IPL SPORTS ANALYSIS

IT606 - Programming Lab (Python)

Where knowledge meets ANALYSIS;
Every INSIGHT is valuable

By: Nishant Koradia(202118009)

Vidhi Shah(202118037)

Table of Contents

1. Problem Definition

Exploratory Data Analysis and Statistical Analysis with IPL matches and deliveries dataset from the seasons 2008 to 2019.

2. Problem Explanation

Indian Premier League (IPL) is a Twenty20 cricket format league in India. It is usually played in April and May every year. As of 2019, the title sponsor of the game is Vivo. The league was founded by Board of Control for Cricket India (BCCI) in 2008.There have been twelve seasons of the IPL tournament. The current IPL title holders are the Mumbai Indians, who won the 2019 season.

Cricket is a game of numbers - the runs scored by a batsman, the wickets taken by a bowler, the matches won by a cricket team, the number of times a batsman responds in a certain way to a kind of bowling attack, etc. The capability to dig into cricketing numbers for both improving performance and studying the business opportunities, overall market, and economics of cricket via powerful analytics tools. Cricket analytics provides interesting insights into the game and predictive intelligence regarding game outcomes.

The dataset has been downloaded from Kaggle Dataset. The dataset contains two csv files but i.e. matches.csv and deliveries.csv has been used in this project for the analysis. The libraries for data analysis and visualization used in this project are numpy, pandas, matplotlib and seaborn.

This project aims to collect the various data related to Indian Premier League regardless of any particular season. Our basic focus is to bring unseen facts through visualizing and statistically analysing the various data figures. We can also avail a range of good ideas & scopes about the possible data science work. We strive to analyse the data which can change the dynamics of IPL.

3. Analysis Performed

1) Importing Libraries

- We have imported *pandas* library as *pd* variable which provides high-performance, easy-to-use data structures and data analysis tools for the Python programming language.
- We have imported *Numpy* library as *np* variable used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices.
- We have imported *matplotlib.pyplot* as *plt* variable provides an implicit, MATLAB-like, way of plotting. It also opens figures on your screen, and acts as the figure GUI manager.
- We have imported *seaborn* as *sns* variable which further provides a high-level interface for drawing attractive and informative statistical graphics.
- We have imported *warnings* which *filters* the unwanted warnings occurs during the execution of code.
- We have installed *kaggle package* to use Kaggle's public API as we used dataset of Kaggle.

2) Reading Dataset

- We are Reading a .csv file named "*deliveries*" and "*matches*" from a URL storing a .csv returns the .csv as a *Pandas pd.DataFrame*.
- After that using *head* and *tail* method, we are displaying first and last 5 rows in our data frame.

3) Data Cleaning

- Using *drop* method, we have dropped inessential columns from our Data Frame.
- After that, we have replaced unnecessary values with *NaN* that is basically null values using *replace* function.
- By using *sum* function in *isnull* method, we are summing up the null values of each column.

4) Understanding Dataset

- Firstly, we are displaying the total number of rows and columns in both the dataset using *shape* method which returns the shape of an array.

- After that, we have described the dataset using *describe* method in which it shows basic statistical details like mean, count, standard deviation, percentiles and max of data frames or series of numerical values.
- By using *info* function, we are retrieving the information of matches dataset so that we can get a quick overview of dataset, it comes with concise summary and helps in EDA.
- Using *len()* method, we have displayed the number of matches played in total till 2019 season.
- Using *unique* function, we are displaying the unique values of some columns.
- Using *nunique* function, we are displaying total unique count of each columns.
- Using *select_dtypes()*, we are displaying object in our columns which are of string datatype in Python.

4. Statistical Analysis

Analysis 1

- **Function code:** We are displaying the Details of match in which the team won by the maximum wickets by using *max()* function which returns the highest value in column. Also, we have used *iloc()* method to display the whole row and *idxmax()* to display the first occurrence of the maximum value.

- **Observation:** Kolkata Knight Riders won by 10 wickets which is the maximum number of win by wickets in a match.

Analysis 2

- **Function code:** We are displaying the Details of match in which the team won by the maximum margin of runs by using *max()* function which returns the highest value in column. Also, we have used *iloc()* method to display the whole row and *idxmax()* to display the first occurrence of the maximum value.

- **Observation:** Mumbai Indians won by 146 runs which is the maximum margin of win by runs in all the matches played till 2019.

Analysis 3

- **Function code:** We are displaying the unique count of matches where D/L method was applied by using *value_counts()* method and percentage also for the same by rounding the decimal values up to 2 numbers using *round* function.

- **Observation:** Total 19 matches were affected by rain and D/L method was applied in order to get the result out of total 756 matches which implies that in 2.51% of matches D/L method was applied till 2019.

Analysis 4

- **Function code:** We are displaying the total number of times a team is won or lost after winning the toss by using *groupby* function in order to split each team into one group and compare the toss winner and match winner columns and percentage also for the same by rounding the decimal values up to 2 numbers using *round* function.

- **Observation:** As we can observe that out of total 756 matches, 393 times team won the match after winning the toss and 363 times team lost the match after winning the toss. Also, 51.98% of matches shows the occurrence of team winning a match after winning the toss.

Analysis 5

- **Function code:** We are displaying the name of stadiums which have hosted D/L method applied matches by using *query()* method used to query the column named 'Venue' where D/L method applied is true in matches dataframe.

- **Observation:** As we can see that, there are total 19 occurrences of stadiums from which Eden Gardens is repeated four times which showcases that chances of rain are highest there and result can be decided by D/L method.

Analysis 6

- **Function code:** We are displaying the average margin of win by runs of each team whenever they have won by runs by applying *mean()* function of *numpy* library to find the mean and *sort_values()* function to sort the values in *descending* order.

- **Observation:** As we can infer that, RCB has highest average margin of win by runs which is 35.771429 and Gujarat Lions has lowest which is 1.00. It showcases that, RCB generally wins by maximum margin of runs compared to other teams and GL wins by lowest margin of runs.

Analysis 7

- **Function code:** We are displaying the top 10 run-scorers of all time in IPL till 2019 by merging both the datasets into a new variable and after applying *sum()* method to showcase the total sum of runs scored by each batsman and using *sort_values()* function to sort values in *descending* order which showcases the top 10 values by using *head* function.

- **Observation:** As we can see that, Virat Kohli is the most successful batsman till IPL 2019 by scoring 5434 runs and similarly all other 9 players also.

Analysis 8

- **Function code:** We are displaying the orange cap winners of each season of IPL till 2019 by merging both datasets using *merge* function and using *sum()*, *sort_values()* and *groupby* function same like previous Analysis. Also, we used *drop_duplicates()* function in order to drop duplicate values of season.

- **Observation:** As we can infer that, Virat Kohli has scored 973 runs in 2016 season which is highest by a player in a single season till 2019. Similarly other batsman also scored highest runs in particular single season.

Analysis 9

- **Function code:** We are displaying total number of matches played by CSK till 2019 by summing up the count of CSK in team 1 and team 2 from *matches* dataset and storing it in new variable and displaying it using *print* function.

- **Observation:** As we can see that, CSK have played 164 matches in IPL till 2019.

Analysis 10

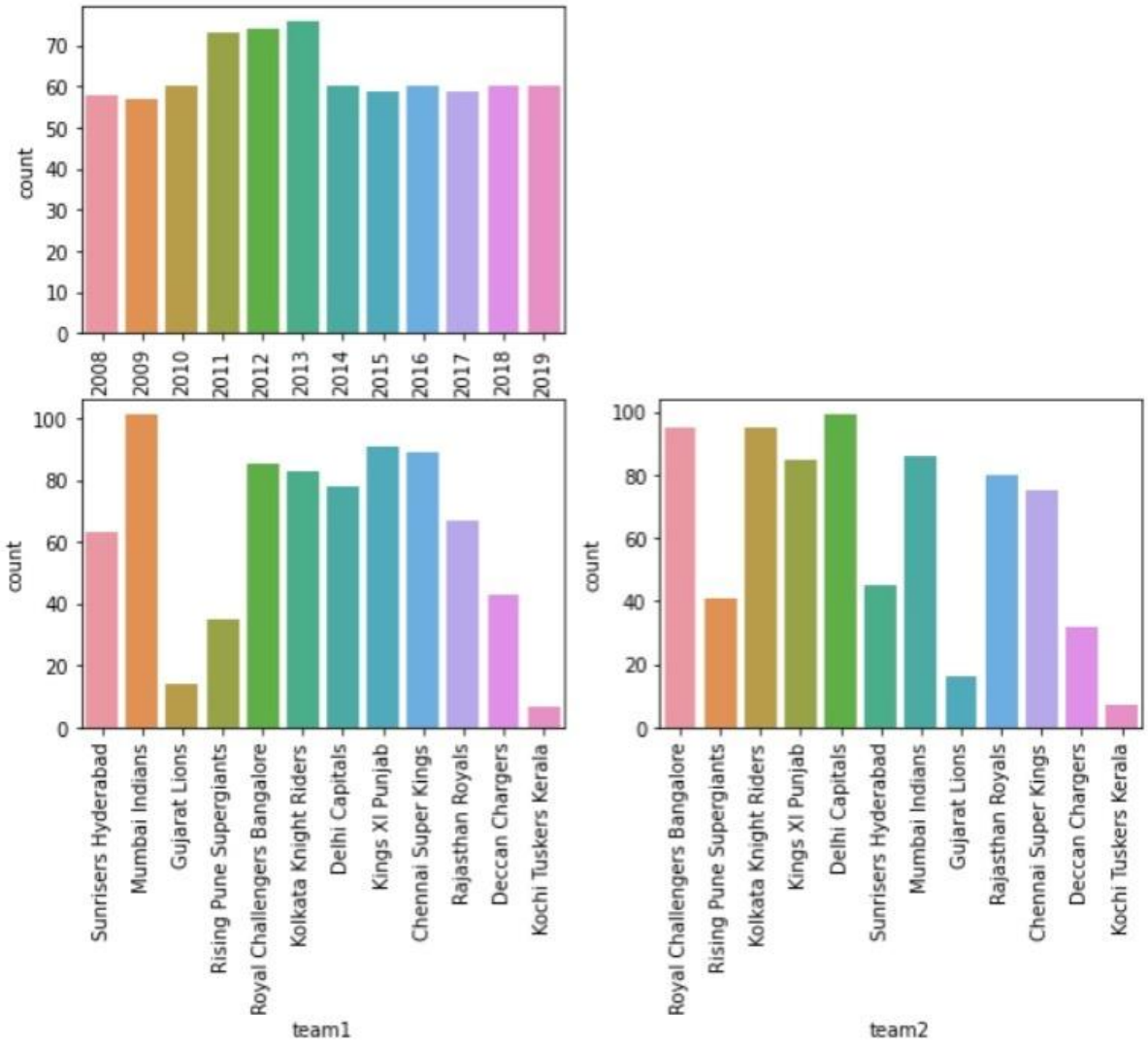
- **Function code:** We are displaying total number of runs scored in IPL till 2019 by merging matches and deliveries dataset and summing up the total runs using *sum()* function.

- **Observation:** Total runs scored in IPL till 2019 is 235290.

5. Exploratory Data Analysis

Graph 1

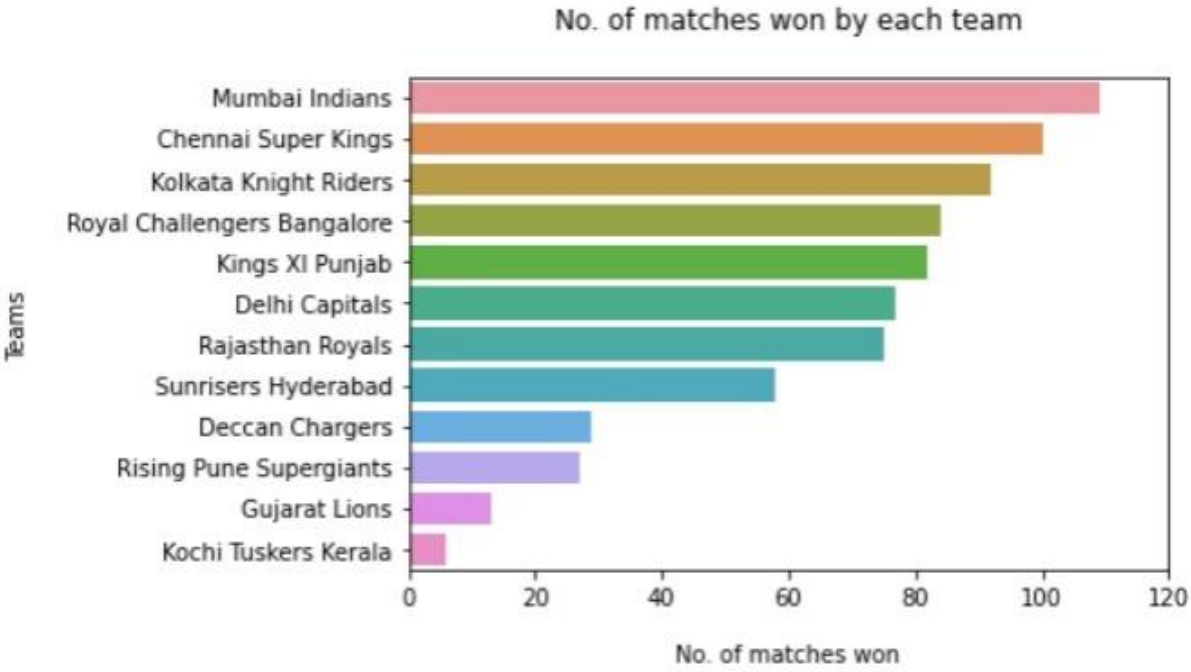
- **Function code:** Using *matplotlib* and *seaborn* library, we are visualising 3 subplots with total counts of particular variable in count plot consisting graphs which shows total matches played per season and also by each team (Team 1 and Team 2). We have plotted graphs using *figure* with *figsize(10,7)*. In *subplot*, we have encoded as a single integer for positioning of this graphs. In *countplot*, we are counting total counts of observations in each categorical bin using bars along with *xticks* and keeping *rotation 90 degree* of labels that are printed in X-axis.



- **Observation:** In 2013, as we can see that highest 76 number of matches played among all seasons till 2019 because of highest numbers of team participating in that season. Also, we can observe that Mumbai Indians played the highest number of matches till IPL 2019 and second highest team with most number of matches is CSK.

Graph 2

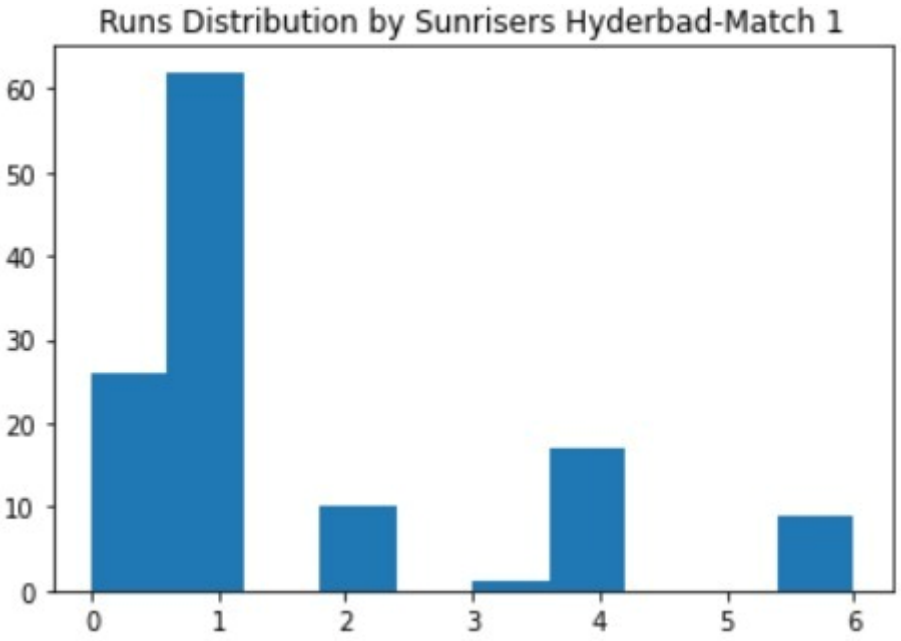
- **Function code:** We have visualised Bar Plot to visualise the total number of matches won by each team till 2019. We used *value_counts()* function of *pandas* library which return a series containing count of unique rows in a dataframe. Using *set_xlimit()*, we are defining X-axis limit from 0 to 120. After that, using *barplot()* function of *seaborn* library, we are defining position and *orientation* of the graph which is *horizontal*. We have also defined *xlabel*, *ylabel* and *title* of the graph by using *matplotlib* library.



- **Observation:** As we can observe that, Mumbai Indians has won the highest number of matches among all the other teams and CSK has won the second highest number of matches till IPL 2019.

Graph 3

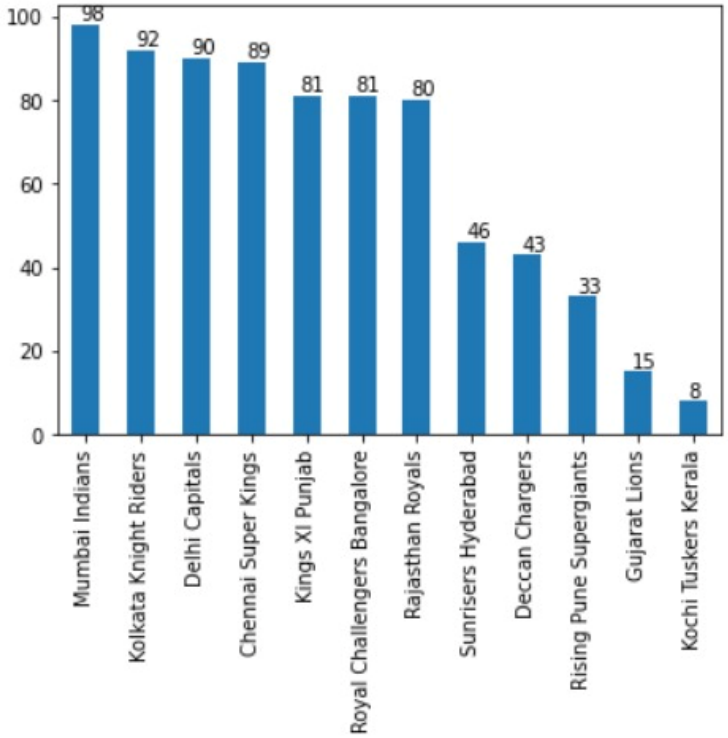
- **Function code:** We are visualising a Histogram that shows total singles, doubles, triples, fours and sixes scored by SRH in match 1. In this, we used *rename()* function to rename the *deliveries match_id* to *id* which is further used to merge both the datasets by using *merge* function. After that using *hist()* function of *matplotlib*, we are plotting histogram by counting series of runs.



- **Observation:** As we can infer that, SRH played total 26 dot balls and scored 62 singles, 10 doubles, 1 triple, 17 fours and 9 sixes in match 1.

Graph 4

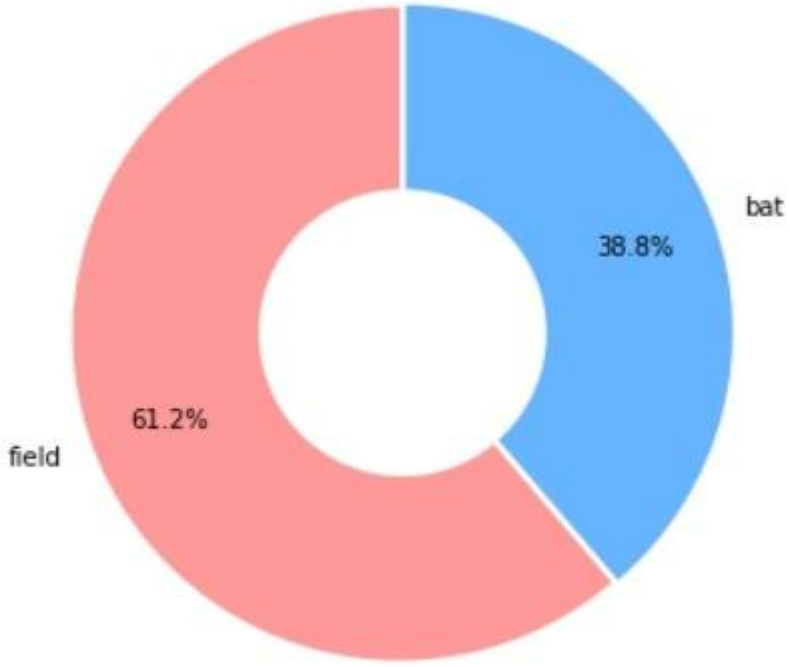
- **Function code:** The Bar Plot of *matplotlib* library that visualises the teams that won the toss most number of times in *descending* order. Using *bar()* function, we are plotting the graph and positioning labels of the graph using *get_height()* and *get_x()* of *annotate* function.



- **Observation:** As we can infer that, Mumbai Indians has won the toss highest number of times which is 98 and second highest is KKR with 92 times winning the toss.

Graph 5

- **Function code:** The pie chart of *matplotlib* library is used to visualise the decision of fielding or batting by the team winning the toss. Using *pie()* function, we have plotted pie chart with the parameters *colours* to select colour, *labels* to name label, *autopct* for string used to label the wedge with their numerical value, *startangle* which means starting from the positive x-axis the wedges are arranged in the counter clock wise direction, *pctdistance* to generate the positioning of the percentage value from the centre of circle and *explode* to highlight or emphasize key data in a pie chart. Using *circle()* function, we have drawn a circle and got the current figure by *gcf()* function and passing it to *gca()* method to get the current artist and adding it by *add_artist()* method.

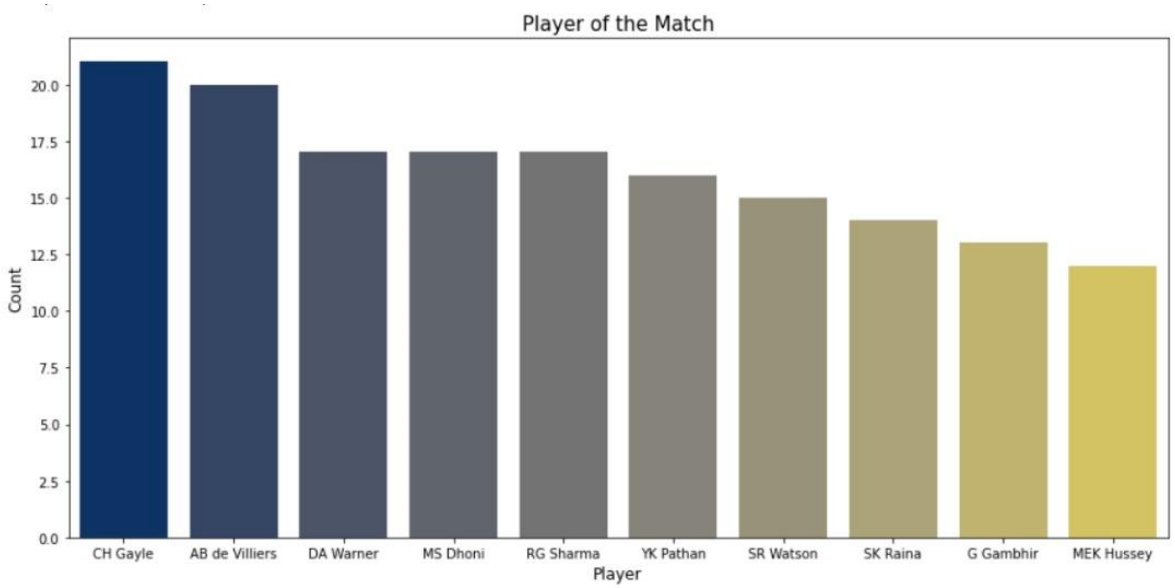


- **Observation:** From this pie chart, we can see that 61.2% of teams decided to field first and 38.8% of teams decides to bat first after winning the toss. So most of the teams feels that chasing is a better option than batting first.

Graph 6

- **Function code:** The Bar Plot of *matplotlib* library shows the top 10 players with highest Man of the Match Awards. Using *value_count()*, we have retrieved a series containing

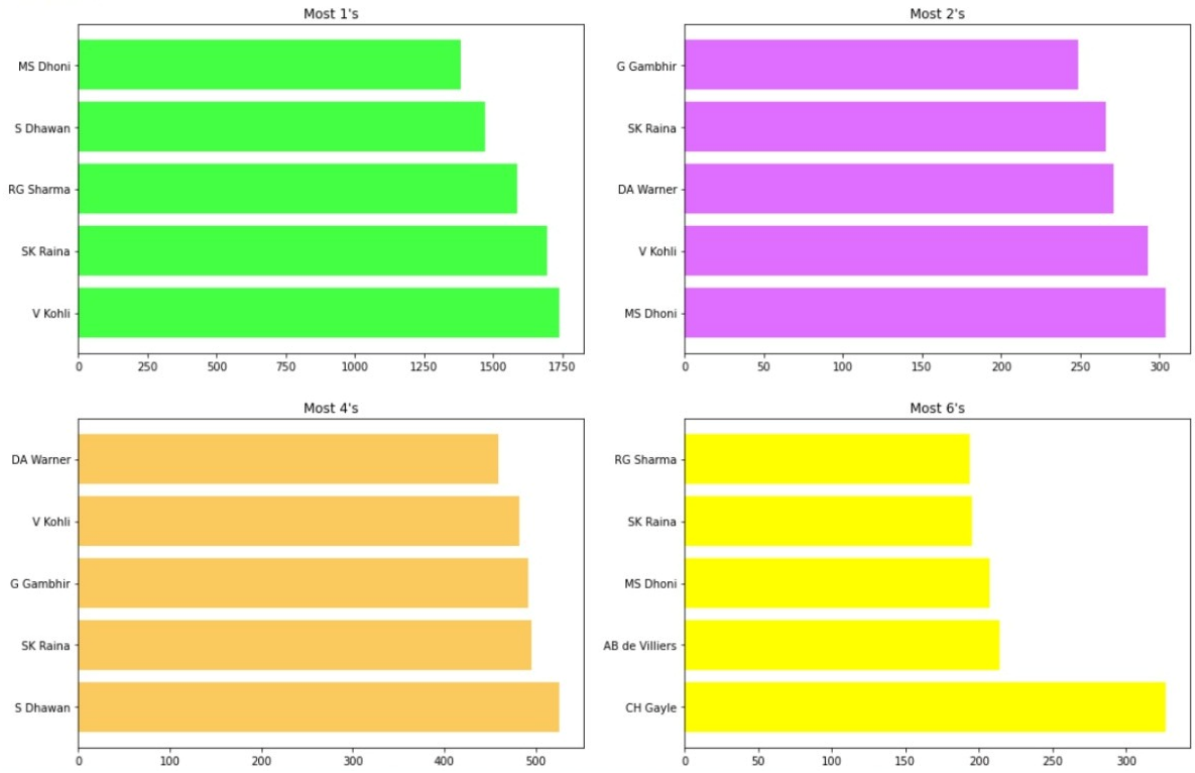
counts of unique values. After that we have used *seaborn* library for plotting the barplot with the use of *title*, *xlabel*, and *ylabel* and sliced top 10 values using *slice()* method.



- **Observation:** As we can observe that, Chris Gayle has won highest number of MoM awards which is 22 and AB de Villiers has won second highest MoM awards for 20 times till IPL 2019.

Graph 7

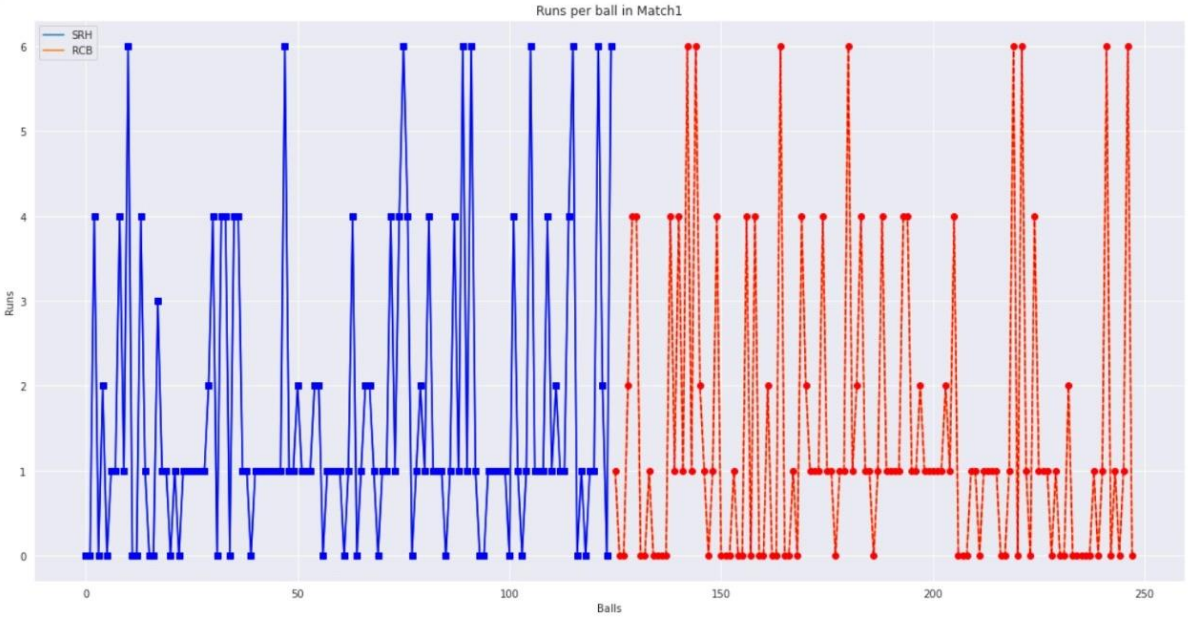
- **Function code:** We are visualising the data of Top 5 players with the highest singles, doubles, fours and sixes by plotting it in sub plots of *matplotlib* library. Using *groupby* function, we have formed the groups of all singles, doubles, fours and sixes with the batsman and counting the total times of all occurrences by using *count* function. After that by using *reset_index()* method to reset the index and using *pivot()* to produce pivot table based on 3 columns of the dataframe. Using *sort_values()* function, we are sorting the values in *ascending* order and keeping the bars *horizontal* using *kind='barh'* parameter. Also by using *width* parameter, we defined the width of each bar.



- **Observation:** From this graph, we can see that Virat Kohli has scored highest number of singles, MS Dhoni with highest number of doubles, Shikhar Dhawan has hit highest number of fours and Chris Gayle has hit highest number of sixes till IPL 2019.

Graph 8

- **Function code:** We have plotted a Line Chart using *seaborn* library by setting style to '*darkgrid'* showing the runs scored ball by ball in first match by both teams. We declared two variables to store the value of each *deliveries* of each team. After that, by using *plot()* method, we plotted both the variables and also setting the *xlabel*, *ylabel*, *legend* and *title* for the same by using *matplotlib* library.



- **Observation:** As we can observe that, it shows that on the 1st delivery of match, 0 run was scored and same for the last delivery also. It also shows the flow of runs in the whole match.

6. Conclusion and Future Work

- The approach has brought out analysis and visualization of various aspects of IPL matches in all the possible ways and gives useful results to the user. This information is of great value. It could be of great help to team owners (who purchase players for their teams in auction every year), captain and coaches to make the right selection for playing 11, to invest in the right team for betting and lastly for the people who are curious about IPL and its statistics.

- This project highlights the player performance especially batsmen and addresses the analysis that is done for Maximum Man of the Matches, Top Batsmen with most fours and sixes, Top 10 Players with Maximum Runs. Statistics of 756 matches have been used in this experiment and even for toss related analysis such as Count of Toss wins, Decision taken by each team after winning the toss, Toss Decision Team Wise. Based on the above analysis, the Indian batsmen are very good and are on top choice by the selectors. Selectors have the clear choice to give preference to Indian Players at first as they performed very well in season from 2008-2019. We also presented toss related analysis, in which Rohit Sharma is the best captain for MI who won the toss maximum times having count of 98. Selectors have the clear choice to select batsmen from Mumbai Indians and Kings XI Punjab as this two teams handled the pressure very well during all the seasons from 2008-2019. By considering all this visualization and toss related analysis. Team Management can select the right players and rights teams at the time of auction. A good and strong cricket team can be formed within a given budget, which will have the highest chance of winning.

- In future, making Machine Learning model can also be made to work with the ODI and test matches. The international matches can be analysed in a similar way and more visualizations can be added to the functions. The system can also be made to adapt more file formats of data for better analysis of varied forms of data collected. We can also add more data of recently completed seasons and can analyse it using various techniques and can get valuable information in order to predict the outcome of a player or a team performance in particular match based on venue and pitch.

7. Learning from the Project

- Sports Analytics is a game changer when it comes to how professional games are played, especially how strategic decision making happens, which until recently was primarily done based on “gut feeling" or adherence to past traditions. We learnt that numpy, seaborn, matplotlib and pandas forms a solid foundation for a large set of Python packages which provide higher level functions related to data analytics, machine learning, and AI algorithms. These packages are widely deployed to gain real-time insights that help in decision making for game-changing outcomes, both on field as well as to draw inferences and drive business around the game of cricket. Finding out the hidden parameters, patterns, and attributes that lead to the outcome of a cricket match helped us to take notice of game insights that are otherwise hidden in numbers and statistics.

- We also learned that numpy is the first step towards Data Analysis using Python, in which we learned about numpy arrays – creation, methods, and attributes, Basic math with arrays manipulation with arrays and using numpy for simulations. Data Analysis with pandas Series, dataframes & all operations with it. We also learnt about matplotlib which is needed for visualizing data. Also, we applied applications of Higher level libraries for plotting: seaborn and pandas. However, since both of these libraries are built on top of matplotlib we need to acquire the basic terminology and concepts of matplotlib because frequently we were supposed to make modifications to the objects and plots produced by those higher level libraries.

- We found that Exploratory data analysis is an approach to analysing data sets to summarize their main characteristics, often with visual methods. It is used to understand the data, get context about it, understand the variables and the relationship between them, and formulate hypothesis that could be useful when building predictive models.

8. Bibliography

ESPNcricinfo. (n.d.). Retrieved from https://www.espncricinfo.com/

GitHub. (n.d.). Retrieved from https://github.com/

IPL T20. (n.d.). Retrieved from https://www.iplt20.com/

Jovian. (n.d.). Retrieved from https://www.jovian.ai/

Kaggle. (n.d.). Retrieved from https://www.kaggle.com/akashkothare/tsf-datasets