# IPL Analysis

Mini Project Report

#### Duration: Six Weeks

**By**

**AYUSH PRASAD**

**Roll Number: 1900320100049**

### ASHISH KUMAR SINGH

#### Roll Number: 1900320100043

**College Name: ABES Engineering College**

**Under the guidance of**

**Mr. Anurag Mishra**

**Mr. Gopal Gupta**

# TABLE OF CONTENTS

Page No.

DECLARATION ii

[CERTIFICATE iii](#_TOC_250010)

ACKNOWLEDGEMENTS iv

[ABSTRACT v](#_TOC_250009)

[LIST OF TABLES viii](#_TOC_250008)

[LIST OF FIGURES ix](#_TOC_250007)

CHAPTER-1 INTRODUCTION 1

* 1. [Problem Definition 1](#_TOC_250006)
  2. [Motivation 1](#_TOC_250005)
  3. Objective of the Project 1
  4. Scope of the Project 1
  5. Need of Work 1

CHAPTER 2 RELATED WORK 2

CHAPTER 3 PROPOSED METHODOLOGY 3

* 1. [Dataset Description 3](#_TOC_250004)
  2. [Methods 4](#_TOC_250003)
  3. Hardware / Software Requirements 10

3.3 Our Methodology 10

CHAPTER 4 EXPERIMENT AND RESULT ANALYSIS 29

CHAPTER 5 CONCLUSION 34

* 1. [Discussion 34](#_TOC_250002)
  2. [Future Work 34](#_TOC_250001)

[REFERENCES 35](#_TOC_250000)

# CERTIFICATE

This is to certify that Project Report entitled “IPL Analysis” which is submitted by Ayush Prasad and Ashish Kumar Singh in partial fulfillment of the requirement for the summer internship of Data Analysis And Machine Learning in Department of Computer Science and Engineering of ABES Engineering College, is a record of the candidate own work carried out by him under my/our supervision.

**Mr. Gopal Gupta**

**Date: 10th of December, 2021**

# ACKNOWLEDGEMENT

*It gives us a great sense of pleasure to present the report of the Higher End Trainig undertaken during B.Tech, 3nd Year CSE. We owe special debt of gratitude to Mr. Gopal Gupta for his constant support and guidance throughout the course of our work. His constant motivation has been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.*

*We also do not like to miss the opportunity to acknowledge the motivation of Department Of Computer Science And Engineering, ABES Engineering College to provide us this opportunity*

*Name: Ayush Prasad*

*Roll No.:1900320100049*

*Date: 10th December 2021*

*Name: Ashish Kumar Singh*

*Roll No.: 1900320100043*

*Date : 10th December 2021*

# ABSTRACT

The project is based on that for a given IPL Analysis, which IPL player is best for the ipl team so that we can bulid best ipl team out of every team in IPL.

The data set we have worked upon has been downloaded from Kaggle.com.

Firstly, we analyzed the data set by simply opening the “csv” file. Then we pre-processed the whole data set to remove the insignificant columns and empty rows. Also, to get to know the all unique values of each attributes.

Further, we performed the data visualization on the data set. We performed Univariate analysis of each to get an idea about how the values are entered in each attribute with reference to no. of entries.

The support of an objective which will use the predictive and anaysied capabilities of the players to target best playing 11 including batsman ,blower, wicketkeeper and allrounders to pick best 11 for the auction or building best team possible.

**CHAPTER 1**

### INTRODUCTION

#### Problem Definition: -

* + - While making policies for a certain mass of people, to plan better services and life, it becomes easier to make it if we know the income of the people.
    - Therefore to make it easier we are using data analysis and ML model for income classification.
    - We are predicting whether a person has an income more or less than 50K a year using Census data.

#### Motivation:

* + - As already stated in our problem statement, this model helps in predicting income classification.
    - In current situation the world is hit by nightmare called COVID-19, which has caused a great blow to global economic which further has impacted many government and company policies.
    - By using the data analysis and machine learning model we will be predicting the income of an individual which when further enhanced will give us an average income of a region which can help in making policies for betterment of public.
  1. **Objective of the Project:** The Project is based on a machine learning model, that can predict if a person has an income over 50K or not, based on different attributes from a census dataset.
  2. **Scope of the Project:** The scope of the project is very vast, as it can be used in many fields of administration. As already stated, that it can be used in policy making as well categorizing individuals.
  3. **Need of Work: -** The support of an objective which will use the predictive capabilities of the proposed models to target individuals is reviewed including the use of lift analysis to indicate the likely level of return on investment.
  4. **Hardware / Software Requirements Minimums Hardware Requirements:**

RAM: 2 GB

Processor: Intel Core 2 Duo Hard disk: 50GB

#### Minimums Software Requirements:

Pandas: 0.24.2

Numpy; 1.16.4

Matplotlib: 3.1.0

Seaborn: 0.9.0

Python: 3.7

Scikit-Learn: 0.21.2

#### Our Methodology:

* Firstly, we downloaded the data set from UCI machine learning repository of income classification.
* Then we loaded the data set into Jupyter Notebook

#### Fig. 3.7: Importing numpy and pandas

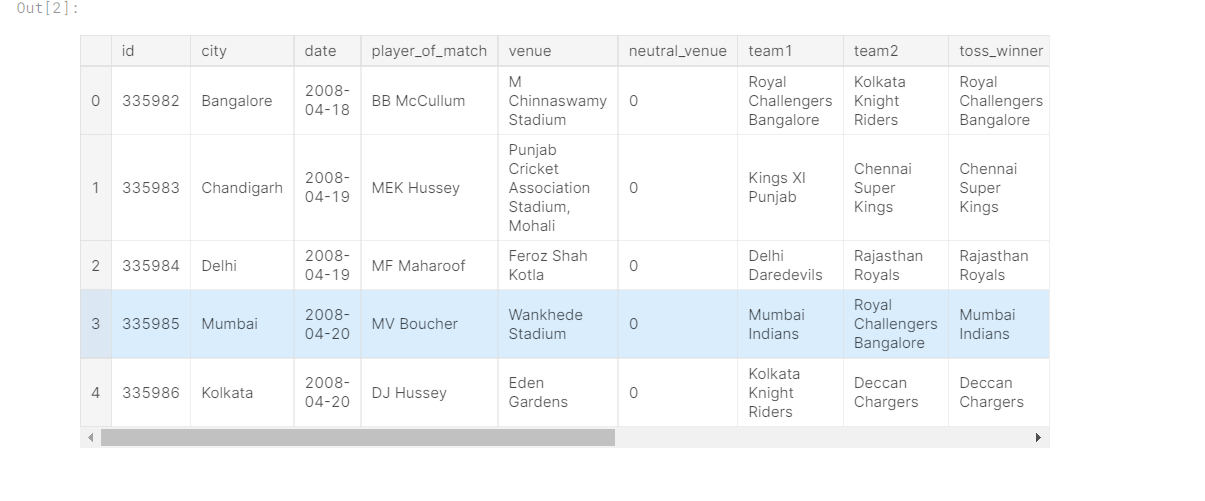
Importing numpy and pandas packages.

NumPy is a python library used for working with arrays. Pandas is a high-level data manipulation tool.



#### Fig. 3.8: read\_csv()

* **Describing the data:**



#### Fig. 3.9: describe() function

## ****Import Libraries****

import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns%matplotlib inline

## ****Load the data****

match=pd.read\_csv('D:\Data Science\IPL\Dataset\matches.csv')  
delivery=pd.read\_csv('D:\Data Science\IPL\Dataset\deliveries.csv')

Load the data using the above code in your Jupyter Notebook. For Google Colab you have to upload the dataset to your drive and then import it.

## ****Analyse the Data****

Now take a look at the data we are working on:

match.head(5)  
delivery.head(5)

When you run the shell then in the match section you will see the first match back in 2008 was played between KKR and RCB. KKR has won the match at M Chinnaswamy Stadium and the Player of the Match was BB McCullum. The match result was decided by runs.

The same kind of analysis is present in the ball delivery section too. If you want to look at the top 5 bottom data of the table and then you have to run the program match.tail(5) in the Jupyter Notebook or Google Colab cell.

## ****More Information about Matches and Ball Deliveries between 2008–2020****

match.info() #816  
delivery.info() #193468

## ****List of the Participating Teams****

all\_teams = match['team1'].tolist() + match['team2'].tolist()  
all\_teams = list(set(all\_teams))  
all\_teams

You will get the list of teams that played between the period 2008 to 2020. If you are a pro IPL fan then you will see some old team names on the list which are not playing these days but they contributed some valuable information in the IPL history.

## ****Number of Matches per Venue****

sns.countplot('venue', data=match)  
plt.xticks(rotation='vertical')

As you see that Eden Gardens is the fan-favourite ground of IPL, nearly 80 matches have been hosted there.

## ****Matches Played by Each Team****

x = match['team1'].value\_counts()  
y = match['team2'].value\_counts()  
(x+y).plot(kind='barh')

We count the value of each team playing in column one and add to the count of each team from team two to get the desired output. For example, if CSK played 90 times from team one and 85 times from team 2 then the total of175 matches are shown in the graph. You can see that Mumbai Indians played the highest number of matches in the IPL.

## ****Matches Won by Each Team****

x=pd.DataFrame({"Winner":match['winner']}).value\_counts()  
print(x)

When you run this cell you will see that Mumbai Indians win the highest number of matches followed by CSK and other teams. Now if you want to plot this result in graph form then run this program in the next cell.

sns.countplot('winner', data=match)  
plt.xticks(rotation='vertical')

## ****Top 5 Players with the Highest Number of Man of the Match Awards****

If you are a team management official and these players go under the hammer then you must have to keep eye on these players as these players have the highest number of Man of the Match awards.

Let’s check out how to find this:

temp\_data=match['player\_of\_match'].value\_counts().head()  
print(temp\_data)  
sns.barplot(x=temp\_data.index,y=temp\_data.values,data=match)plt.title("Top 5 MoM")  
plt.xticks(rotation=90)  
plt.xlabel("Match Count")  
plt.ylabel("Player")  
plt.show()

Is your favourite player present in the above list?

## ****The Top Batsman in the IPL****

For this, we have to find out the player with the highest number of runs. To find out this, we have to sum up the batsman’s run from the delivery dataset and the batsman who scored that run. It’s simple logic, right?

Demonstrated below:

top\_batsman=delivery.groupby('batsman')['batsman\_runs'].agg('sum').reset\_index().sort\_values('batsman\_runs', ascending=False).head(10)top\_batsman.set\_index('batsman', inplace=True)  
top\_batsman.plot(kind='bar')

We grouped the top 10 batsmen from the delivery dataset and summed up their runs. After this, we plot this information into a graph.

King Kohli is at the top followed by Suresh Raina and other batsmen.

## ****The Bowler Who Has Given the Highest Number of Runs****

delivery.groupby('bowler')['total\_runs'].agg('sum').reset\_index().sort\_values('total\_runs', ascending=False).head(10)

For this, we grouped the top 10 bowlers who have given runs on his delivery in IPL matches and summed up that run for the final outlet.

## ****The Bowler with Team-wise Performance****

Let’s suppose you are playing against CSK and you have to find out which bowler’s performance was good in the previous years against this team. To find out the team-wise performance analysis of a bowler, you have to run the following program in your Notebook cell:

mask=delivery['bowler']=='PP Chawla'  
delivery[mask].groupby('batting\_team')['total\_runs'].agg('sum').plot(kind='bar')

We are taking the example of PP Chawla. This bowler has given the highest number of runs in the IPL history till 2020. We summed up the total runs given by PP Chawla to the opponent team.

PP Chawla's bowling performance against IPL teams.

It’s clear that if you have PP Chawla in your team then don’t let him play against MI, CSK, RCB, RR, and DC.

## ****Over-wise Batting Performance of Each Team in the IPL (2008–2020)****

delivery6=delivery[mask]  
delivery6=delivery6[['batting\_team','over','batsman\_runs']]  
x=delivery6.pivot\_table(values='batsman\_runs', index='batting\_team', columns='over', aggfunc='count')sns.heatmap(x, cmap='summer')

For this, we are using a pivot table and then count the over-wise run of batsmen of the batting team. Then convert the data into a heatmap as given below:

As you can see, if you are playing against MI or CSK, then you have to play with your best bowling attack line-up from the first over. MI’s batsmen are silent in the second and third over, after that they go on rampage mode against their opponent. The same goes for CSK and RCB too. This data is not only helpful from the bowling team’s perspective but also the batting team. If you are a team manager and you see using this data that your team is not performing well in the death overs then you probably should focus on buying a good finisher in the next auction. As you see in the above heatmap, most of the teams are lagging at the end of the map, except CSK and MI.

I think that’s why MI and CSK are the two most successful franchises in the IPL.

## ****Dismissal Kind****

sns.countplot('dismissal\_kind', data=delivery)  
plt.xticks(rotation='vertical')

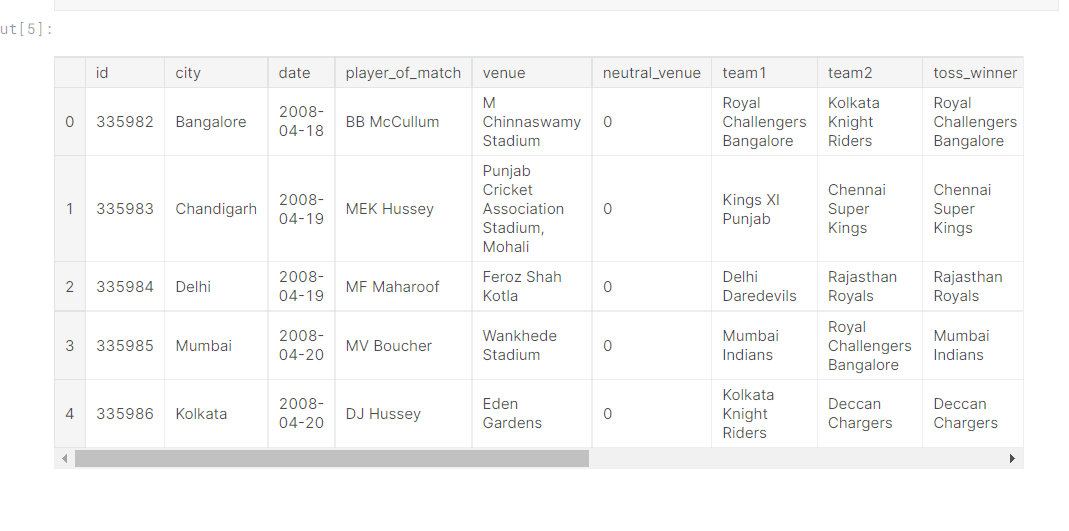
Dismissal kind in IPL(2008–2020)

Now, if you want to know that how many runs Virat Kohli scored when he faced Jasprit Bumrah, use the following code:

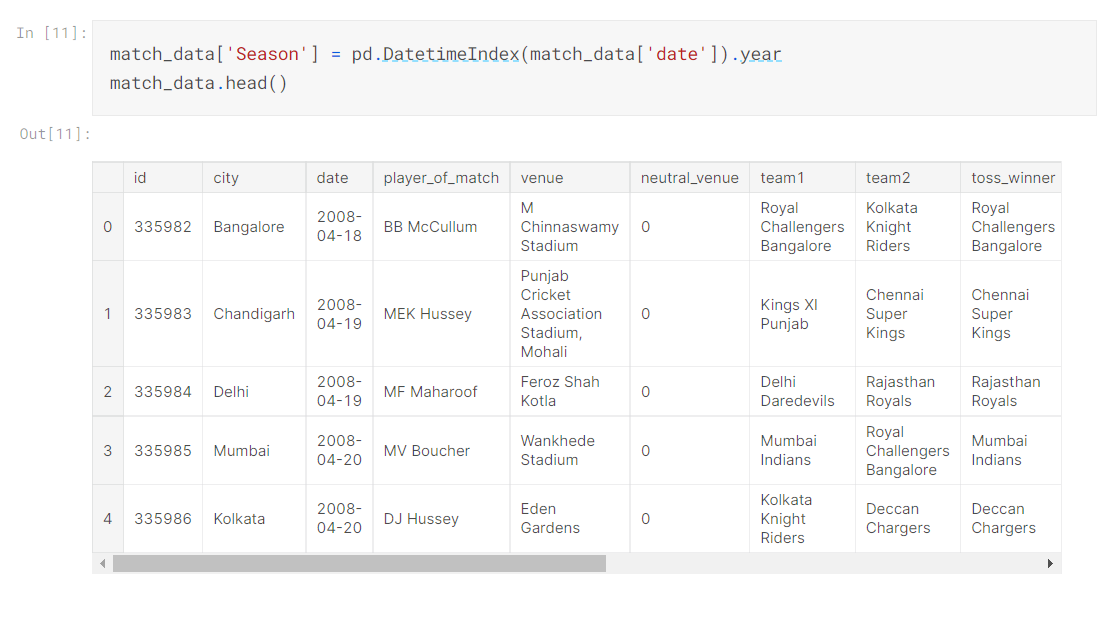
mask=delivery['bowler']=='JJ Bumrah'  
mask2=delivery['batsman']=='V Kohli'  
delivery[mask].groupby('batsman')['batsman\_runs'].agg('count').sort\_values(ascending=False)['V Kohli']

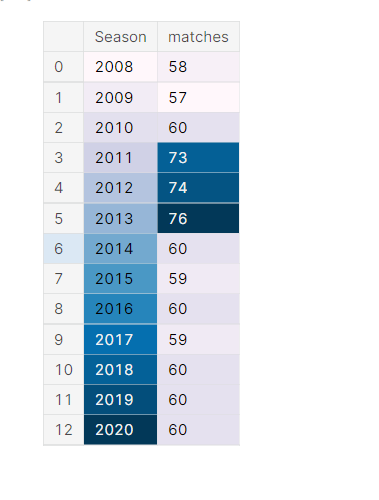
Sum of the run when the bowler is Bumrah and the batsman is V Kohli. You will get the output.

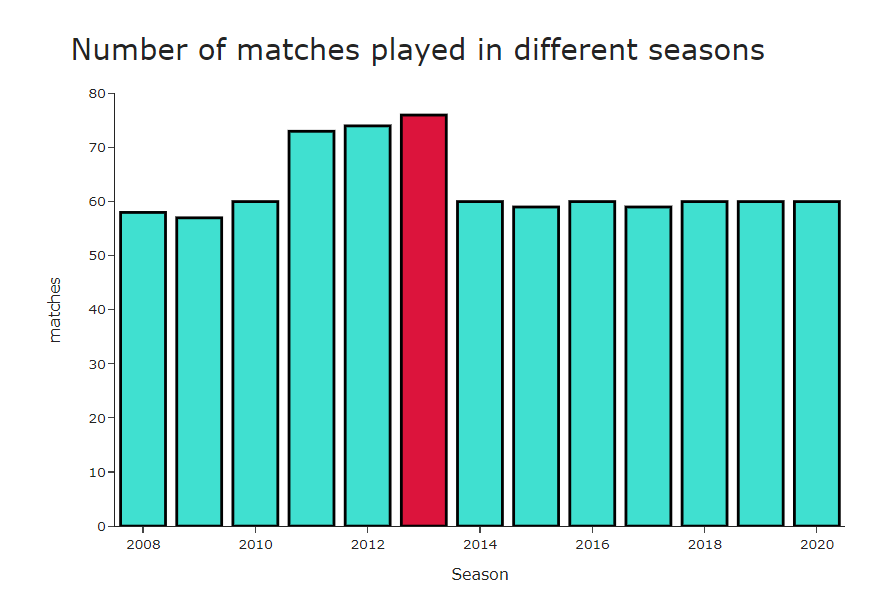
# Basic data exploration:



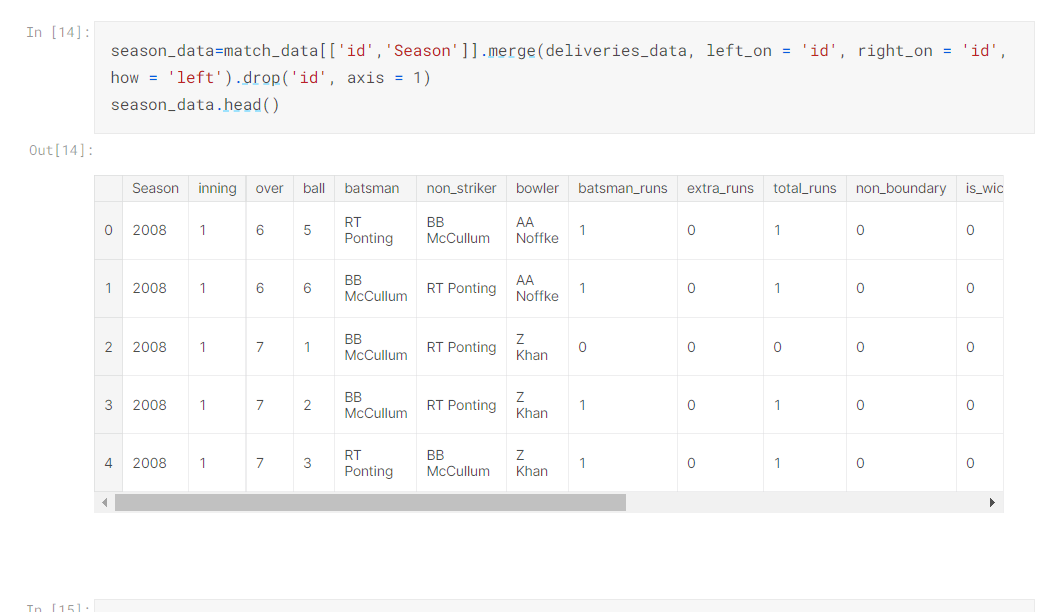
# Number of matches played in various seasons :

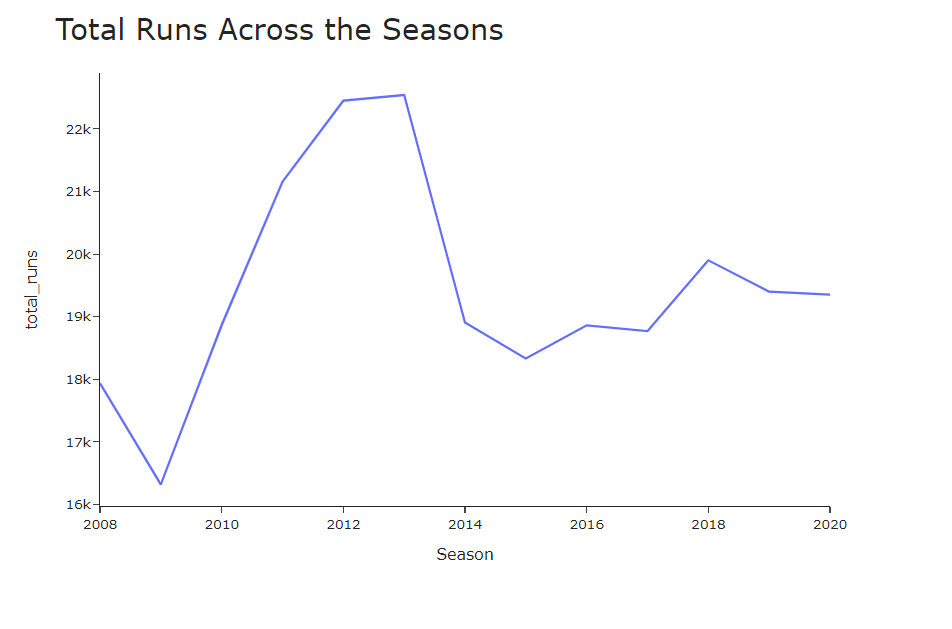




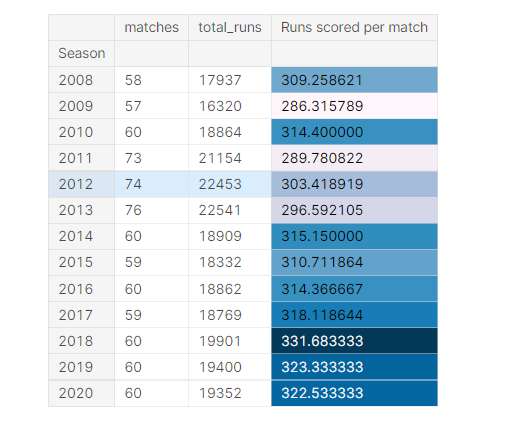


# Total number of runs scored across seasons :

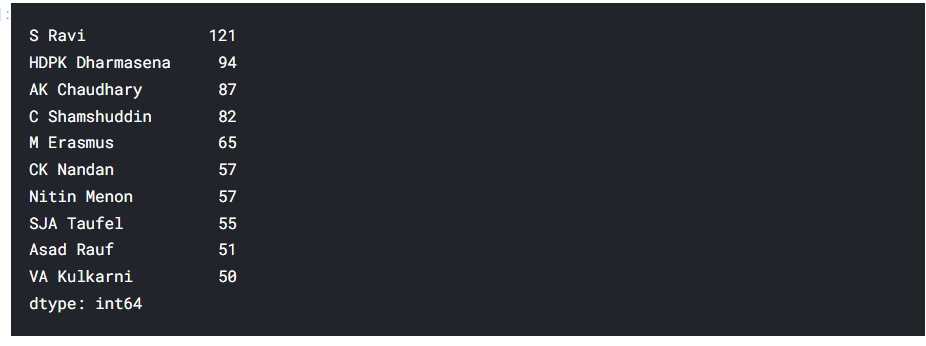




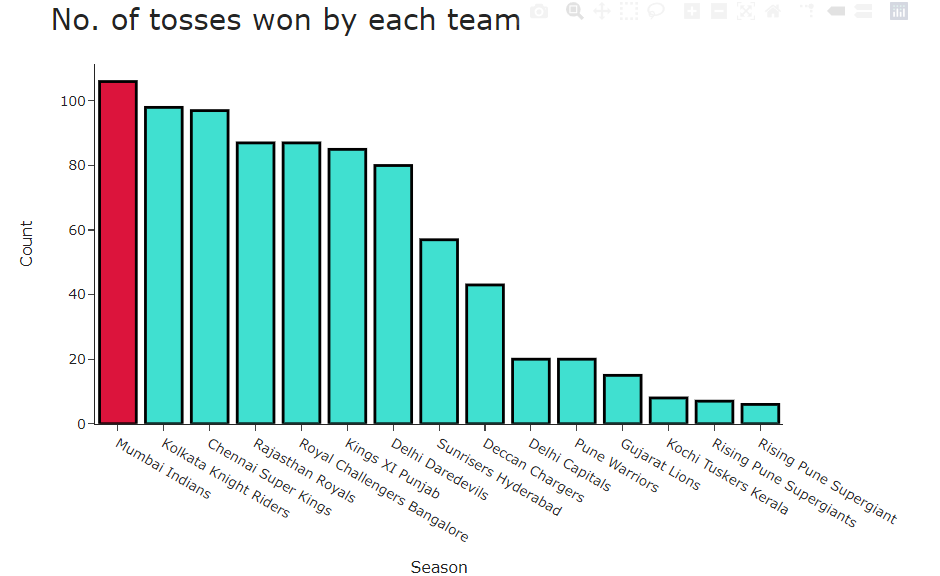
# Runs scored per match across seasons :



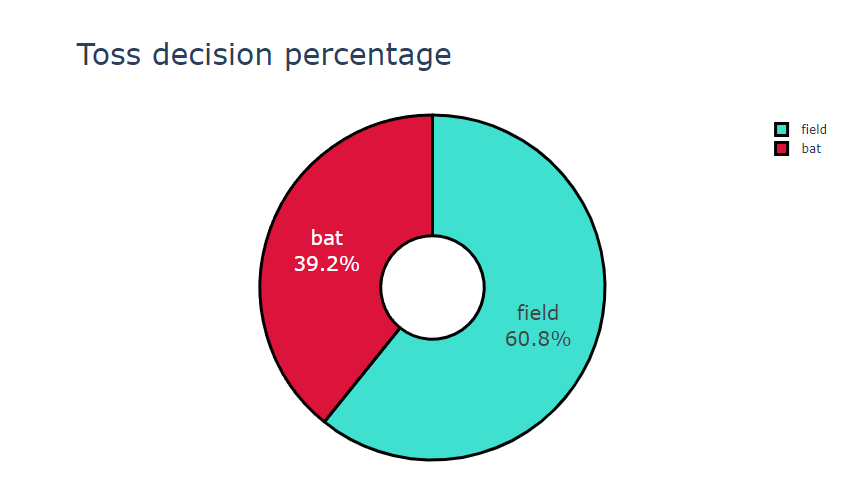
# Count of matches by umpires :

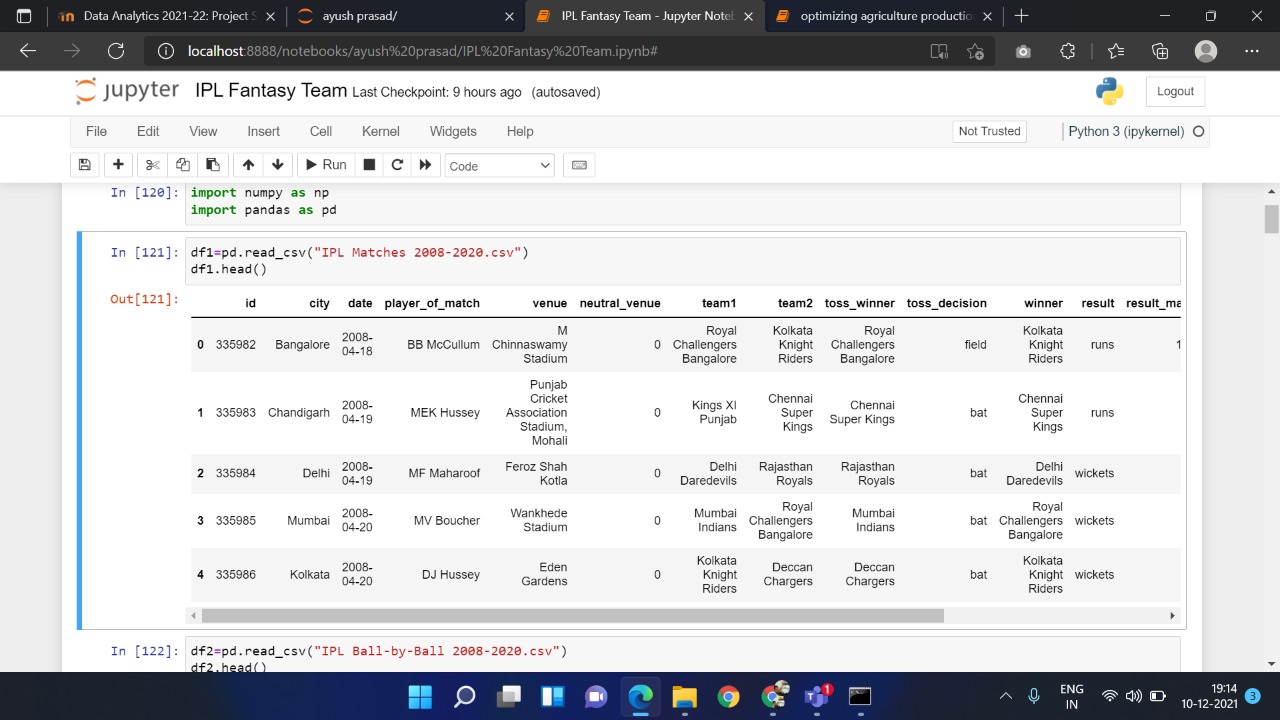


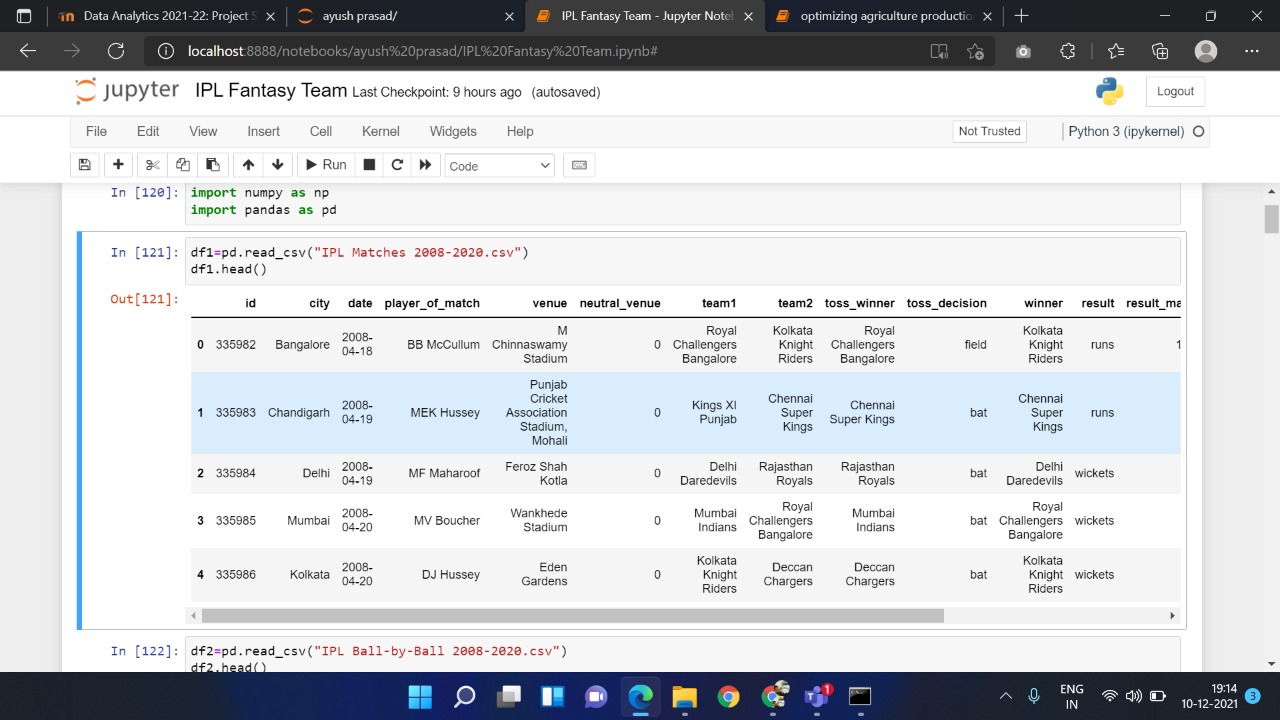
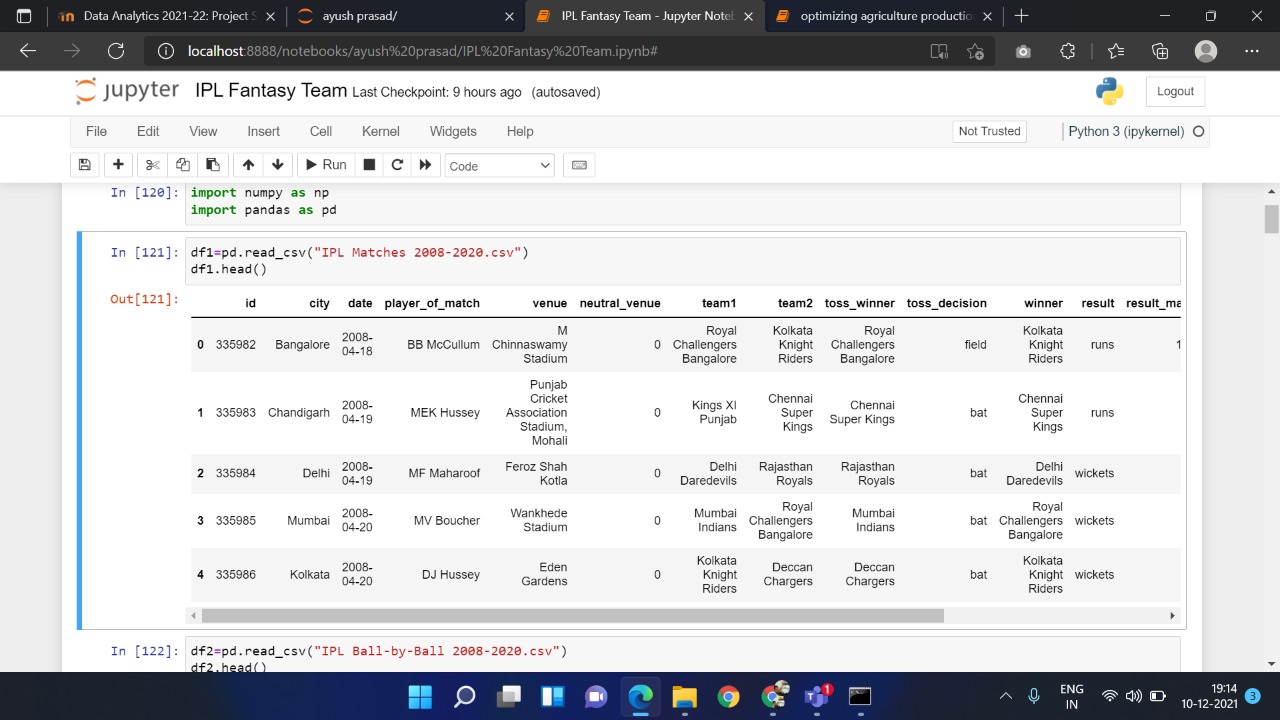
# Number of tosses won by teams :

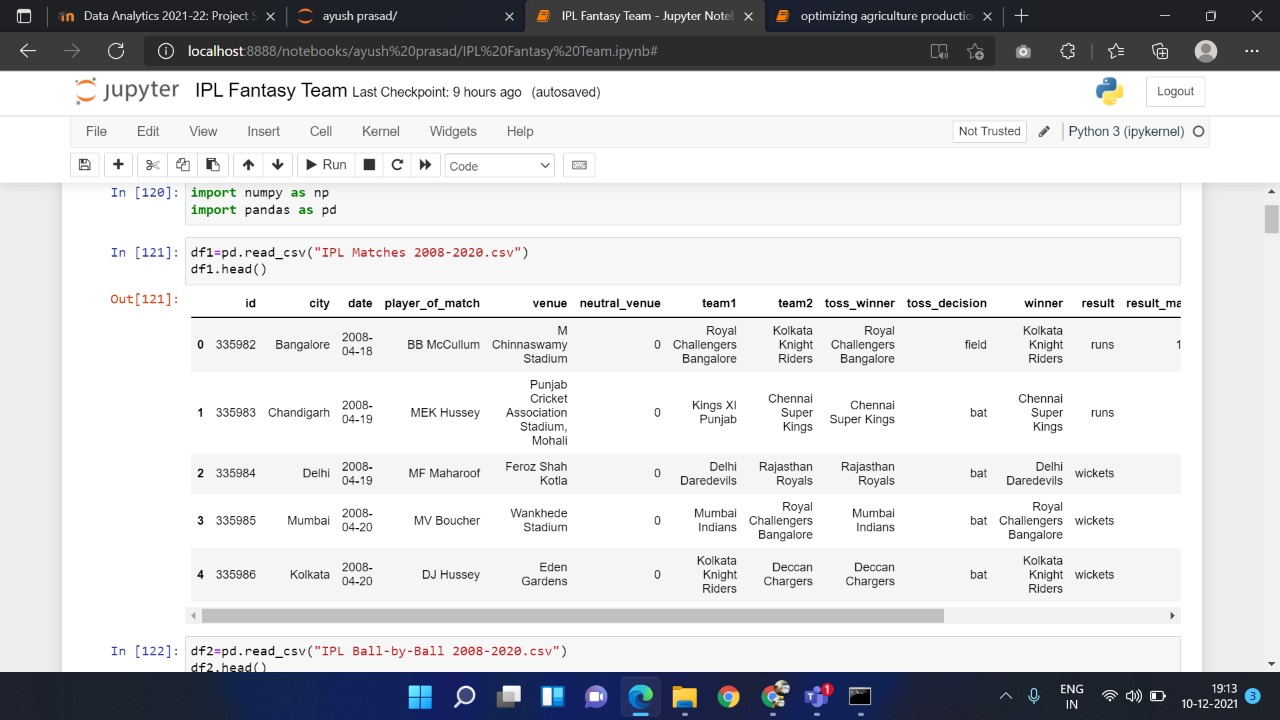
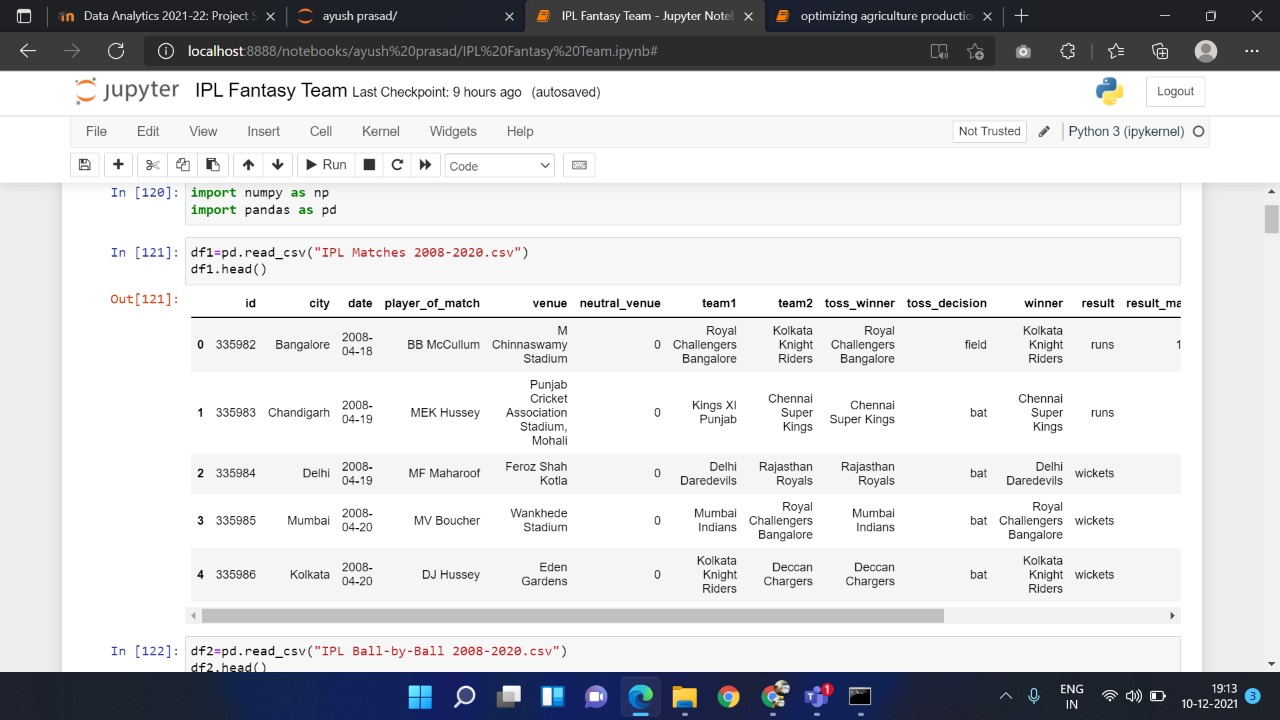


# Decision made after winning the toss









**Downloading data set:**

## REFERENCES

<https://archive.ics.uci.edu/ml/machine-learning-databases/adult/> <http://abhigyan.dataritz.com/mod/folder/view.php?id=602> **Loading and analyzing**

<https://pandas.pydata.org/>

<https://numpy.org/>

### Pre-processing

<https://towardsdatascience.com/data-preprocessing-in-python-b52b652e37d5> <https://www.geeksforgeeks.org/data-preprocessing-machine-learning-python/> <https://scikit-learn.org/stable/modules/preprocessing.html>

[https://www.tutorialspoint.com/machine\_learning\_with\_python/machine\_learning\_with\_python\_dat](https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_data_preprocessing_analysis_visualization.htm) [a\_preprocessing\_analysis\_visualization.htm](https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_data_preprocessing_analysis_visualization.htm)

<https://www.datacamp.com/courses/preprocessing-for-machine-learning-in-python>

**Data Visualization** <https://matplotlib.org/tutorials/index.html> <https://python-graph-gallery.com/seaborn/> **Machine Learning Algorithm**

<https://www.datacamp.com/community/tutorials/understanding-logistic-regression-python>

<https://www.datacamp.com/community/tutorials/random-forests-classifier-python> <https://www.datacamp.com/community/tutorials/decision-tree-classification-python> <https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn> **GUI- Tkinter**

<https://realpython.com/python-gui-tkinter/>