No.of Matches hosted by each stadium



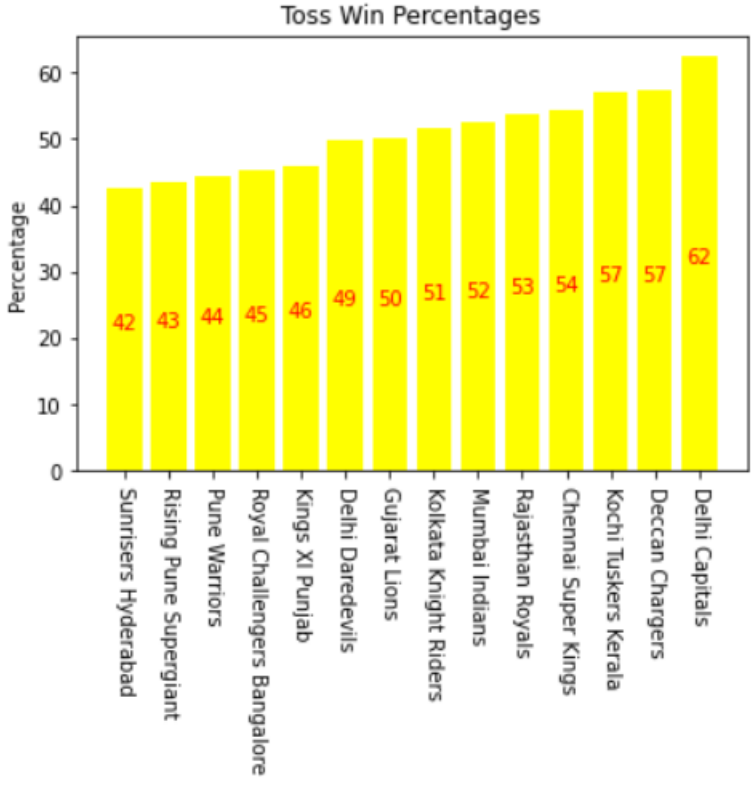
Here, the stadiums present in cities/ states having a team in the tournament are seen to have a way higher number of matches hosted, than the rest.
Highest number of matches have been hosted in Eden Gardens, Kolkata.

2.3 Win Percentages



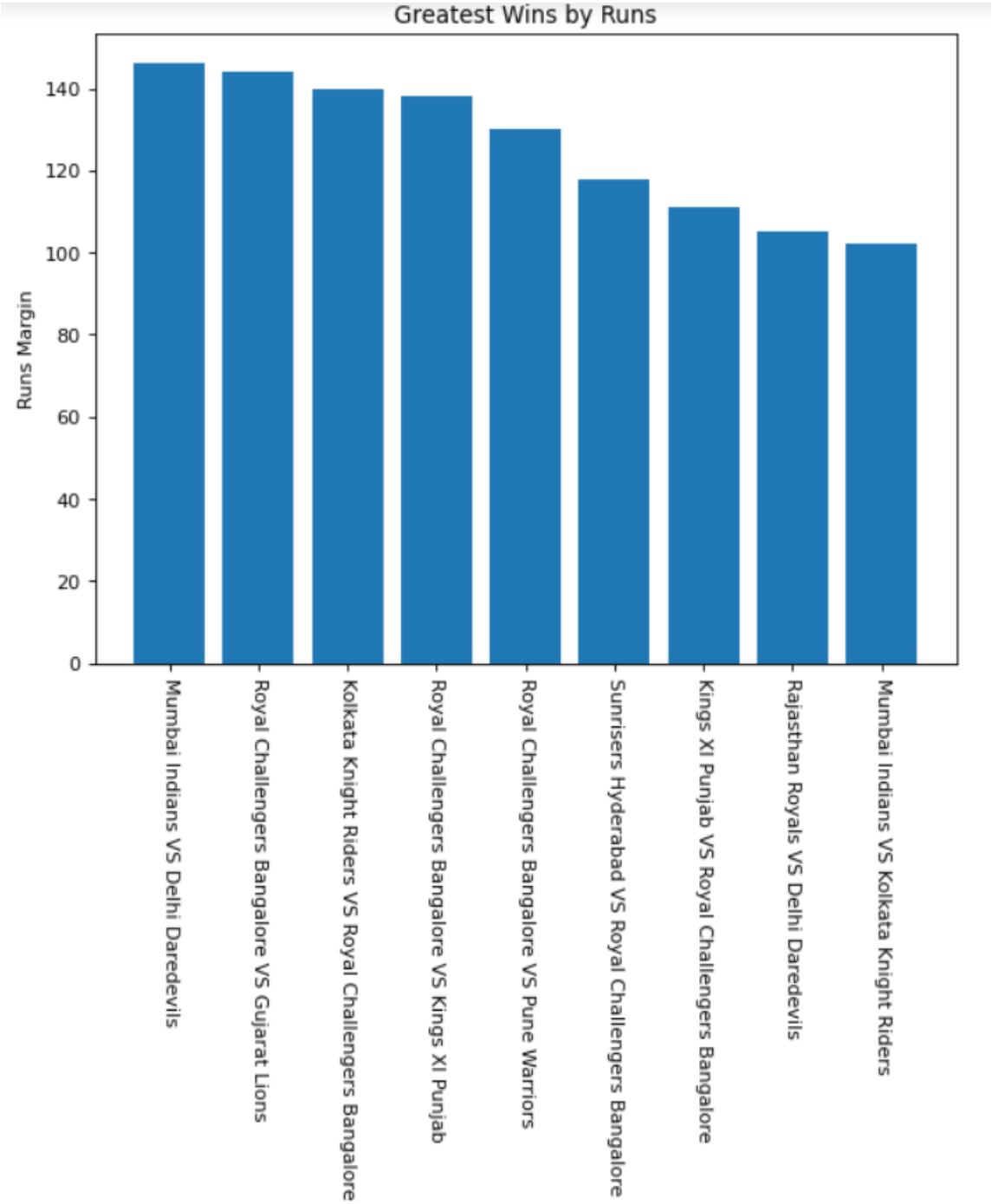
As shown in the plot, Delhi Capitals has the maximum win percentage of 62%, however, this is not a true measure of the strength of the teams as it does not take into account the total number of matches played. Delhi Capitals is a relatively new team, thus, one good season boosts up its win percentage as opposed to some of the older teams, like Mumbai Indians.

2.4 Toss Win Percentage



2.5 Top Ten Greatest Victories

2.5.1 By Runs

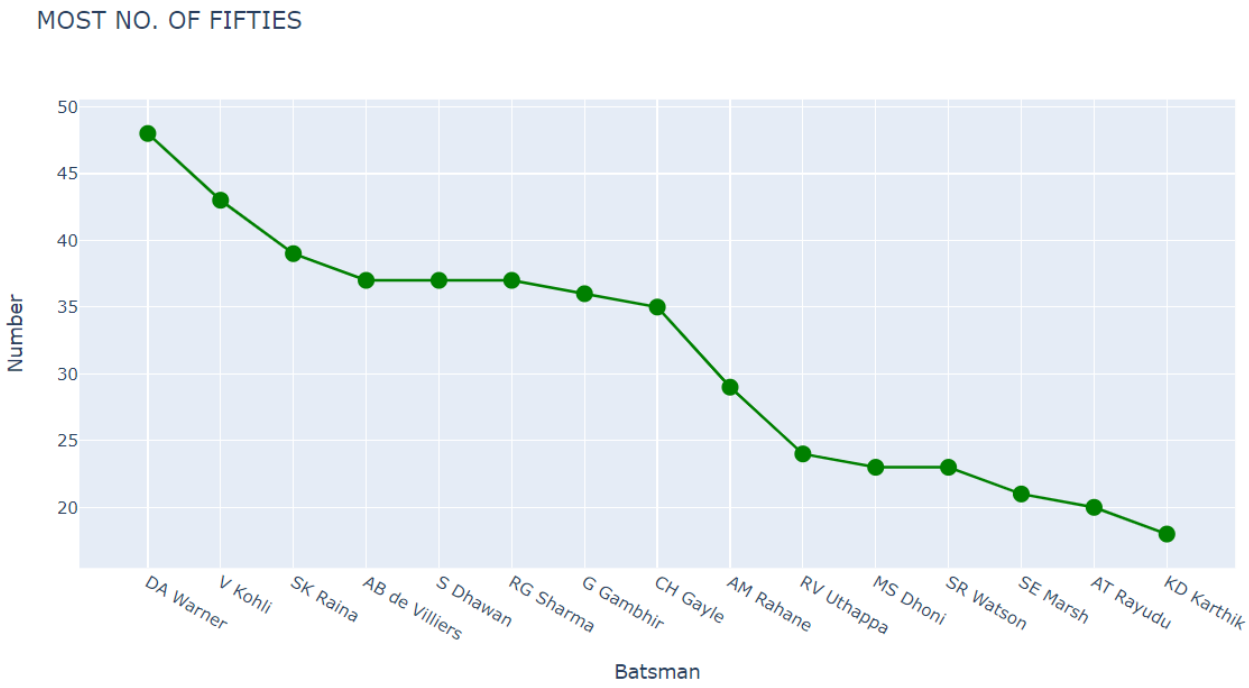


2.5.2 By Wickets

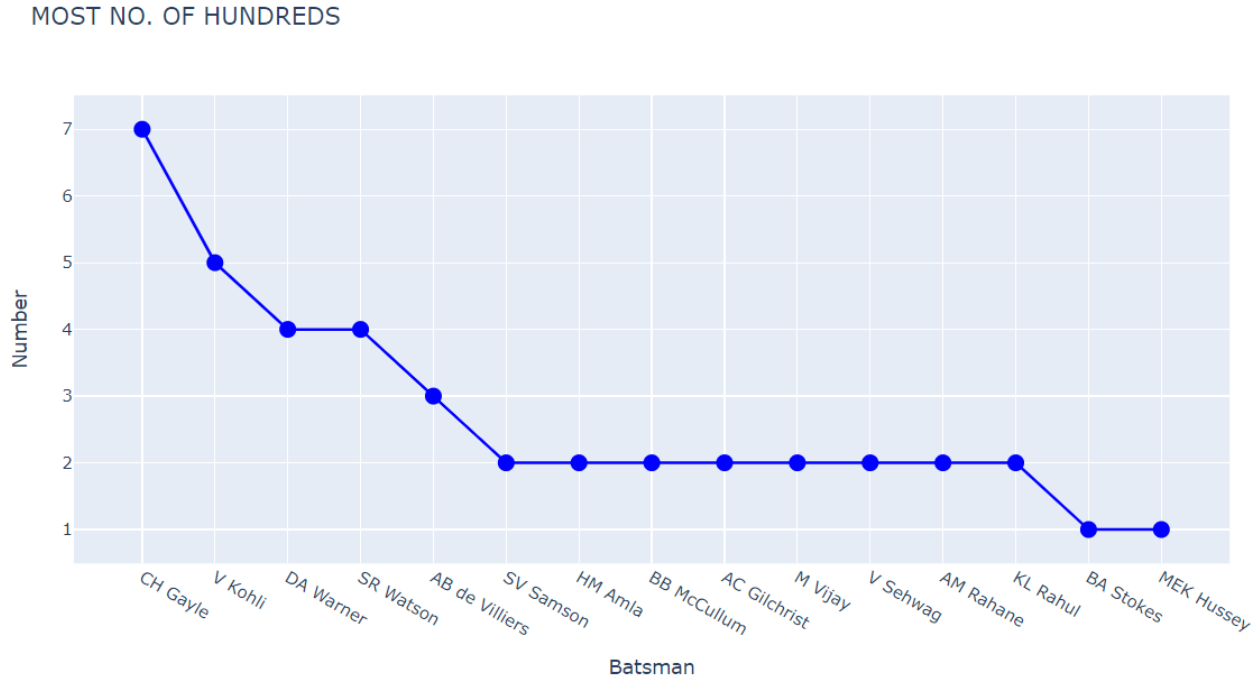
There have been a number of matches (more than 10) where the team batting second have won without losing a wicket, i.e. by 10 wickets. Since the margin cannot increase any further, this is the highest margin by which a team has won batting second.

2.6 Most 50s and 100s Scored

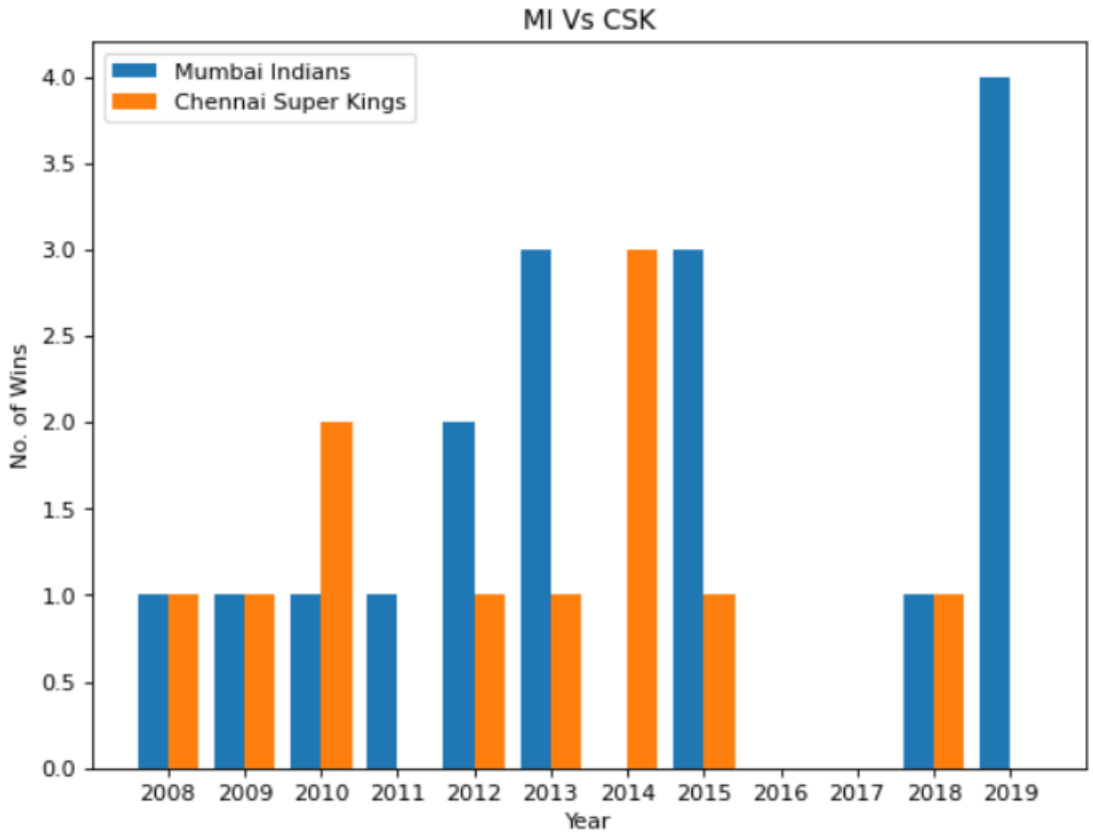
2.6.1 Most Number of 50s by a Batsman



2.6.2 Most Number of 100s by a Batsman

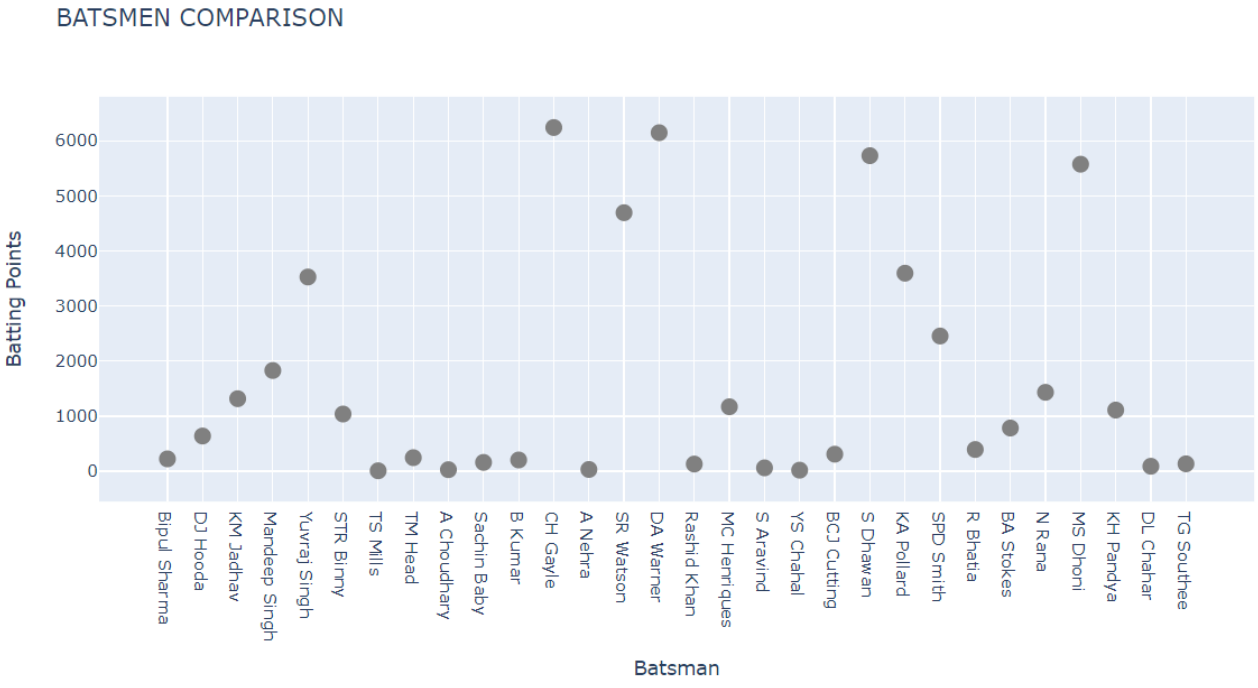


2.7 Comparison between Teams



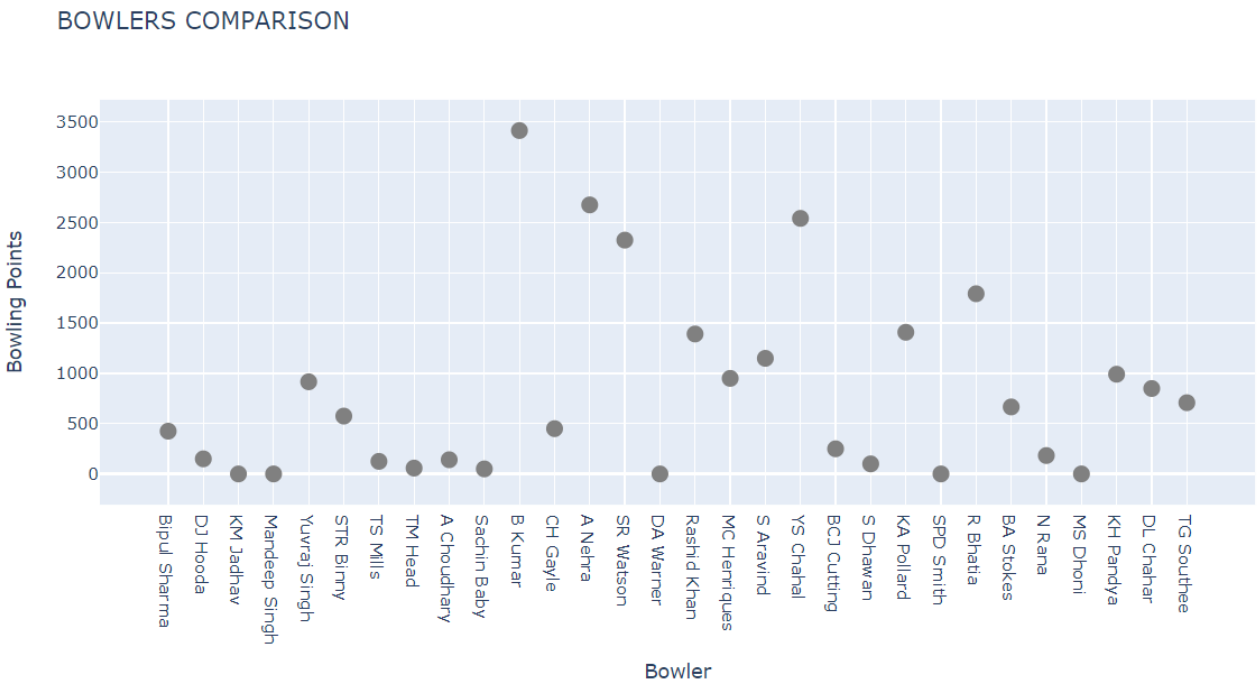
The above plot shows the number of wins a team has gotten against the other, in various seasons. Note the the area around years 2016 and 2017 appears bare, this is because Chennai Super Kings did not play in these seasons.
A similar plot can be generated between any two teams playing in the IPL.

2.8 Comparison between Batsmen



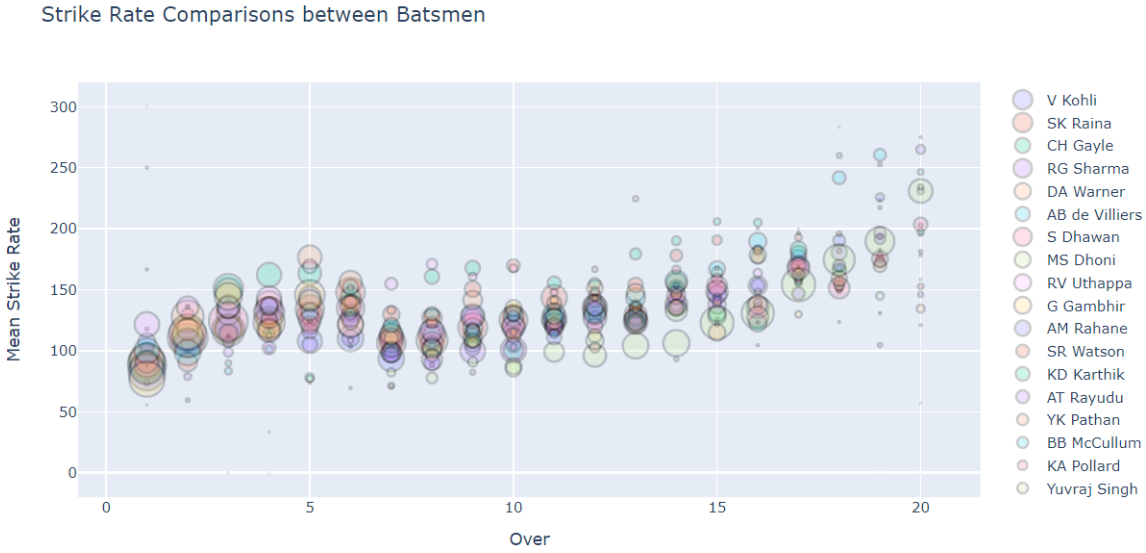
The above scatter plot shows the comparison among 10 batsmen, based on *Batting Points*, a feature generated taking into account their performance in IPL matches.
A similar plot can be generated for any list of batsmen.

2.9 Comparison between Bowlers



The above scatter plot shows the comparison among 10 bowlers, based on *Bowling Points*, a feature generated taking into account their performance in IPL matches.
A similar plot can be generated for any list of bowlers.

2.10 Mean Strike Rate Comparison



The above bubble plot shows the Mean Strike Rate variation of 10 chosen batsmen, across different overs. The colours of the bubbles indicate the Batsman, and the size of the bubble indicate the prior number of balls faced by them, to measure the experience gained on field.
A similar plot can be generated for any list of batsmen.

3 ML Models

3.1 Task 1 : Regression

Here, we attempt to predict the final score of an innings based on factors like present runs, present no.of wickets fallen, venue of the match, opponent team and toss results.
Three models were applied for this purpose.

3.1.1 Linear Regression

Linear regression, at it's core, is a way of calculating the relationship between two variables. It assumes that there's a direct correlation between the two variables, and that this relationship can be represented with a straight line.

These two variables are called the independent variable and the dependent variable, and they are given these names for fairly intuitive reasons. Linear regression creates a linear mathematical relationships between these two variables. It enables calculation predicting the independent variable if the dependent variable is known.

3.1.2 Gradient Boosting Regressor

Gradient boosting is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees. It is one of the most powerful techniques for building predictive models.

3.1.3 Random Forest Regressor

Random forest is a Supervised Learning algorithm which uses ensemble learning method for classification and regression. It is a bagging technique and not a boosting technique. The trees in random forests are run in parallel. There is no interaction between these trees while building the trees.
It operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

3.1.4 Results

First, we pre-process the data. This includes dealing with missing values, removing data on super-overs/ matches with no results, removing redundant/ not useful data and one-hot encoding of fields containing categorical values.
Then, we add three columns to the dataset : '*Cumulative Runs*', '*Cumulative Wickets*' and '*Final Score*'.
Finally, we split the data for training and testing, and scale it appropriately.

Here are the results obtained :

Model	F1 Score	RMSE
Linear Regression	0.469	17255787.135
Gradient Boosting Regressor	0.993	218882.797
Random Forest Regressor	0.988	380756.48

Clearly, Linear Regression, being a naive model, performed worse than sophisticated models like Random Forest and Gradient Boosting.
Gradient Boosting Regressor could procure the maximum accuracy (among the three) in its predictions.

3.2 Task 2 : Classification

Here, we attempt to predict the final outcome of the match, based on venue of play, the teams playing and toss results. The features used in this prediction are very naive, and the outcome depends on many other factors such as conditions on weather, pitch, out-field, the form of players/ coaches etc. Nonetheless, as we will see, this also gives a reasonable level of accuracy.
Four models were applied for this purpose

3.2.1 Logistic Regression

Logistic regression is a process of modeling the probability of a discrete outcome given an input variable. The most common logistic regression models a binary outcome; something that can take two values such as true/false, yes/no, and so on. Multinomial logistic regression can model scenarios where there are more than two possible discrete outcomes.
Logistic regression is a useful analysis method for classification problems, where you are trying to determine if a new sample fits best into a category.

3.2.2 Support Vector Machine

"Support Vector Machine" (SVM) is a supervised machine learning algorithm that can be used for both classification or regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is a number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well.

3.2.3 Random Forest Classifier

This model is similar to the one described in section *3.1.3*. The only difference lies in the fact that this model outputs one out of a discrete set of values.

3.2.4 Gradient Boosting Classifier

This model is similar to the one described in section *3.1.2*. The only difference lies in the fact that this model outputs one out of a discrete set of values.

3.2.5 Results

First, we pre-process the data. This includes dealing with missing values, removing data on super-overs/ matches with no results, removing redundant/ not useful data and one-hot encoding of fields containing categorical values.
Finally, we split the data for training and testing, and scale it appropriately.

Here are the results obtained :

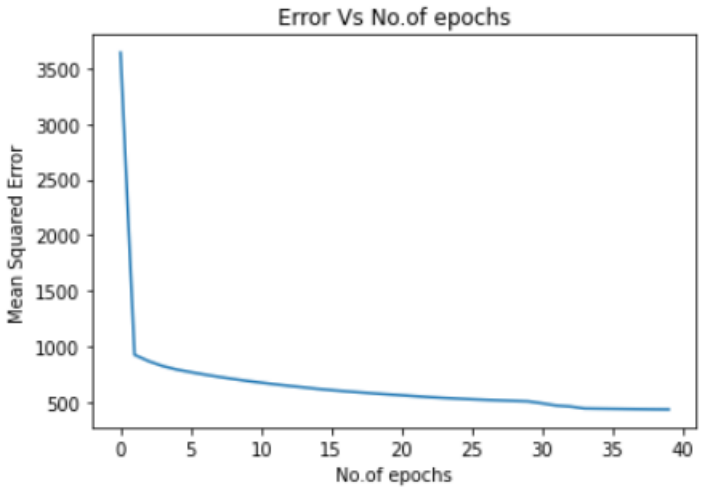
Model	Accuracy
Logistic Regression	60.396 %
Support Vector Classifier	51.485 %
Random Forest Classifier	58.416 %
Gradient Boosting Classifier	50.495 %

Logistic Regression outperforms the other models.

4 DL Models

Firstly, we pre-process the data and add relevant columns as described in the first part of section *3.1.4*

We then Construct a basic Neural Network with one hidden layer containing 5 neurons. Several values of *batch-sizes* and *epoch-sizes* are tried to see which fits best for out data.
Batch-size of 10 and epoch-size of 40 appears to work for our purposes.
Here is the plot of the mean squared error, against the no.of epochs.



Notice how the error falls off sharply in the beginning, but flattens out later.

5 Web Scraping

Some amount of data gathered for this project was obtained by scraping from the official site of the Indian Premier League, iplt20.com.
For Web Scraping, I have used the following python libraries : **BeautifulSoup4** and **urllib**.
This data was processed and used as an addition to the data already present in the form of downloadable CSVs.

6 References

1. Complete ipl dataset https://www.kaggle.com/patrickb1912/ipl-complete-dataset-20082020
2. Official Site iplt20.com
3. Application of models https://scikit-learn.org/
4. Tutorial on Plotly https://www.geeksforgeeks.org/python-plotly-tutorial/
5. Tutorial on TensorFlow for DL https://www.tensorflow.org/tutorials