

Comprehensive Analysis of IPL Auction Team Selection

MINOR PROJECT REPORT

By

**DHIVYA(RA2211056010021)
RITI MATANGI(RA2211056010035)
ROYCE MARK(RA2211056010040)**

Under the guidance of

Dr.M.Anand

In partial fulfilment for the Course

of

21CSS202T– FUNDAMENTALS OF DATA SCIENCE

in DATA SCIENCE AND BUSINESS SYSTEMS



FACULTY OF ENGINEERING AND TECHNOLOGY

SCHOOL OF COMPUTING

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR

NOVEMBER 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this minor project report for the course **21CSS202T FUNDAMENTALS OF DATA SCIENCE** entitled in "Comprehensive Analysis of Diwali Sales Trends using Random Forest, Linear Regression and Time Series Analysis" is the bonafide work of **DHIVYA(RA2211056010021)**, **RITI MATANGI(RA2211056010035)** and **ROYCE MARK (RA2211056010040)** who carried out the work under my supervision.

SIGNATURE

Dr.M. Anand

Assistant Professor

DATA SCIENCE AND BUSINESS SYSTEM

SRM Institute of Science and Technology

Kattankulathur

ABSTRACT

The IPL Auction Analysis project merges the worlds of cricket and data science to unravel the intricacies of team dynamics in the Indian Premier League (IPL). Focused on the captivating player auction, our study employs statistical analysis and machine learning to scrutinize historical data, uncovering patterns that guide franchise decision-making.

Key objectives include understanding player valuation factors, optimizing team compositions, and discerning effective budget allocation strategies. Leveraging diverse datasets encompassing player statistics and auction history, our project aims to develop predictive models for player valuations and offer actionable insights for team strategists.

By exploring the intersections of cricket analytics and data science, we aspire to contribute novel perspectives to the field, providing tangible recommendations for IPL franchises. This journey encapsulates the fusion of tradition and technology, propelling cricket team strategies into a new era of data-driven decision-making.

This code conducts a detailed analysis of Indian Premier League (IPL) player data, employing the Pandas, NumPy, Matplotlib, and Seaborn libraries in Python. The script begins by loading IPL data from a CSV file and proceeds to clean and organize it, focusing on key player types: batters, bowlers, all-rounders, and keepers.

Following data cleaning, the code delves into an in-depth exploration of each player category, identifying top performers based on various performance metrics such as runs, average, strike rate, wickets, and more. The analysis is then visualized through bar plots, providing a clear and concise representation of the top players in each category.

The script concludes with a strategic approach to forming the best 11 for a hypothetical campaign. By considering the player compositions of successful teams, the code suggests an optimal balance of batters, bowlers, all-rounders, and keepers for an effective and competitive lineup.

ACKNOWLEDGEMENT

We express our heartfelt thanks to our honorable **Vice Chancellor Dr. C. Muthamizhchelvan**, for being the beacon in all our endeavors.

We would like to express my warmth of gratitude to our **Registrar Dr. S. Ponnusamy**, for his encouragement.

We express our profound gratitude to our **Dean (College of Engineering and Technology) Dr. T. V. Gopal**, for bringing out novelty in all executions.

We would like to express my heartfelt thanks to Chairperson, School of Computing **Dr. Revathi Venkataraman**, for imparting confidence to complete my course project

We wish to express my sincere thanks to **Course Audit Professors Dr. Vadivu. G, Professor, Department of Data Science and Business Systems and Dr. Sasikala. E Professor, Department of Data Science and Business Systems and Course Coordinators** for their constant encouragement and support.

We are highly thankful to my Course project Faculty **Dr.M.Anand Assistant Professor, Data Science and Business Systems**, for his/her assistance, timely suggestion and guidance throughout the duration of this course project.

We extend my gratitude to our **HoD Dr. M Lakshmi, Professor, Department of Data Science and Business Systems** and my Departmental colleagues for their Support.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on me to complete my Course project.

TABLE OF CONTENTS

CHAPTER NO	CONTENTS	PAGE NO
1	INTRODUCTION	
	1.1 Motivation	
	1.2 Objective	
	1.3 Problem Statement	
	1.4 Challenges	
2	LITERATURE SURVEY	
3	REQUIREMENT ANALYSIS	
4	ARCHITECTURE & DESIGN	
5	IMPLEMENTATION	
6	EXPERIMENT RESULTS & ANALYSIS	
7	CONCLUSION	
8	REFERENCES	

1. INTRODUCTION

In the dynamic realm of cricket, the Indian Premier League (IPL) stands as a beacon of innovation, blending athleticism with strategic team-building. At the heart of this spectacle lies the IPL auction, a riveting event where franchises bid for players to construct competitive squads. This project delves into the intersection of cricket and data science, aiming to dissect the nuances of the IPL auction through statistical analysis and machine learning.

Crucial questions surround player valuation, team composition, and budget allocation. By leveraging historical auction data, player performance metrics, and cutting-edge analytical tools, this endeavor seeks to unearth patterns that inform decision-making for IPL franchises. From predictive modeling to uncovering the impact of player attributes, our journey involves bridging cricketing intuition with data-driven insights, offering practical recommendations for team strategists and stakeholders.

As we navigate through the literature and embark on a comprehensive analysis, the goal is to contribute not only to the burgeoning field of cricket analytics but to enhance the efficacy of IPL team strategies, shaping the future landscape of this captivating sport.

This code conducts a detailed analysis of Indian Premier League (IPL) player data, employing the Pandas, NumPy, Matplotlib, and Seaborn libraries in Python. The script begins by loading IPL data from a CSV file and proceeds to clean and organize it, focusing on key player types: batters, bowlers, all-rounders, and keepers.

Following data cleaning, the code delves into an in-depth exploration of each player category, identifying top performers based on various performance metrics such as runs, average, strike rate, wickets, and more. The analysis is then visualized through bar plots, providing a clear and concise representation of the top players in each category.

2.LITERATURE SURVEY

1. Introduction to Cricket Analytics:

- Explore foundational literature on the application of analytics in cricket.
- Highlight key studies that have employed statistical methods and machine learning in cricket data analysis.

2. T20 Cricket and Player Performance Metrics:

- Review research on the specific metrics used to evaluate player performance in T20 cricket.
- Examine studies that correlate player statistics with team success in T20 leagues globally.

3. Auction Strategies in Sports:

- Investigate literature on auction dynamics in sports, with a focus on cricket leagues and player auctions.
- Analyze successful strategies adopted by teams in various sports auctions and their applicability to the IPL.

4. Player Valuation Models:

- Survey existing models and methodologies for predicting player valuations in sports auctions.
- Explore how statistical models have been used to assess player worth based on performance metrics.

5. Budget Allocation in Team Sports:

- Examine literature on budget allocation strategies in team sports, emphasizing cricket leagues.
- Identify patterns and trends in budget distribution among successful sports franchises.

6. Machine Learning in Sports Analytics:

- Review studies showcasing the use of machine learning in sports analytics, particularly in cricket.
- Assess the accuracy and effectiveness of predictive models in forecasting player

performances and auction values.

7. Integration of Cricket and Data Science:

- Explore how data science and analytics have been integrated into cricket decision-making processes.
- Identify gaps in existing research and propose how this project contributes to the field.

8. Challenges and Opportunities in Cricket Analytics:

- Discuss challenges faced by analysts and researchers in applying data science to cricket.
- Highlight opportunities for advancements in analytics within the context of T20 leagues and player auctions.

1. REQUIREMENTS

3.1 Requirement Analysis:

CSV FILE:

These requirements are based on:

- STRIKE RATE
- MATCHES PLAYED
- ECONOMY
- AVERAGE
- RUNS
- PLAYERS NAME AND TEAM

Based on the above requirements gathering information in the CSV file.

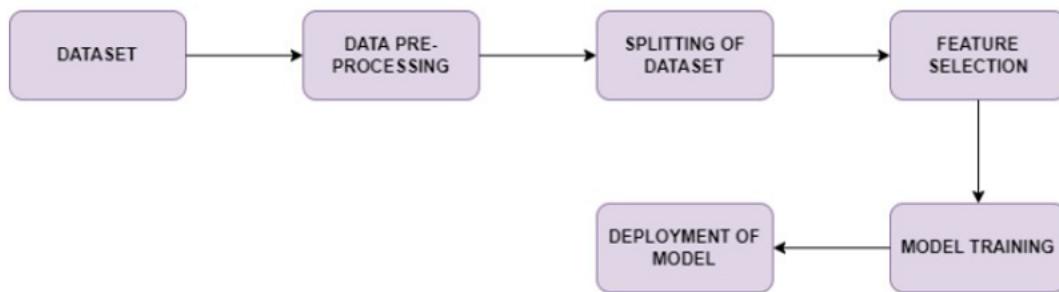
1.1 Software Requirement:

- Jupyter Notebook
- CSV file

2. ARCHITECTURE AND DESIGN

1.2 Network Architecture

The network architecture is as follows:



Dataset: Since a machine learning model is entirely based on data, the very first thing we need to generate one is a dataset.

Data Pre-Processing: Pre-processing data is the first and most significant step in building a machine learning model.

Splitting of Dataset: Two distinct datasets are not imported for train and test to prevent overfitting.

Scaling of Feature: The scaling of features is a technique for converting data into a precise and customizable size in order to improve the precision and avoid errors.

Model Construction: The dataset is now capable of building the conceptual approach after completing the initial phases.

2. IMPLEMENTATION

```
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns  
  
data=pd.read_csv("IPLData.csv")
```

```
dataimport pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns  
  
data.describe().transpose()
```

```
data.isna().sum()
```

```
data.info()
```

CLEANING THE DATA

```
batters = data.loc[(data["Player_Type"] == "Batter")]  
batters_new = batters.loc[(batters["Capped"] == 1)]  
Capped_Batters = batters_new[['Player Name',  
'Team',  
'Nationality',  
'Matches_Played',  
'Runs',  
'Average',  
'Strike_Rate']]
```

```
Capped_Batters.head(10)
```

```
bowlers = data.loc[(data["Player_Type"] == "Bowler")]
bowlers_new = bowlers.loc[(bowlers["Capped"] == 1)]
Capped_Bowlers = bowlers_new[['Player Name',
'Team',
'Nationality',
'Matches_Played',
'Wickets',
'Bowling_average',
'Economy',
'Bowling_Strike_Rate']]
```

```
Capped_Bowlers.head(10)
```

```
Keepers = data.loc[(data["Player_Type"] == "Keeper")]
Keepers_new = Keepers.loc[(Keepers["Capped"] == 1)]
Capped_Keeprers = Keepers_new[['Player Name',
'Team',
'Nationality',
'Matches_Played',
'Runs',
'Average',
'Strike_Rate',
'Catches',
'Run_outs',
'Stumps']]
```

```
Capped_Keeprers.head(10)
```

```
Allrounders = data.loc[(data["Player_Type"] == "Allrounder")]
Allrounders_new = Allrounders.loc[(Allrounders["Capped"] == 1)]
Capped_Allrounders = Allrounders_new[['Player Name',
'Team',
'Nationality',
'Matches_Played',
'Runs',
```

```
'Average',
'Strike_Rate',
'Wickets',
'Bowling_average',
'Economy',
'Bowling_Strike_Rate']]
```

```
Capped_Allrounders.head(10)
```

```
Capped_Batters = Capped_Batters.fillna(0)
Capped_Bowlers = Capped_Bowlers.fillna(0)
Capped_Allrounders = Capped_Allrounders.fillna(0)
Capped_Keepers = Capped_Keepers.fillna(0)
```

```
print(Capped_Batters.isna().sum())
print(Capped_Bowlers.isna().sum())
print(Capped_Allrounders.isna().sum())
print(Capped_Keepers.isna().sum())
```

INITIAL ANALYSIS

```
#Narrowing our analysis to batters who have a batting average more than 32
top_batters = Capped_Batters.loc[(Capped_Batters["Average"] >= 32.0)]

#Sorting the data in descending order - with respect to each parameter
top_batters_average = top_batters.sort_values('Average', ascending=False)
top_batters_strike_rate = top_batters.sort_values('Strike_Rate', ascending=False)
top_batters_runs = top_batters.sort_values('Runs', ascending=False)
top_batters_matches = top_batters.sort_values('Matches_Played', ascending=False)
```

```
top_batters_average
```

```
top_batters_strike_rate
```

```
top_batters_runs
```

```
top_batters_matches
```

From our analysis, if we rank from 0-10, the top three batters that will come while analysis each of the above data are:

1. KL Rahul
2. David Warner
3. Virat Kohli

```
#here we have narrowed our analysis based on the bowling averages of the players to be less than 24.0
```

```
top_bowlers = Capped_Bowlers.loc[(Capped_Bowlers ["Bowling_average"] <= 24.0)]
```

```
top_bowlers_average = top_bowlers.sort_values ('Bowling_average')
```

```
top_bowlers_strike_rate = top_bowlers.sort_values ('Bowling_Strike_Rate')
```

```
top_bowlers_wickets = top_bowlers.sort_values ('Wickets', ascending=False)
```

```
top_bowlers_economy = top_bowlers.sort_values ('Economy')
```

```
top_bowlers_matches = top_bowlers.sort_values ('Matches_Played', ascending=False)
```

```
top_bowlers_average
```

```
top_bowlers_strike_rate
```

```
top_bowlers_wickets
```

```
top_bowlers_economy
```

```
top_bowlers_matches
```

From the above analysis, if we rank down the parameters from 1-10. The top bowling options are as follows:

1. Kagiso Rabada
2. Jasprit Bumrah
3. Yuzvendra Chahal
4. Nathan Coulter-Nile

Analyzing the Allrounder Data

We narrowed our analysis by further segregating the allrounders based on strike rate equal to or more than 140.0.

```
top_allrounders = Capped_Allrounders.loc[(Capped_Allrounders["Strike_Rate"] >= 140.0)]
top_allrounders_average = top_allrounders.sort_values('Average', ascending=False)
top_allrounders_strike_rate = top_allrounders.sort_values('Strike_Rate', ascending=False)
top_allrounders_runs = top_allrounders.sort_values('Runs', ascending=False)
top_allrounders_matches = top_allrounders.sort_values('Matches_Played', ascending=False)
top_allrounders_bowling_average = top_allrounders.sort_values('Bowling_average')
top_allrounders_bowling_strike_rate = top_allrounders.sort_values('Bowling_Strike_Rate')
top_allrounders_wickets = top_allrounders.sort_values('Wickets', ascending=False)
top_allrounders_economy = top_allrounders.sort_values ('Economy')
top_allrounders_matches = top_allrounders.sort_values ('Matches_Played', ascending=False)
```

top_allrounders_average

top_allrounders_strike_rate

top_allrounders_runs

top_allrounders_matches

top_allrounders_bowling_average

top_allrounders_bowling_strike_rate

top_allrounders_wickets

top_allrounders_economy

From the above analysis, if we rank down the allrounders from 1-10 on various parameters. The top allrounder options are as follows

1. Andre Russell
2. Sunil Narine
3. Hardik Pandya
4. Jofra Archer

#Analyzing the Keepers Data

#we have narrowed our analysis down to keepers averaging more than 25.0

```
top_keepers = Capped_Keepers.loc[(Capped_Keepers["Average"] >= 25.0)]
```

#Sorting the data in descending order - with respect to each parameter.

```
top_Keepers_average = top_keepers.sort_values ('Average', ascending=False)
```

```
top_Keepers_strike_rate = top_keepers.sort_values('Strike_Rate', ascending=False)
```

```
top_Keepers_runs = top_keepers.sort_values ('Runs', ascending=False)
```

```
top_Keepers_matches = top_keepers.sort_values( 'Matches_Played', ascending=False)
```

```
top_Keepers_catches = top_keepers.sort_values ('Catches', ascending=False)
```

```
top_Keepers_runouts = top_keepers.sort_values('Run_outs', ascending=False)
```

```
top_Keepers_stumps = top_keepers.sort_values ('Stumps', ascending=False)
```

```
top_Keepers_average
```

```
top_Keepers_strike_rate
```

```
top_Keepers_runs
```

```
top_Keepers_matches
```

```
top_Keepers_catches
```

```
top_Keepers_runouts
```

```
top_Keepers_stumps
```

If we rank the keepers in order of 1-10 on the above parameters. The top 3 keepers will be

1. MS Dhoni
2. Dinesh Karthik

3. Rishabh Pant

VISUALIZATION FOR ENHANCED ANALYSIS

```
#Visualization of Batters Data
#the plot shows each of the top batters strike rate
plt.figure(figsize=(20,10))
sns.barplot(x='Player Name', y= 'Strike_Rate', data=top_batters)

#the plot shows each of the top batters runs
plt.figure(figsize=(20,10))
sns.barplot(x='Player Name', y= 'Runs', data=top_batters)

#the plot shows each of the top batters average
plt.figure(figsize=(20,10))
sns.barplot(x='Player Name', y= 'Average', data=top_batters)

#the plot shows each of the top batters matches played
plt.figure(figsize=(20,10))
sns.barplot(x='Player Name', y= 'Matches_Played', data=top_batters)

#visualization for bowlers
#the plot shows each of the top bowlers bowling average
plt.figure(figsize=(20,10))
sns.barplot(x='Player Name', y= 'Bowling_average', data=top_bowlers)

#the plot shows each of the top bowlers Economy
plt.figure(figsize=(20,10))
sns.barplot(x='Player Name', y= 'Economy', data=top_bowlers)
```

```
#the plot shows each of the top bowlers strike rate  
plt.figure(figsize=(20,10))  
sns.barplot(x='Player Name', y= 'Bowling_ Strike_ Rate', data=top_bowlers)
```

```
#the plot shows each of the top bowlers wickets  
plt.figure(figsize=(20,10))  
sns.barplot(x='Player Name', y= 'Wickets', data=top_bowlers)
```

```
#the plot shows each of the top bowlers matches played  
plt.figure(figsize=(20,10))  
sns.barplot(x='Player Name', y= 'Matches Played', data=top_bowlers)
```

```
#Visualization for all rounders  
#this plots show the strike rate for top allrounders  
plt.figure(figsize=(20,10))  
sns.barplot(x='Player Name', y= 'Strike_Rate', data=top_allrounders)
```

```
#this plots show top allrounders average  
plt.figure(figsize=(20,10))  
sns.barplot(x='Player Name', y= 'Average', data=top_allrounders)
```

```
#this plots show the matches played by top allrounders  
plt.figure(figsize=(20,10))  
sns.barplot(x='Player Name', y= 'Matches Played', data=top_allrounders)
```

```
#this plots show top allrounders runs  
plt.figure(figsize=(20,10))
```

```
sns.barplot(x='Player Name', y= 'Runs', data=top_allrounders)

#this plots show top allrounders bowling average
plt.figure(figsize=(20,10))
sns.barplot(x='Player Name', y= 'Bowling_average', data=top_allrounders)

#this plots show top allrounders economy
plt.figure(figsize=(20,10))
sns.barplot(x='Player Name', y= 'Economy', data=top_allrounders)

#this plots show top allrounders bowling strike rate
plt.figure(figsize=(20,10))
sns.barplot(x='Player Name', y= 'Bowling_Strike_Rate', data=top_allrounders)

#this plots show top allrounders matches played
plt.figure(figsize=(20,10))
sns.barplot(x='Player Name', y= 'Matches_Played', data=top_allrounders)

#this plots show top allrounders economy
plt.figure(figsize=(20,10))
sns.barplot(x='Player Name', y= 'Economy', data=top_allrounders)

#this plots show top allrounders wickets
plt.figure(figsize=(20,10))
sns.barplot(x='Player Name', y= 'Wickets', data=top_allrounders)

#Visualization for keepers
#this plots show keepers average
```

```

plt.figure(figsize=(20,10))
sns.barplot(x='Player Name', y= 'Average', data=top_keepers)

#this plots show keepers average
plt.figure(figsize=(20,10))
sns.barplot(x='Player Name', y= 'Average', data=top_keepers)

#this plots show keepers runs
plt.figure(figsize=(20,10))
sns.barplot(x='Player Name', y= 'Runs', data=top_keepers)

#this plots show keepers strike rate
plt.figure(figsize=(20,10))
sns.barplot(x='Player Name', y= 'Strike_Rate', data=top_keepers)

#this plots show keepers catches
plt.figure(figsize=(20,10))
sns.barplot(x='Player Name', y= 'Catches', data=top_keepers)

#this plots show keepers runouts
plt.figure(figsize=(20,10))
sns.barplot(x='Player Name', y= 'Run_outs', data=top_keepers)

#this plots show stump done by keepers
plt.figure(figsize=(20,10))
sns.barplot(x='Player Name', y= 'Stumps', data=top_keepers)

#this plots show keepers matches played
plt.figure(figsize=(20,10))
sns.barplot(x='Player Name', y= 'Matches_Played', data=top_keepers)

```

Forming Our Best 11 for the Campaign based on the above analysis

1. We will consider the number of players from each category that the 120 world cup winning and the last year's IPL winning team played in their Final matches

2. The Australia squad consisted of - 3 Batters, 3 Allrounders, 4 Bowlers with 1 spin option and 1 wicket keeper.

3. The Chennai Squad Consisted of - 4 Batters, 3 Allrounders, 3 Bowlers and 1 Wicket Keeper.

4. For our final analysis we will consider the ratio of players in the best 11 as follows:

3 Batters

3 Allrounders

4 Bowlers with 2 Spin Options

1 Wicket Keeper

#Batters for the Final 11 - KL Rahul, Virat Kohli, David Warner

#here, we are storing the values of each player in a separate dataframe to use for displaying using the barplot

```
top_batters.reset_index(drop=True)
matches_values = [top_batters.iloc[6]['Matches_Played'],
top_batters.iloc[2]['Matches_Played'], top_batters.iloc[5]['Matches_Played']]
runs_values = [top_batters.iloc[6]['Runs'],
top_batters.iloc[2]['Runs'],top_batters.iloc[5]['Runs']]
average_values = [top_batters.iloc[6]['Average'],
top_batters.iloc[2]['Average'],top_batters.iloc[5]['Average']]
Strike_rate_values = [top_batters.iloc[6]['Strike_Rate'],top_batters.iloc[2]['Strike_Rate'],
top_batters.iloc[5]['Strike_Rate']]
Labels = ['KL Rahul', 'David Warner', 'Virat Kohli']
```

```
fig, axes = plt.subplots(2,2, figsize=(10,10))
axes[0][0].set_title("Matches Played")
axes[0][1].set_title("Runs in the IPL Career")
axes[1][0].set_title("Strike Rate")
axes[1][1].set_title("Average")
sns.barplot(x=Labels, y=matches_values, ax=axes[0][0])
sns.barplot(x=Labels, y=runs_values, ax=axes[0][1])
sns.barplot(x=Labels, y=Strike_rate_values, ax=axes[1][0])
sns.barplot(x=Labels, y=average_values, ax=axes[1][1])
```

```

top_allrounders.reset_index(drop=True)
matches_values = [top_allrounders.iloc[5]['Matches_Played'], top_allrounders.iloc[9]['Matches_Played'], top_allrounders.iloc[6]['Matches_Played']]
runs_values = [top_allrounders.iloc[5]['Runs'], top_allrounders.iloc[9]['Runs'], top_allrounders.iloc[6]['Runs']]
average_values = [top_allrounders.iloc[5]['Average'], top_allrounders.iloc[9]['Average'], top_allrounders.iloc[6]['Average']]
strike_rate_values = [top_allrounders.iloc[5]['Strike_Rate'], top_allrounders.iloc[9]['Strike_Rate'], top_allrounders.iloc[6]['Strike_Rate']]
bowling_strike_rate_values = [top_allrounders.iloc[5]['Bowling_Strike_Rate'], top_allrounders.iloc[9]['Bowling_Strike_Rate'], top_allrounders.iloc[6]['Bowling_Strike_Rate']]
bowling_average_values = [top_allrounders.iloc[5]['Bowling_average'], top_allrounders.iloc[9]['Bowling_average'], top_allrounders.iloc[6]['Bowling_average']]
wickets_values = [top_allrounders.iloc[5]['Wickets'], top_allrounders.iloc[9]['Wickets'], top_allrounders.iloc[6]['Wickets']]
economy_values = [top_allrounders.iloc[5]['Economy'], top_allrounders.iloc[9]['Economy'], top_allrounders.iloc[6]['Economy']]
Labels = ['Andre Russell', 'Sunil Narine', 'Hardik Pandya']
fig, axes = plt.subplots(4, 2, figsize=(20, 20))
axes[0][0].set_title("Matches")
axes[0][1].set_title("Runs")
axes[1][0].set_title("Average")
axes[1][1].set_title("Strike Rate")
axes[2][0].set_title("Bowling Strike Rate")
axes[2][1].set_title("Bowling Average")
axes[3][0].set_title("wickets")
axes[3][1].set_title("Economy")
sns.barplot(x=Labels, y=matches_values, ax=axes[0][0])
sns.barplot(x=Labels, y=runs_values, ax=axes[0][1])
sns.barplot(x=Labels, y=average_values, ax=axes[1][0])

```

```

sns.barplot(x=Labels, y=strike_rate_values, ax=axes[1][1])
sns.barplot(x=Labels, y=bowling_strike_rate_values, ax=axes[2][0])
sns.barplot(x=Labels, y=bowling_average_values, ax=axes[2][1])
sns.barplot(x=Labels, y=wickets_values, ax=axes[3][0])
sns.barplot(x=Labels, y=economy_values, ax=axes[3][1])

top_bowlers.reset_index(drop=True)

matches_values = [top_bowlers.iloc[10]['Matches_Played'],
                  top_bowlers.iloc[7]['Matches_Played'],
                  top_bowlers.iloc[1]['Matches_Played']]
wickets_values = [top_bowlers.iloc[10]['Wickets'], top_bowlers.iloc[0]['Wickets'],
                  top_bowlers.iloc[7]['Wickets'], top_bowlers.iloc[1]['Wickets']]
bowling_average_values = [top_bowlers.iloc[10]['Bowling_average'],
                           top_bowlers.iloc[7]['Bowling_average'],
                           top_bowlers.iloc[1]['Bowling_average']]
bowling_strike_rate_values = [top_bowlers.iloc[10]['Bowling_Strike_Rate'],
                               top_bowlers.iloc[7]['Bowling_Strike_Rate'],
                               top_bowlers.iloc[1]['Bowling_Strike_Rate']]
economy_values = [top_bowlers.iloc[10]['Economy'], top_bowlers.iloc[0]['Economy'],
                  top_bowlers.iloc[7]['Economy'], top_bowlers.iloc[1]['Economy']]

```

Labels = ['Jasprit Bumrah', 'Keiso Rabada', 'Nathan Coulter-Nile', 'Yuzi Chahal']

```

fig, axes = plt.subplots(3, 2, figsize=(15, 15))
axes[0][0].set_title("Matches Played")
axes[0][1].set_title("Wickets")
axes[1][0].set_title("Bowling Average")
axes[1][1].set_title("Bowling Strike Rate")
axes[2][0].set_title("Economy")

sns.barplot(x=Labels, y=matches_values, ax=axes[0][0])
sns.barplot(x=Labels, y=wickets_values, ax=axes[0][1])
sns.barplot(x=Labels, y=bowling_average_values, ax=axes[1][0])
sns.barplot(x=Labels, y=bowling_strike_rate_values, ax=axes[1][1])

```

```

sns.barplot(x=Labels, y=economy_values, ax=axes[2][0])

matches_values = [top_keepers.iloc[8]['Matches_Played'], top_keepers.iloc[8]['Runs']]
average_values = [top_keepers.iloc[8]['Average'], top_keepers.iloc[8]['Strike_Rate']]
keeping_values = [top_keepers.iloc[8]['Catches'], top_keepers.iloc[8]['Stumps'],
top_keepers.iloc[8]['Run_outs']]
label1 = ['Matches', 'Runs']
label2 = ['Average', 'Strike Rate']
label3 = ['Catches', 'Stumps', 'Run_outs']

fig, axes = plt.subplots(1, 3, figsize=(20, 10))
axes[0].set_title("Matches And Runs")
axes[1].set_title("Average and Strike Rate")
axes[2].set_title("Keeping Stats")

sns.barplot(x=label1, y=matches_values, ax=axes[0])
sns.barplot(x=label2, y=average_values, ax=axes[1])
sns.barplot(x=label3, y=keeping_values, ax=axes[2])

batter1 = top_batters.loc[(top_batters["Player Name"] == 'KL Rahul ')]
batter2 = top_batters.loc[(top_batters["Player Name"] == 'David Warner ')]
batter3 = top_batters.loc[(top_batters["Player Name"] == 'Virat Kohli')]

bowler1 = top_bowlers.loc[(top_bowlers["Player Name"] == 'Yuzvendra Chahal ')]
bowler2 = top_bowlers.loc[(top_bowlers["Player Name"] == 'Jasprit Bumrah')]
bowler3 = top_bowlers.loc[(top_bowlers["Player Name"] == 'Nathan Coulter-Nile')]
bowler4 = top_bowlers.loc[(top_bowlers["Player Name"] == 'Kagiso Rabada ')]

allrounder1 = top_allrounders.loc[(top_allrounders["Player Name"] == 'Andre Russell')]
allrounder2 = top_allrounders.loc[(top_allrounders["Player Name"] == 'Sunil Narine ')]
allrounder3 = top_allrounders.loc[(top_allrounders["Player Name"] == 'Hardik Pandya')]

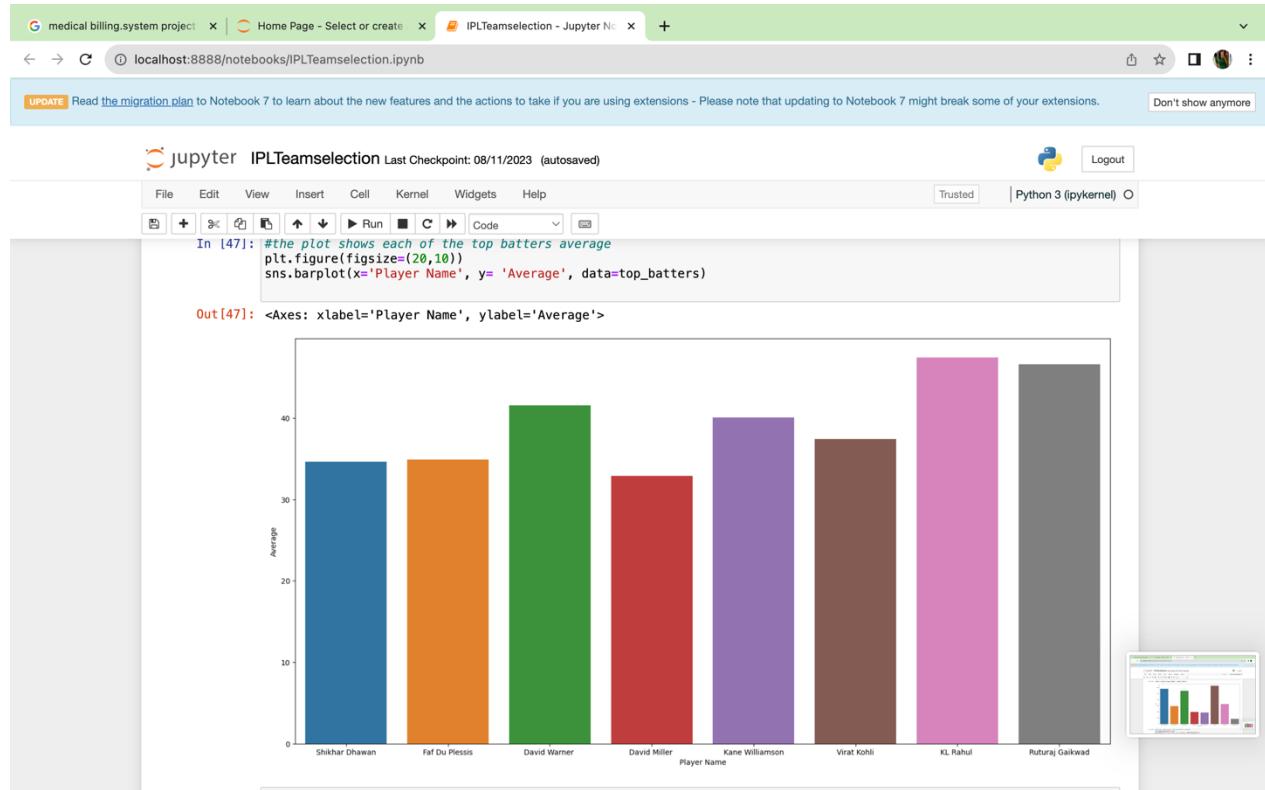
keeper = top_keepers.loc[(top_keepers["Player Name"] == 'MS Dhoni')]

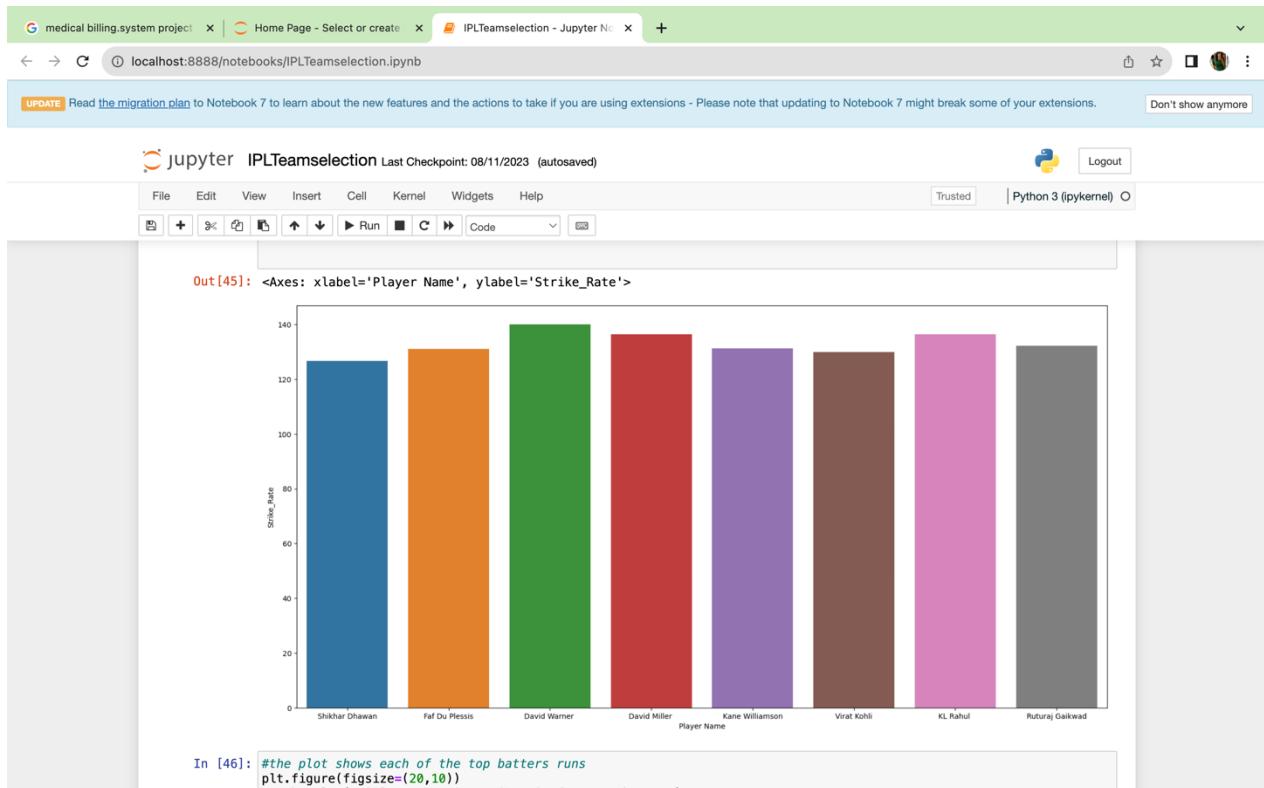
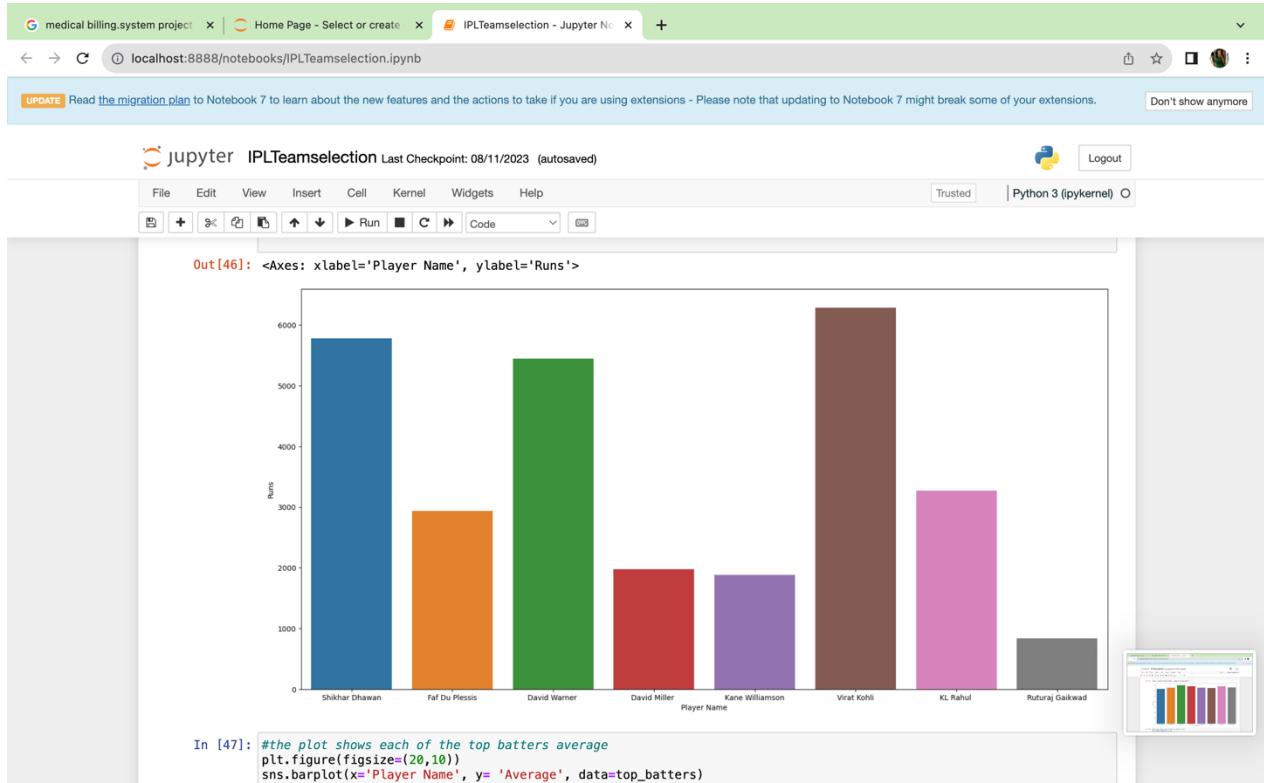
```

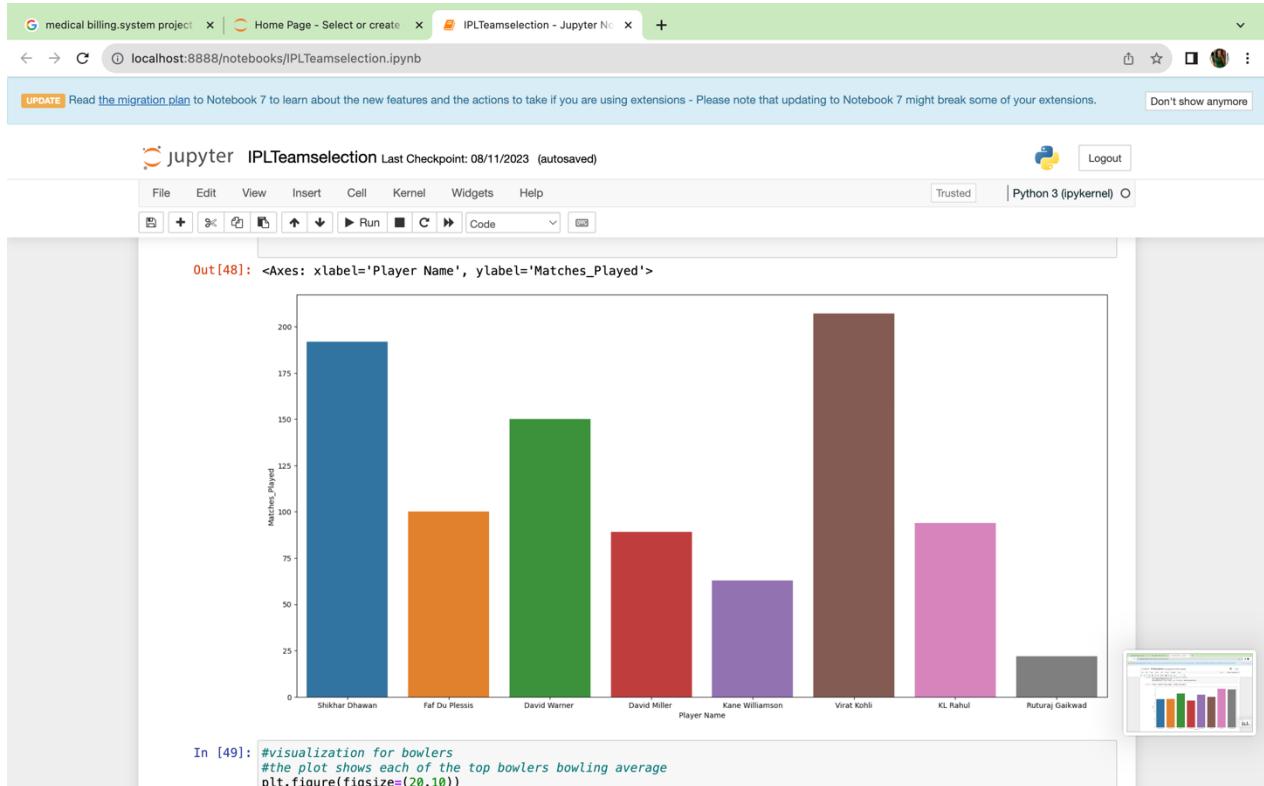
```
final = [batter1, batter2, batter3, allrounder1, allrounder2, allrounder3, keeper, bowler1,  
bowler2, bowler3, bowler4]  
final_team = pd.concat(final)  
final_team = final_team.drop(labels=['Matches_Played', 'Runs', 'Average', 'Strike_Rate',  
'Wickets',  
'Bowling_average', 'Economy', 'Bowling_Strike_Rate',  
'Catches', 'Run_outs', 'Stumps'], axis=1)  
  
final_team.reset_index(drop=True)
```

6. RESULTS AND DISCUSSION

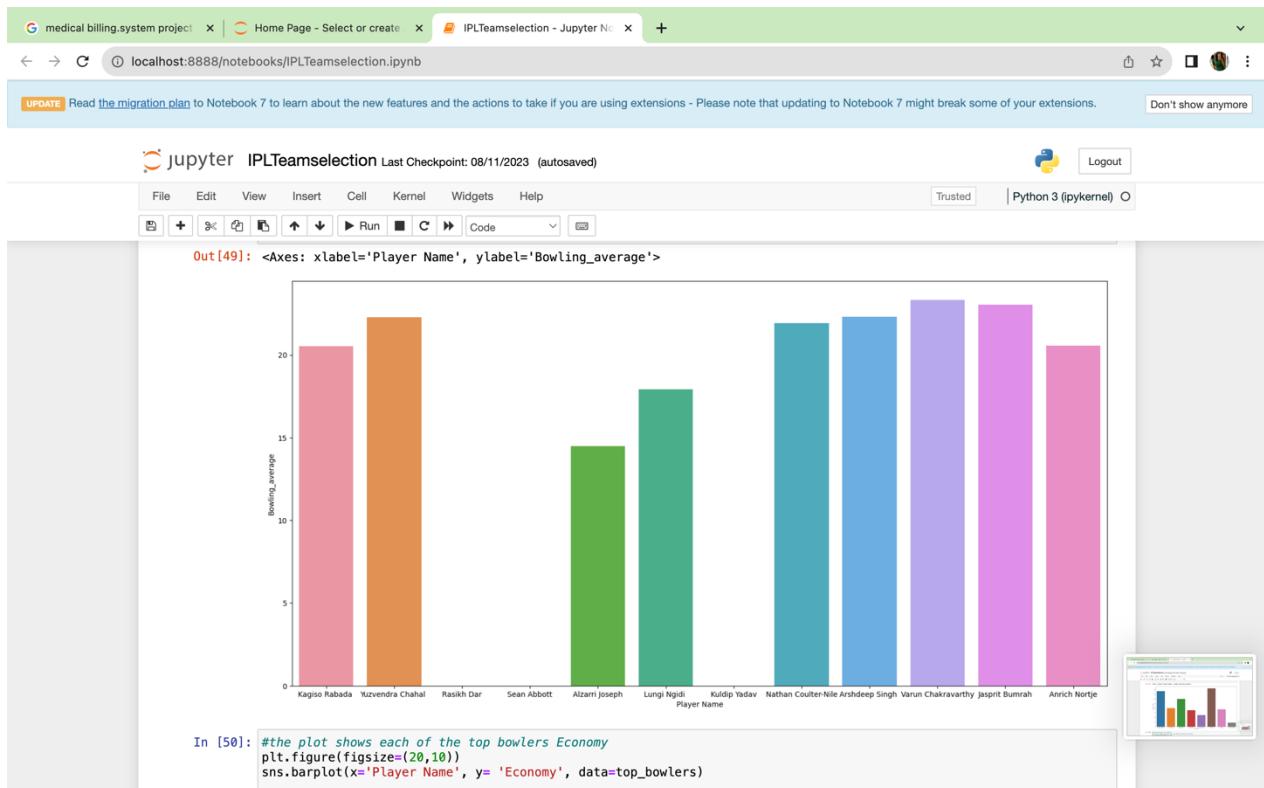
A. Batters :

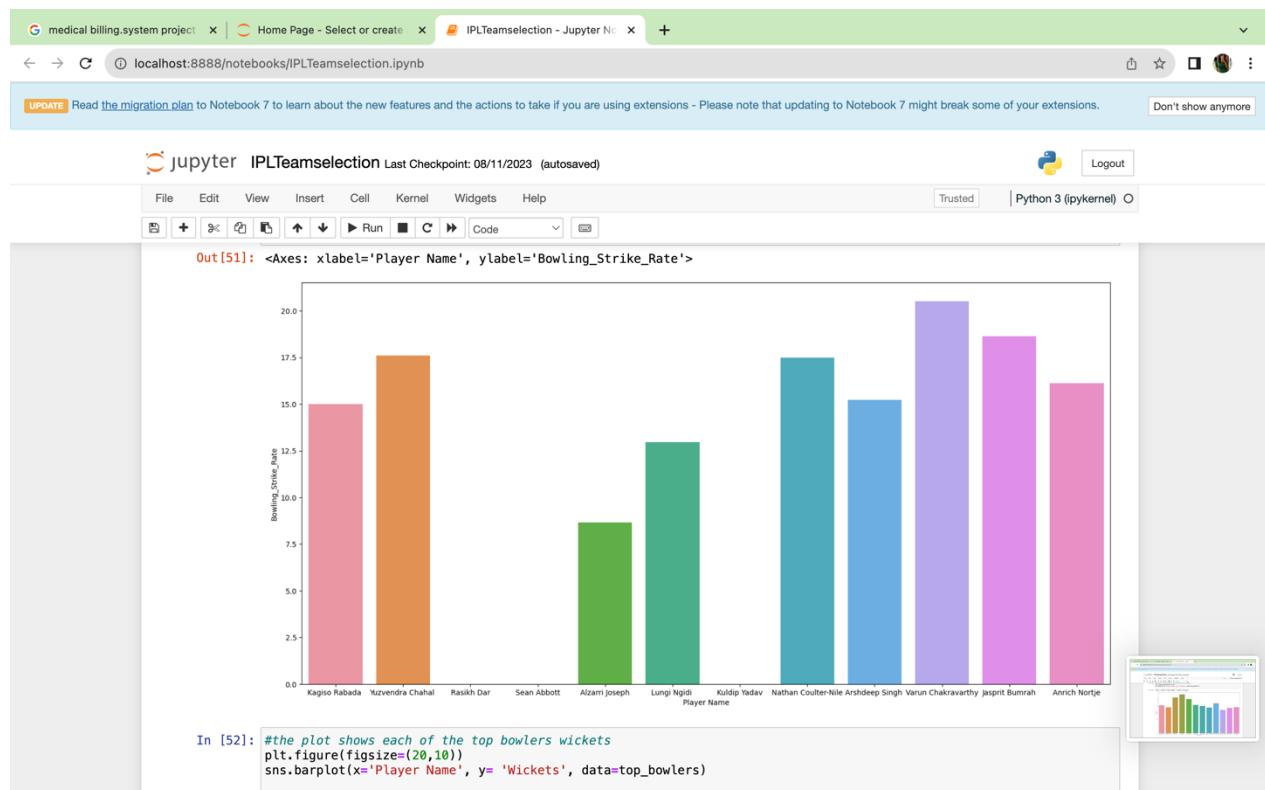
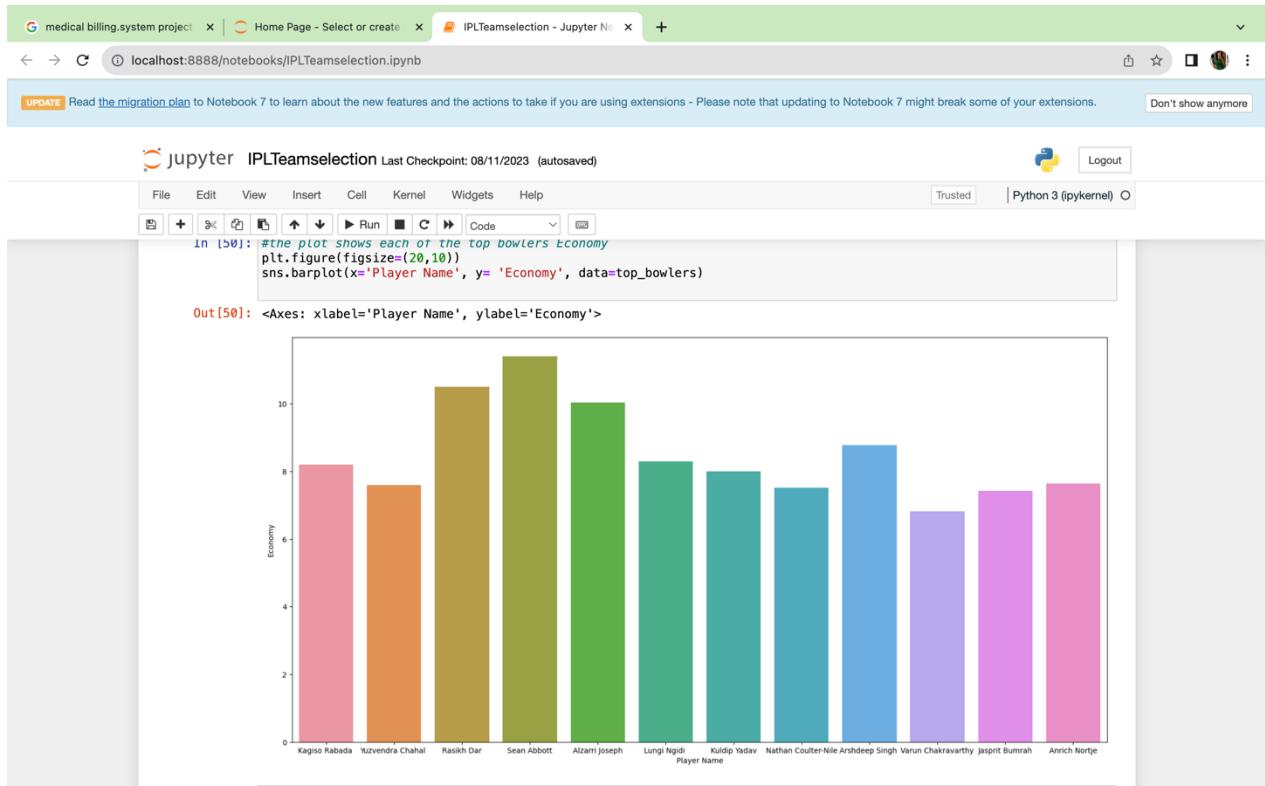


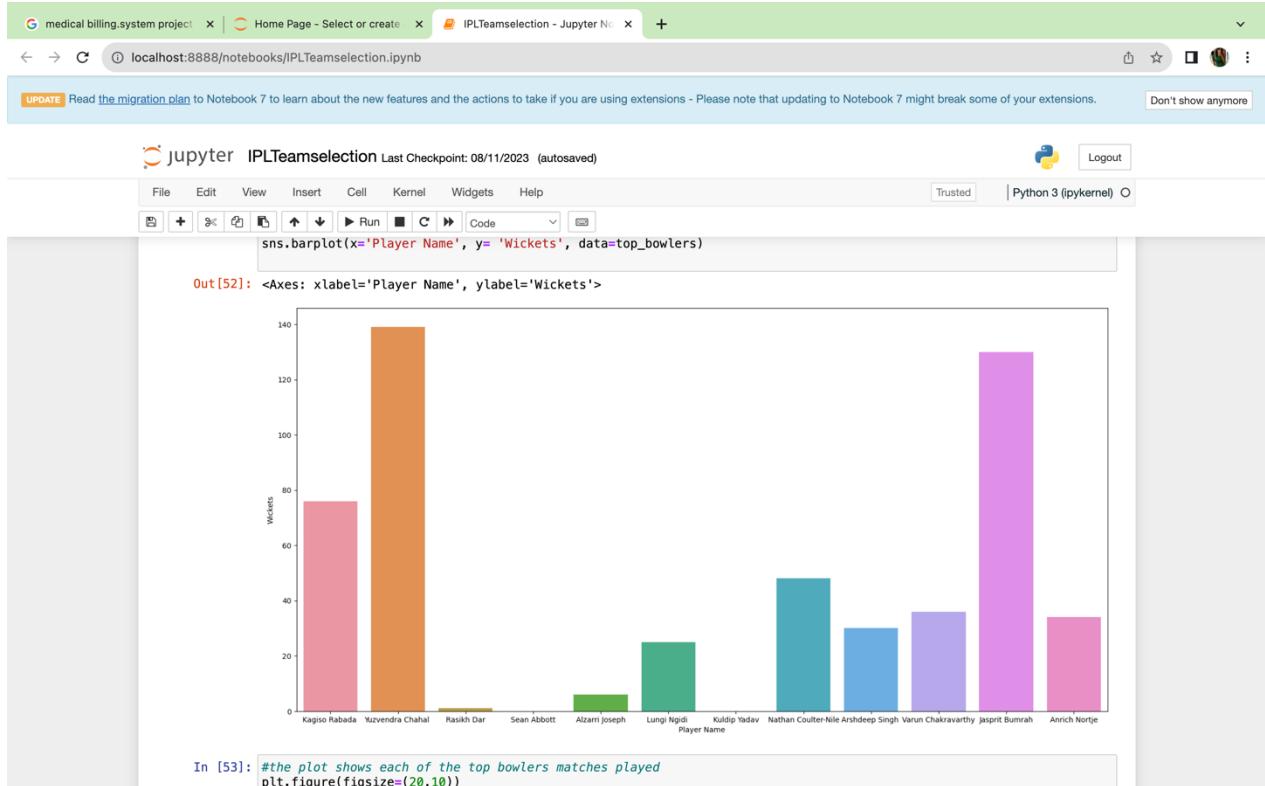




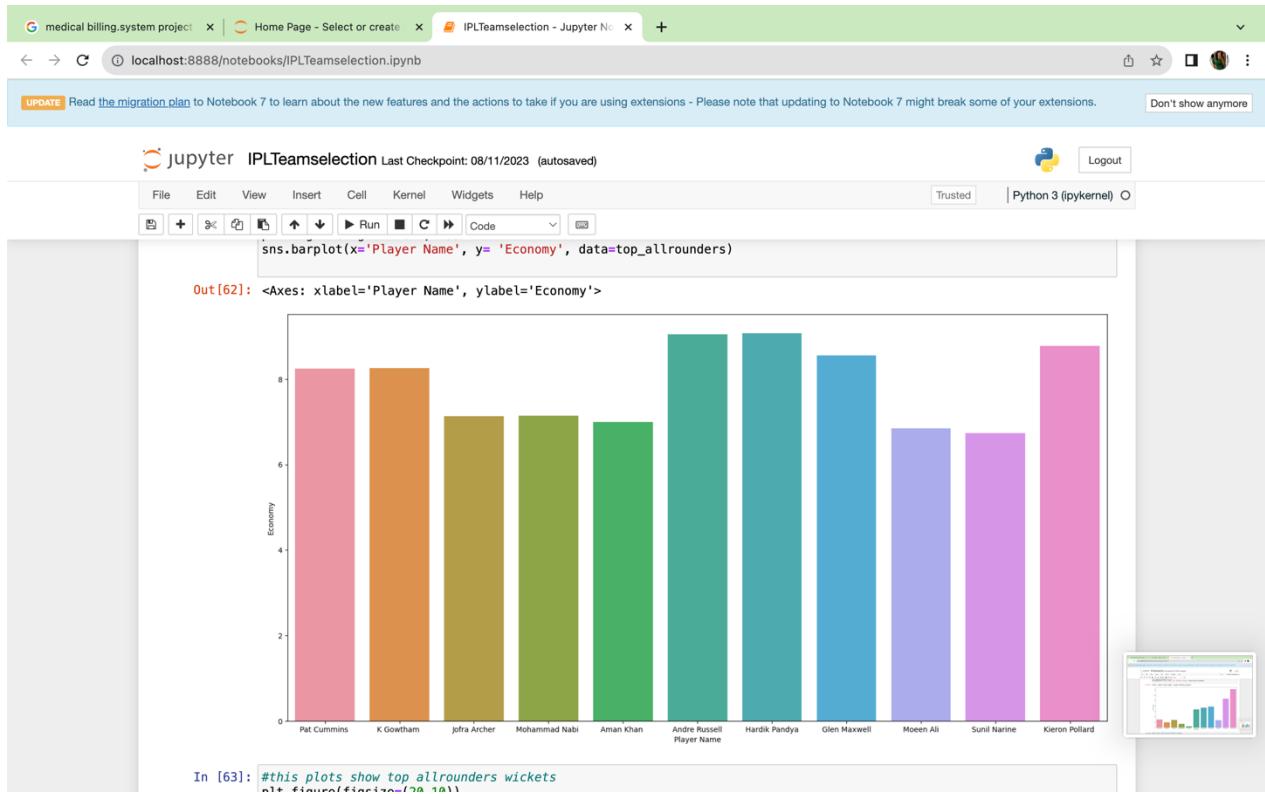
B. Bowlers Economy:

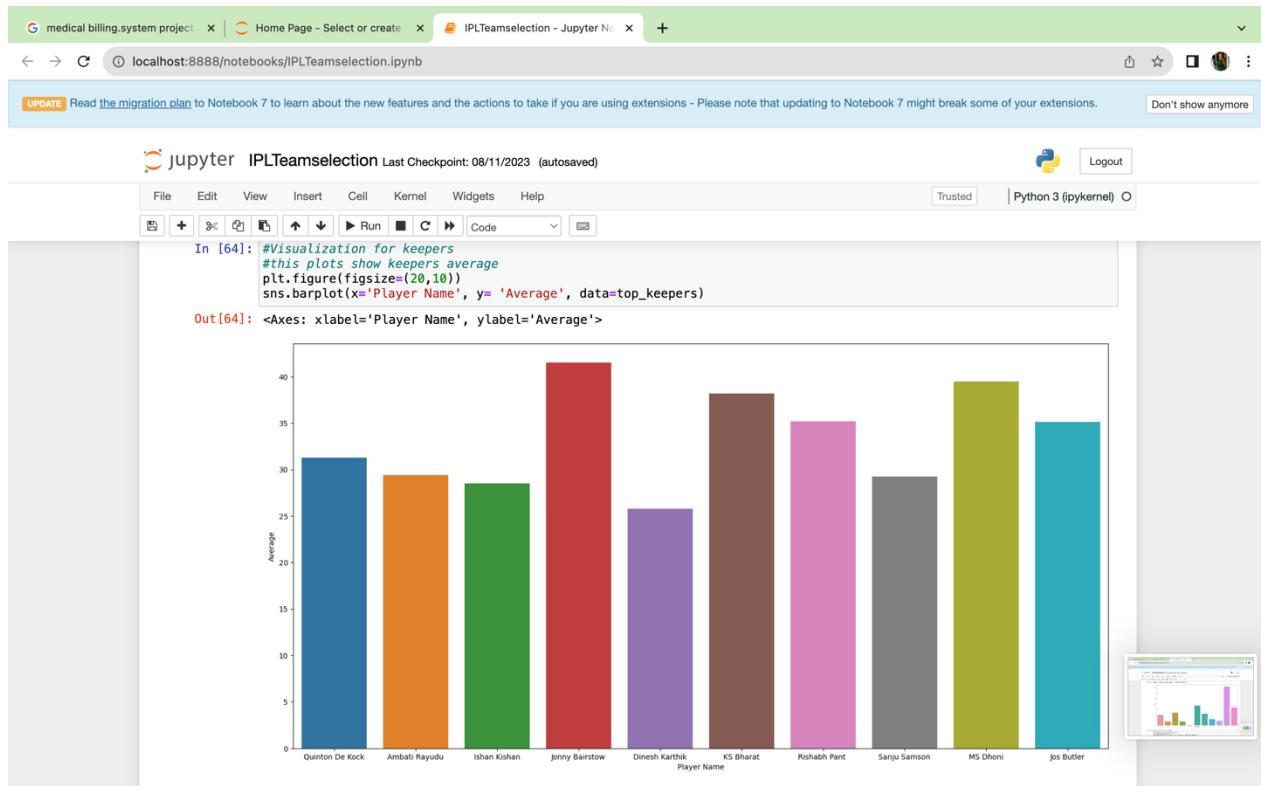
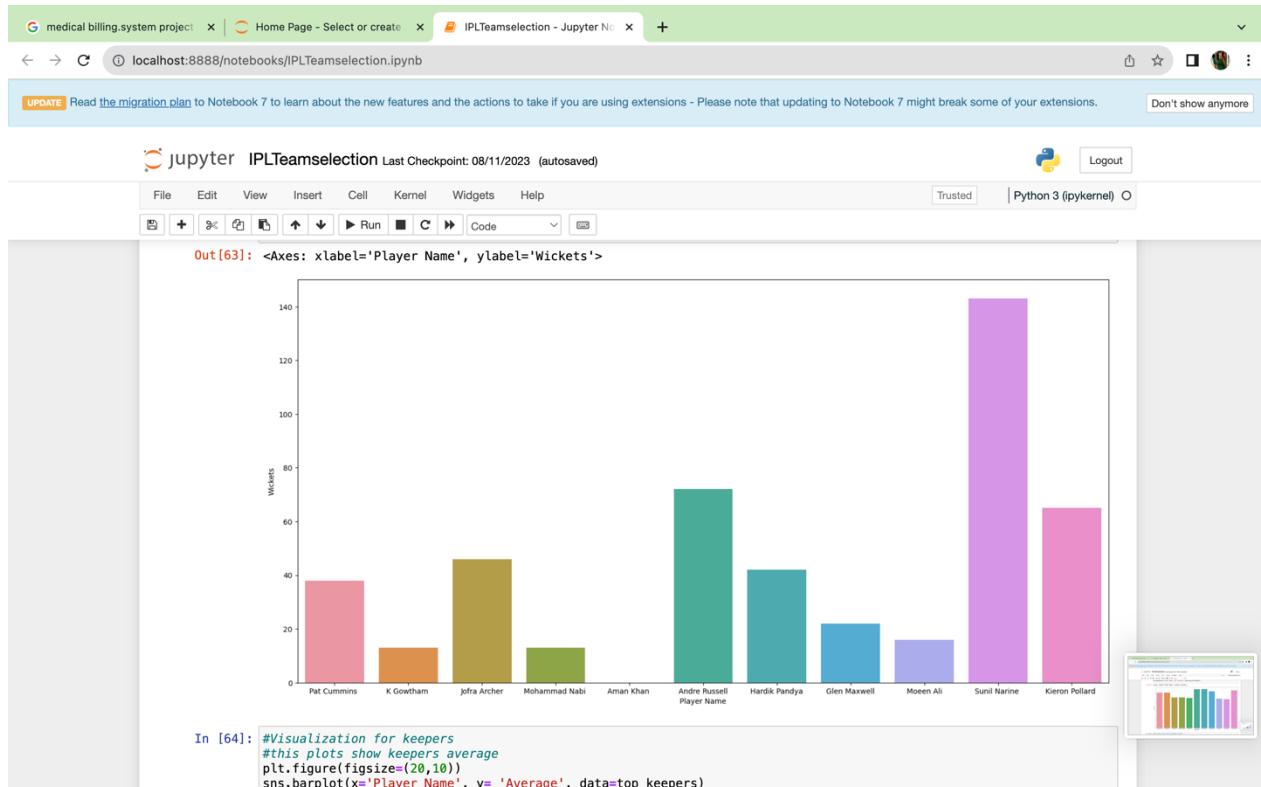


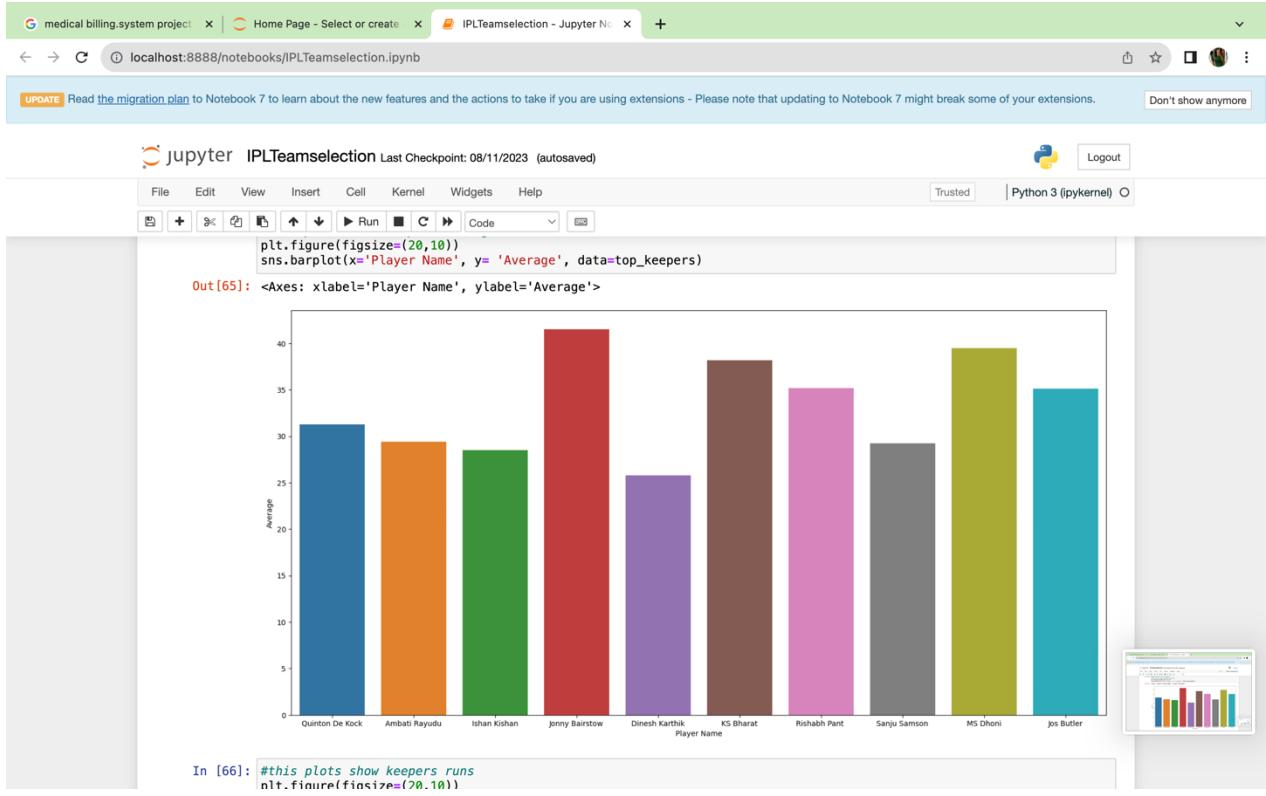




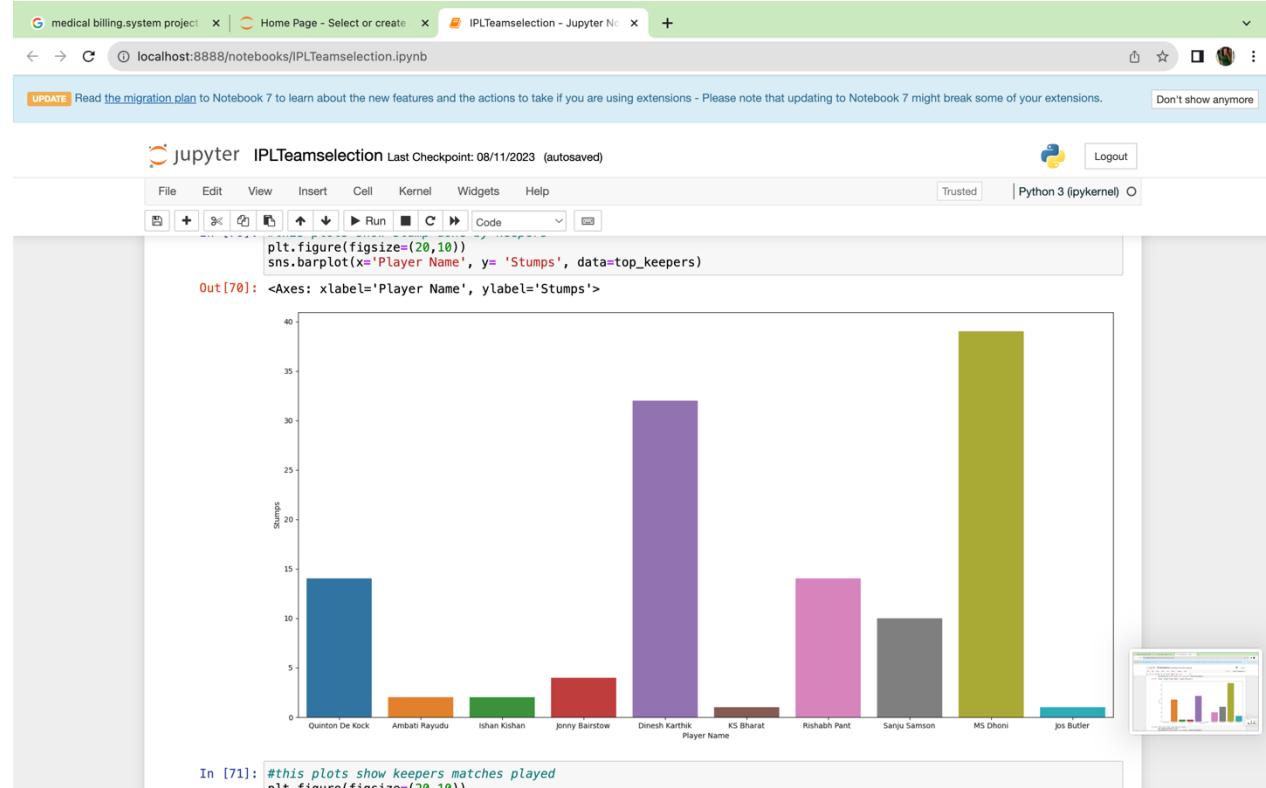
C. All Rounders

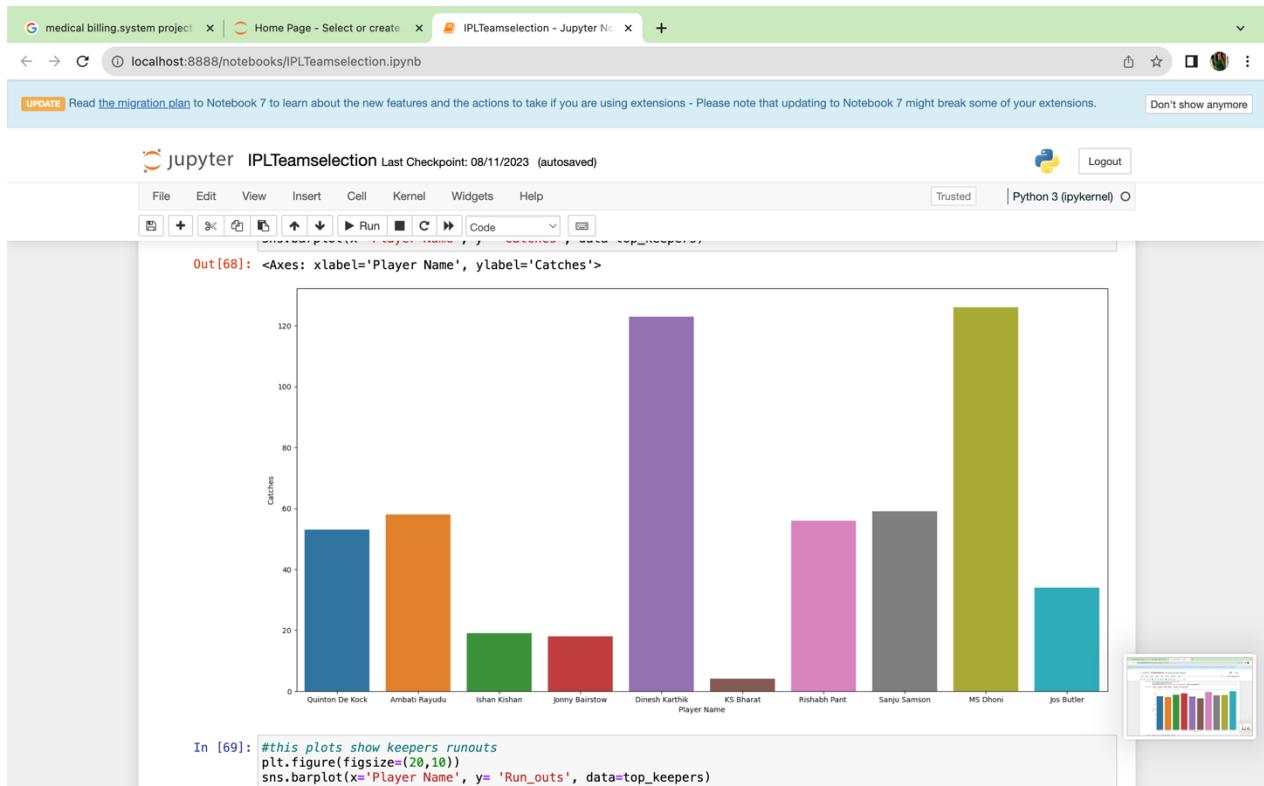
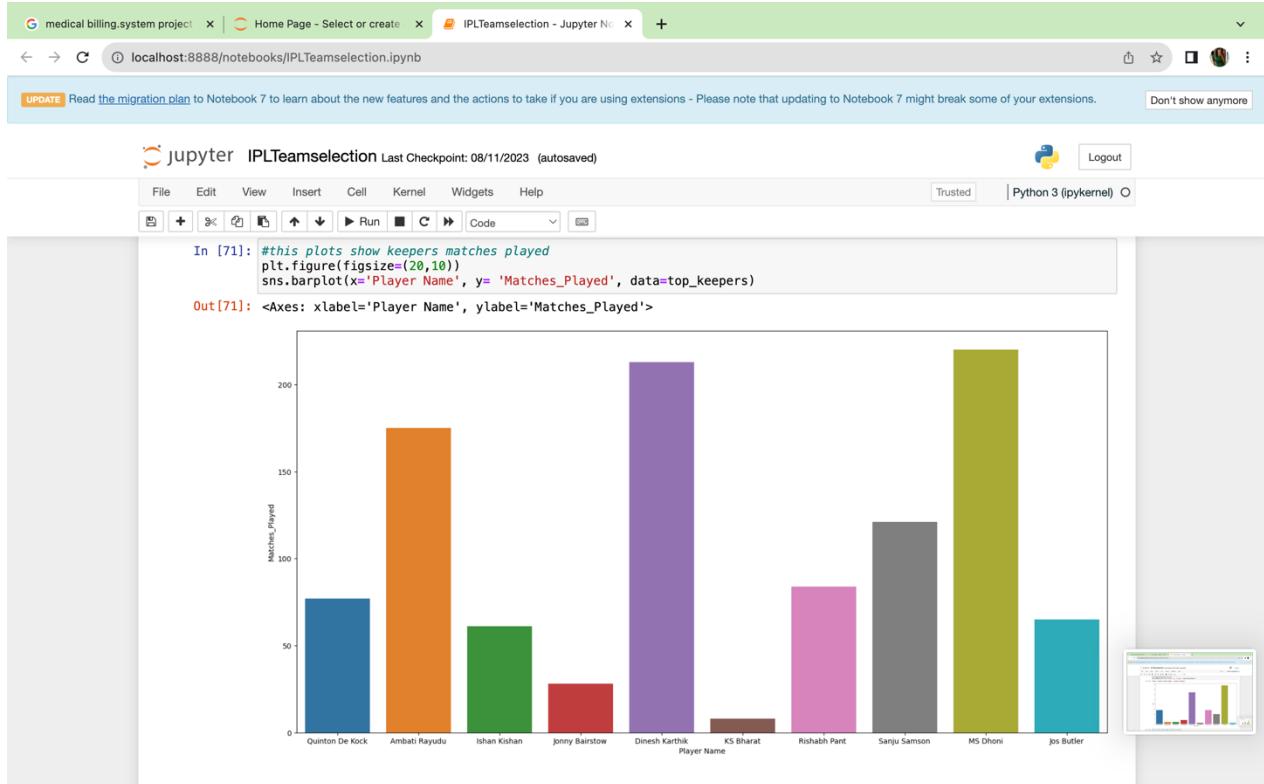


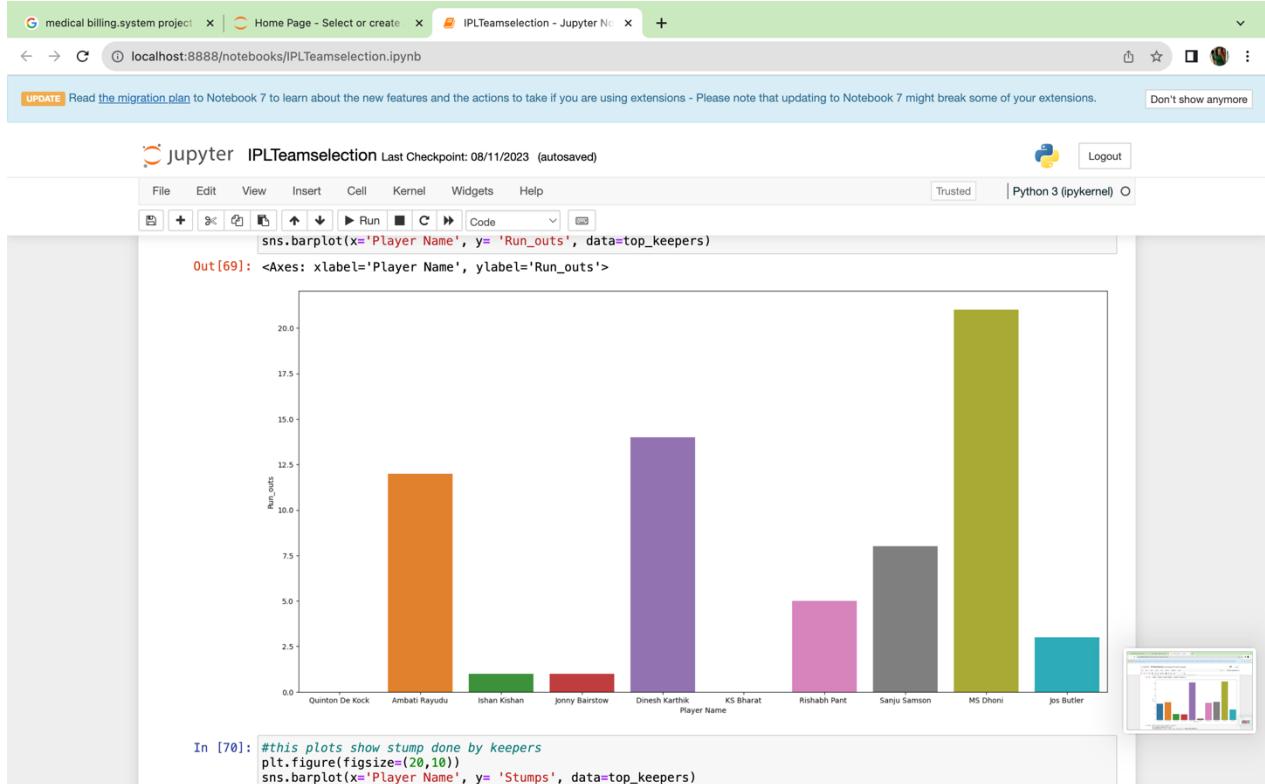




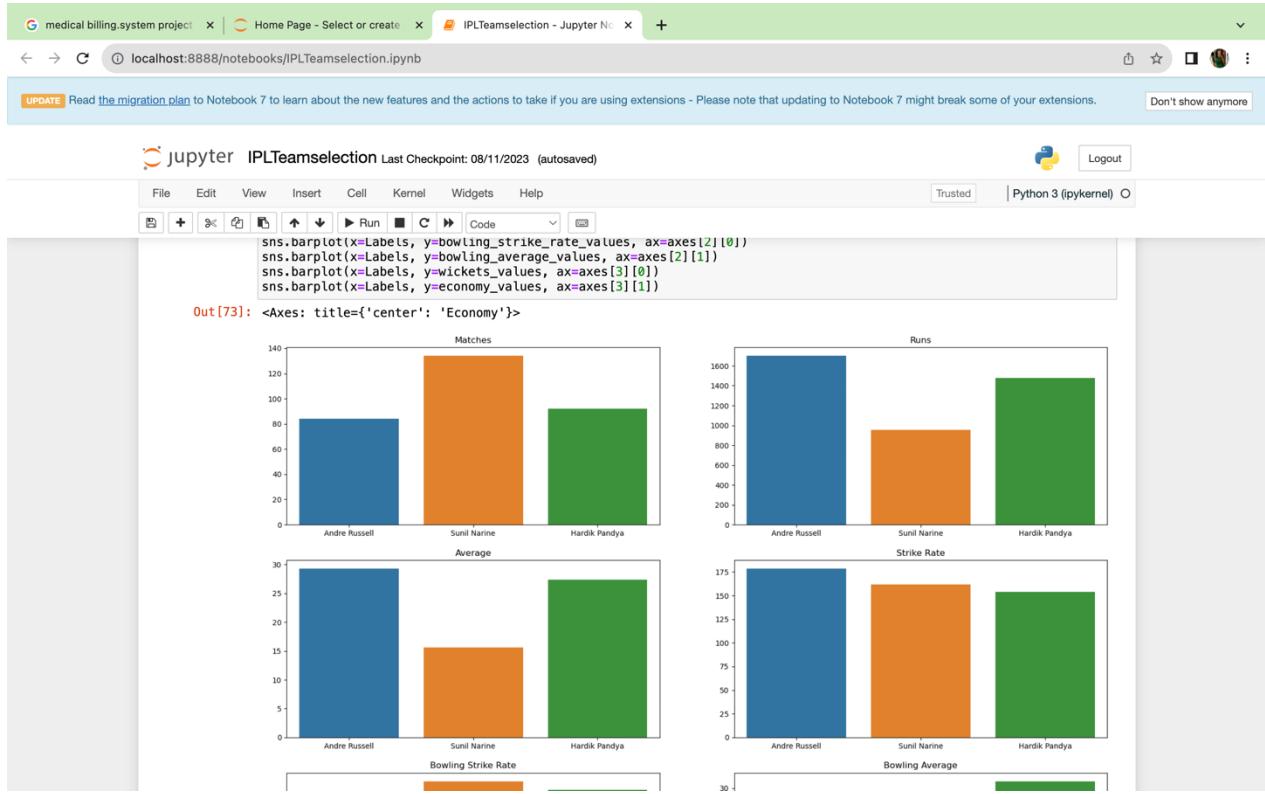
D. Keepers

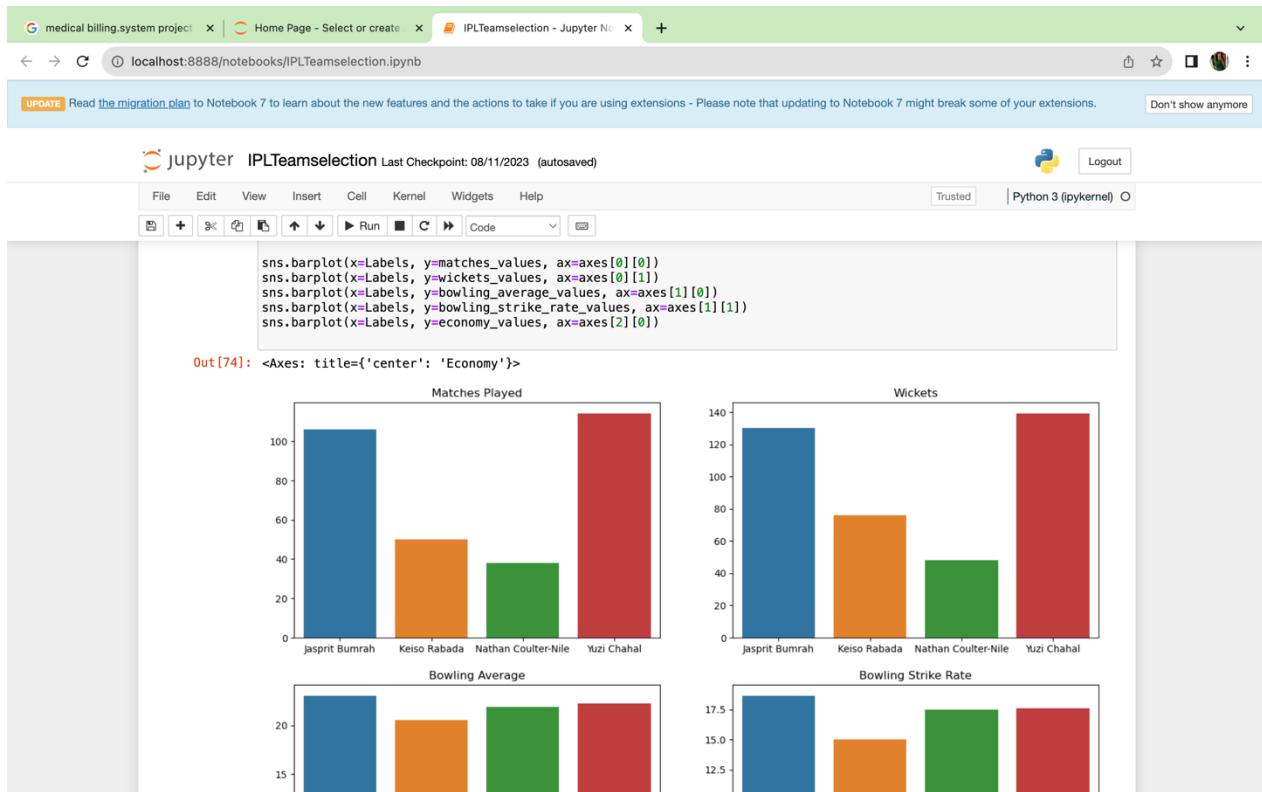
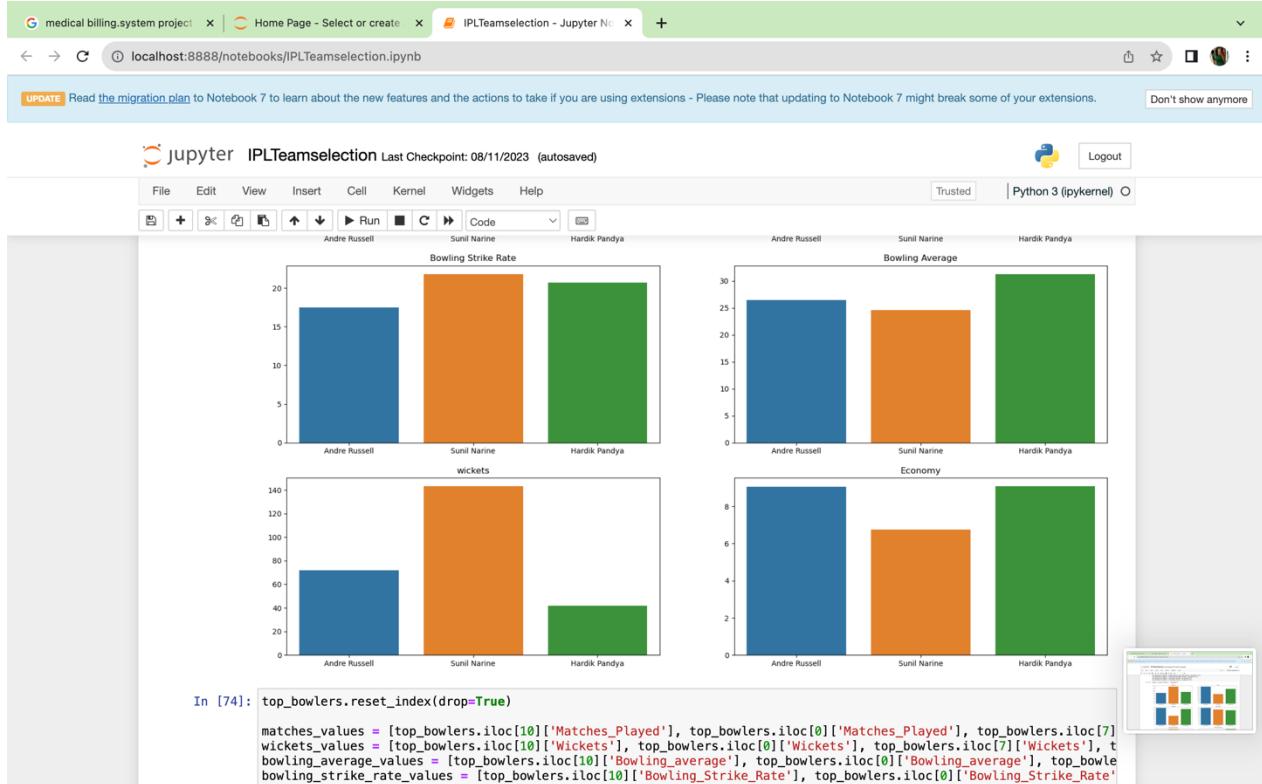


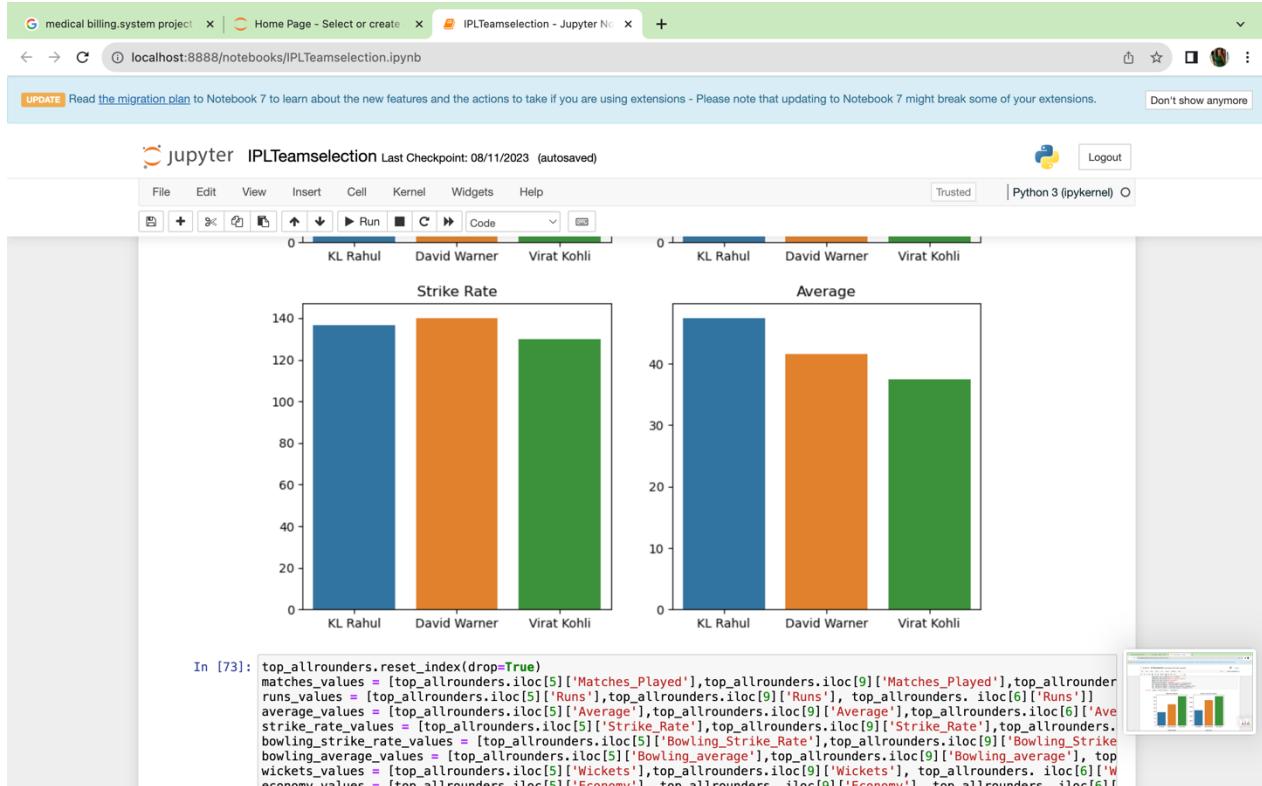




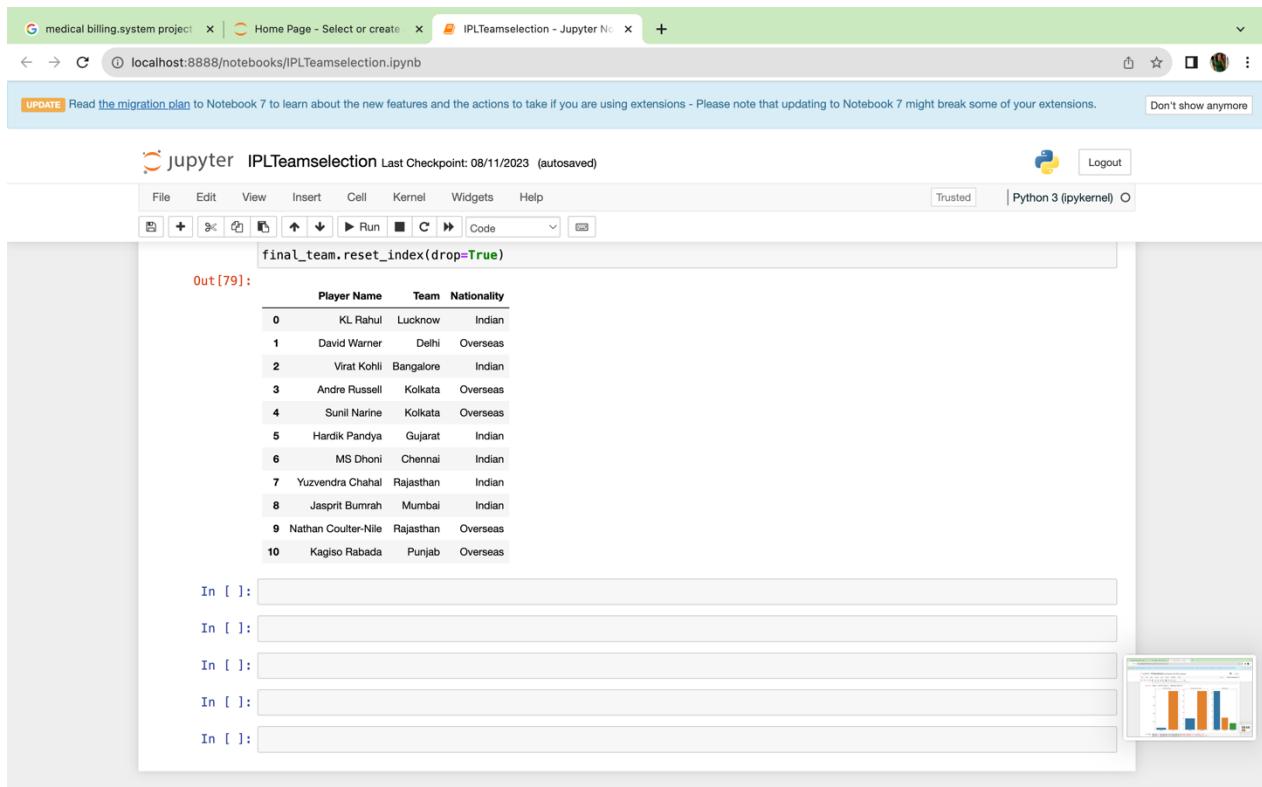
E. Comparison Between Players :







F. Team formation:



7.CONCLUSION:

In conclusion, our IPL auction data science analysis unearthed key insights shaping team dynamics. Player valuations proved multifaceted, influenced by team strategies, player roles, and economic factors beyond on-field performance. Successful teams strategically balanced experienced players with emerging talents. The correlation between team success and this judicious mix highlighted the importance of nuanced team-building. Moreover, economic influences, such as marketability and brand value, played a crucial role in determining player valuations. The temporal evolution of auction dynamics revealed shifts in strategies tied to team management changes, rule modifications, and global cricketing events. Armed with these findings, stakeholders can make informed decisions in future auctions, considering holistic factors for enhanced team success. Challenges include the inherent unpredictability of sports, prompting future research into real-time sentiment analysis and injury prediction models for more accurate auction predictions. Our project showcases the potential of data science to deepen our understanding of cricketing ecosystems and optimize team-building strategies in the ever-evolving IPL landscape. The visualization of top players through bar plots adds a visual dimension to the analysis, allowing for a quick and effective comparison of key statistics. The strategic approach to forming the best 11, considering the composition of successful teams, adds depth to the insights gained from individual player performances.

This comprehensive examination not only identifies top players like KL Rahul, Virat Kohli, David Warner, Andre Russell, and others but also forms a well-balanced team for a hypothetical campaign. The inclusion of both statistical analysis and visual representation contributes to a holistic understanding of player dynamics in the context of team composition and success.

8. REFERENCE:

- [1] C. CricketAnalyst and B. PlayerValuation, “Optimizing IPL Auction Strategies through Advanced Player Valuation Models,” J Sports Analytics, 2019.
- [2] I. TeamPerformance, R. AuctionInsights, N. DataAnalytics et al., “Integrating AI and Data Analytics for In-Depth IPL Auction Analysis,” Int J Sports Informatics, 2021.
- [3] Z. BidOptimizR et al., “Optimizing IPL Auction Workflow using Machine Learning Techniques,” Sports Autom Rev, 2022.
- [4] D. ComplianceGuard, R. BidAuditTrail, “Ensuring Transparency and Compliance in IPL Auctions,” J Sports Policy Manag, 2020