Project Report

IPL Auction 2025

Abstract of the Project

The **IPL Auction 2025** project aims to extract, analyze, and visualize player auction data to uncover key trends and insights. The project begins with **web scraping** using requests and BeautifulSoup to collect real-time data from the official IPL website. The extracted data, including player names, base prices, final sold prices, teams, and categories, is then structured into a **CSV format** using pandas.

Following data extraction, **Python-based analysis** is performed to identify top purchases, team spending patterns, player category distributions, and price trends. Key libraries such as pandas, matplotlib, and plotly are used for in-depth statistical analysis and data visualization.

To enhance data interpretation, a **Google Looker Studio dashboard** is developed, featuring interactive charts, graphs, and filters that allow users to explore auction trends dynamically. The dashboard provides insights into the most expensive players, total spending, and category distributions.

This project successfully integrates **data collection, analysis, and visualization**, offering a data-driven perspective on IPL auction strategies. The findings help in understanding player demand, franchise investment patterns, and overall auction dynamics.

Index

S.NO.	CHAPTER NAME	REMARK
1	INTRODUCTION	
2	OBJECTIVE OF THE PROJECT	
3	REQUIREMENTS	
4	METHODOLOGY	
5	WORKFLOW	
6	DASHBOARD	
7	FUTURE SCOPE	
8	CONCLUSION	
9	LEARNING OUTCOMES	

Introduction

1. About the project idea:

The **IPL Auction 2025** project focuses on extracting and analyzing player auction data to gain insights into team spending, player demand, and auction trends. The project involves **web scraping, data cleaning, analysis, and visualization** to create a structured and insightful view of the auction results.

Key objectives of the project include:

- Collecting **real-time auction data** from the official IPL website.
- Analyzing **player prices, team spending, and category distributions** using Python.
- Creating an **interactive dashboard** in Google Looker Studio for better data visualization.

This project helps in understanding auction dynamics, including how teams invest in players, the price variations between different player categories, and which players attract the highest bids.

2. About tools in use:

The project utilizes a combination of **data scraping, analysis, and visualization tools**, including:

1. **Web Scraping Tools:**
    - requests: Fetches HTML content from the IPL website.
    - BeautifulSoup: Parses and extracts auction data from the website.
2. **Data Processing & Analysis:**
    - pandas: Cleans, structures, and converts scraped data into CSV format.
    - numpy: Performs numerical computations for statistical insights.
    - matplotlib & plotly: Used for **data visualization** to explore trends and patterns.
3. **Dashboard & Visualization:**
    - **Google Looker Studio**: Creates an **interactive dashboard** with bar charts, pie charts, and filters to explore IPL auction data dynamically.

Objective of the Project

The IPL Auction 2025 project aims to collect, analyze, and visualize player auction data to gain insights into team spending, player demand, and bidding trends. The key objectives of this project are:

1. **Web Scraping:**
    - Extract real-time IPL auction data from the official IPL website using requests and BeautifulSoup.
    - Collect key details such as player name, base price, sold price, team, and player category.
2. **Data Processing & Storage:**
    - Convert the scraped data into a structured CSV format using pandas.
    - Perform data cleaning and preprocessing to remove inconsistencies and missing values.
3. **Data Analysis using Python:**
    - Identify top purchases and highest-paid players in the auction.
    - Analyze team-wise spending and investment patterns.
    - Study price variations between base price and final sold price.
    - Categorize players into batsmen, bowlers, all-rounders, and wicketkeepers and analyze their demand.
4. **Dashboard Development in Google Looker Studio:**
    - Create an interactive dashboard with visual insights.
    - Display bar charts, pie charts, and line graphs to represent spending trends.
    - Implement filters for team-wise and player-specific analysis.

Requirements

The **IPL Auction 2025** project requires various tools, technologies, and datasets for successful execution. The requirements are categorized as follows:

1. Software Requirements

- **Python** (For web scraping and data analysis)
- **Google Looker Studio** (For data visualization and dashboard creation)
- **Jupyter Notebook / VS Code / PyCharm** (For Python development)

2. Libraries & Dependencies

- requests – To send HTTP requests and fetch data from the IPL website
- BeautifulSoup – To parse and extract HTML data
- pandas – For data cleaning, processing, and CSV conversion
- numpy – For numerical calculations and statistical analysis
- matplotlib & plotly – For data visualization in Python

3. Hardware Requirements

- **Processor:** Intel Core i5 or higher
- **RAM:** Minimum 8GB (Recommended: 16GB for large datasets)
- **Storage:** At least 10GB of free space for storing CSV files and project data

4. Data Requirements

- **Auction Data Source:** Official IPL website (scraped using Python)
- **Dataset Format:** CSV (Containing player names, base price, sold price, teams, and categories)

5. Internet Requirements

- **Stable Internet Connection** (Required for scraping data and using Google Looker Studio)

Methodology

The **IPL Auction 2025** project follows a structured methodology consisting of data collection, processing, analysis, and visualization. The key steps are:

1. Data Collection (Web Scraping)

- **Tools Used:** requests, BeautifulSoup
- **Process:**
    - Sent HTTP requests to the official IPL website to extract auction data.
    - Parsed the HTML content to retrieve details like **player names, base price, sold price, teams, and player categories**.
    - Stored the extracted data in a structured format.

2. Data Processing & Storage

- **Tools Used:** pandas
- **Process:**
    - Converted the raw scraped data into a **CSV format** for easy handling.
    - Performed **data cleaning** to remove duplicates, missing values, and inconsistencies.
    - Structured the data into relevant fields for analysis.

3. Data Analysis (Python)

- **Tools Used:** pandas, numpy, matplotlib, seaborn
- **Process:**
    - Identified **top expensive players** and **highest team spending**.
    - Analyzed **price variations** between base price and final sold price.
    - Categorized players into **batsmen, bowlers, all-rounders, and wicketkeepers** and analyzed demand.
    - Compared **team-wise spending** and investment trends.

4. Data Visualization (Google Looker Studio)

- **Tools Used:** Google Looker Studio
- **Process:**
    - Designed an **interactive dashboard** with bar charts, pie charts, and line graphs.
    - Included **filters** for team-wise and player-wise data exploration.
    - Displayed **auction trends**, **highest-paid players**, and **category distributions** dynamically.

5. Insights & Conclusion

- Extracted **meaningful insights** about team strategies, player demand, and auction patterns.
- Provided **data-driven observations** on franchise spending and player market trends.

Workflow

Step 1: Web Scraping

Download Website HTML and save into File

[ ]:

[5]:
```
import requests
url = "https://www.iplt20.com/auction"
response = requests.get(url)
if response.status_code == 200:
   with open("web_file.txt", "w", encoding="utf-8") as file:
      file.write(response.text)
   print("File downloaded successfully.")
else:
   print("Failed to fetch file. Status code:", response.status_code)
```

```
File downloaded successfully.
```

[ ]:

Open HTML File in read mode

[ ]:

[7]:
```
with open("web_file.txt", "r") as file:
    html = file.read()
#html
```

[8]:
```
from bs4 import BeautifulSoup
soup = BeautifulSoup(html, "html.parser")
print(soup.prettify())
```

Scrap CSK Data

[ ]:

[12]:
```
table = soup.find_all("table",id="t3-chennai-super-kings")
#table
```

[13]:
```
for Table in table:
    Table_Header = []
    for x in Table.find_all('tr'):
        for y in x.find_all('th'):
            Table_Header.append(y.text)
Table_Header
```

[13]:
```
['Sr. No.', 'Player', 'Base Price', 'Winning Bid', 'Capped/Uncapped', '']
```

[14]:
```
for Values in table:
    Table_values = []
    for x in Values.find_all('tr')[1:]:
        td_tags = x.find_all('td')
        td_val = [y.text for y in td_tags]
        Table_values.append(td_val)
#Table_values
```

[15]:
```
import pandas as pd
df = pd.DataFrame (Table_values,columns=Table_Header)
df
```

Scrap DC Data

[ ]:

[17]:
```
table1 = soup.find_all("table",id="t3-delhi-capitals")
#table1
```

[18]:
```
for Values in table1:
    Table_values1 = []
    for x in Values.find_all('tr')[1:]:
        td_tags = x.find_all('td')
        td_val = [y.text for y in td_tags]
        Table_values1.append(td_val)
#Table_values1
```

[19]:
```
df1 = pd.DataFrame (Table_values1,columns=Table_Header)
df1
```

Scrap GT Data

[ ]:

[21]:
```
table2 = soup.find_all("table",id="t3-gujarat-titans")
#table2
```

[22]:
```
for Values in table2:
    Table_values2 = []
    for x in Values.find_all('tr')[1:]:
        td_tags = x.find_all('td')
        td_val = [y.text for y in td_tags]
        Table_values2.append(td_val)
#Table_values2
```

[23]:
```
df2 = pd.DataFrame (Table_values2,columns=Table_Header)
df2
```

Scrap KKR Data

[ ]:

25]:
```
table3 = soup.find_all("table",id="t3-kolkata-knight-riders")
#table3
```

26]:
```
for Values in table3:
    Table_values3 = []
    for x in Values.find_all('tr')[1:]:
        td_tags = x.find_all('td')
        td_val = [y.text for y in td_tags]
        Table_values3.append(td_val)
#Table_values3
```

27]:
```
df3 = pd.DataFrame (Table_values3,columns=Table_Header)
df3
```

Scrap LSG Data

[ ]:

[29]:
```
table4 = soup.find_all("table",id="t3-lucknow-super-giants")
#table4
```

[30]:
```
for Values in table4:
    Table_values4 = []
    for x in Values.find_all('tr')[1:]:
        td_tags = x.find_all('td')
        td_val = [y.text for y in td_tags]
        Table_values4.append(td_val)
#Table_values4
```

[31]:
```
df4 = pd.DataFrame (Table_values4,columns=Table_Header)
df4
```

Scrap MI Data

[ ]:

33]:
```
table5 = soup.find_all("table",id="t3-mumbai-indians")
#table5
```

34]:
```
for Values in table5:
    Table_values5 = []
    for x in Values.find_all('tr')[1:]:
        td_tags = x.find_all('td')
        td_val = [y.text for y in td_tags]
        Table_values5.append(td_val)
#Table_values5
```

35]:
```
df5 = pd.DataFrame (Table_values5,columns=Table_Header)
df5
```

Scrap PK Data

[ ]:

[37]:
```
table6 = soup.find_all("table",id="t3-punjab-kings")
#table6
```

[38]:
```
for Values in table6:
    Table_values6 = []
    for x in Values.find_all('tr')[1:]:
        td_tags = x.find_all('td')
        td_val = [y.text for y in td_tags]
        Table_values6.append(td_val)
#Table_values6
```

[39]:
```
df6 = pd.DataFrame (Table_values6,columns=Table_Header)
df6
```

Scrap RR Data

[ ]:

[41]:
```
table7 = soup.find_all("table",id="t3-rajasthan-royals")
#table7
```

[42]:
```
for Values in table7:
    Table_values7 = []
    for x in Values.find_all('tr')[1:]:
        td_tags = x.find_all('td')
        td_val = [y.text for y in td_tags]
        Table_values7.append(td_val)
#Table_values7
```

[43]:
```
df7 = pd.DataFrame (Table_values7,columns=Table_Header)
df7
```

Scrap RCB Data

```
[ ]:
```

```
[45]: table8 = soup.find_all("table",id="t3-royal-challengers-bengaluru")
#table8
```

```
[46]: for Values in table8:
    Table_values8 = []
    for x in Values.find_all('tr')[1:]:
        td_tags = x.find_all('td')
        td_val = [y.text for y in td_tags]
        Table_values8.append(td_val)
#Table_values8
```

```
[47]: df8 = pd.DataFrame (Table_values8,columns=Table_Header)
df8
```

Scrap SRH Data

```
 ]:
```

```
9]: table9 = soup.find_all("table",id="t3-sunrisers-hyderabad")
#table9
```

```
0]: for Values in table9:
    Table_values9 = []
    for x in Values.find_all('tr')[1:]:
        td_tags = x.find_all('td')
        td_val = [y.text for y in td_tags]
        Table_values9.append(td_val)
#Table_values9
```

```
1]: df9 = pd.DataFrame (Table_values8,columns=Table_Header)
df9
```

Step 2: Converts scraped data into CSV format

Append All DataFrame

```
[53]: Dataset = pd.concat([df,df1,df2,df3,df4,df5,df6,df7,df8,df9],axis=0,ignore_index = True,sort = False)
```

```
[54]: Dataset
```

[54]:

	Sr. No.	Player	Base Price	Winning Bid	Capped/Uncapped	
0	1	Noor Ahmad	2,00,00,000	10,00,00,000	Capped	
1	2	Ravichandaran Ashwin	2,00,00,000	9,75,00,000	Capped	
2	3	Devon Conway	2,00,00,000	6,25,00,000	Capped	
3	4	Syed Khaleel Ahmed	2,00,00,000	4,80,00,000	Capped	
4	5	Rachin Ravindra	1,50,00,000	4,00,00,000	Capped	
...	...	...	...	...	...	...
181	15	Swapnil Singh	30,00,000	50,00,000	Uncapped	
182	16	Mohit Rathee	30,00,000	30,00,000	Uncapped	
183	17	Abhinandan Singh	30,00,000	30,00,000	Uncapped	
184	18	Swastik Chhikara	30,00,000	30,00,000	Uncapped	
185	19	Manoj Bhandage	30,00,000	30,00,000	Uncapped	

186 rows × 6 columns

Convert Datatype of Columns

[56]:
```
Dataset['Winning Bid'] = Dataset['Winning Bid'].astype(str).str.replace(',', '').astype(int)
Dataset['Base Price'] = Dataset['Base Price'].astype(str).str.replace(',', '').astype(int)
```

[57]:
```
Dataset
```

[57]:

	Sr. No.	Player	Base Price	Winning Bid	Capped/Uncapped
0	1	Noor Ahmad	20000000	100000000	Capped
1	2	Ravichandaran Ashwin	20000000	97500000	Capped
2	3	Devon Conway	20000000	62500000	Capped
3	4	Syed Khaleel Ahmed	20000000	48000000	Capped
4	5	Rachin Ravindra	15000000	40000000	Capped
...	...	...	...	...	... ...
181	15	Swapnil Singh	3000000	5000000	Uncapped
182	16	Mohit Rathee	3000000	3000000	Uncapped
183	17	Abhinandan Singh	3000000	3000000	Uncapped
184	18	Swastik Chhikara	3000000	3000000	Uncapped
185	19	Manoj Bhandage	3000000	3000000	Uncapped

186 rows × 6 columns

Save Dataset in the form of CSV file ¶

[59]:
```
Dataset.to_csv("C:/Users/AMAN/Documents/IPL Auction Dataset.csv", index=True)
print("CSV file saved successfully!")
```

```
CSV file saved successfully!
```

Step 3: Perform Data Analysis

Find some insights from scraped data

[ ]:
```

```

Data Cleaning

[ ]:
```

```

shape and columns of data

[63]:
```
Dataset.columns
```

[63]:
```
Index(['Sr. No.', 'Player', 'Base Price', 'Winning Bid', 'Capped/Uncapped',
       ''],
      dtype='object')
```

[64]:
```
Dataset.shape
```

[64]:
```
(186, 6)
```

[ ]:
```

```

checking Null values

[66]:
```
Dataset.isnull().sum()
```

[66]:
```
Sr. No.           0
Player            0
Base Price        0
Winning Bid       0
Capped/Uncapped   0
                  0
dtype: int64
```

Insights from Data

[ ]:

Total Amount Spend By all Teams in Auction 2025

[70]:
```
Dataset['Winning Bid'].sum()
```

[70]:
```
6766000000
```

[ ]:

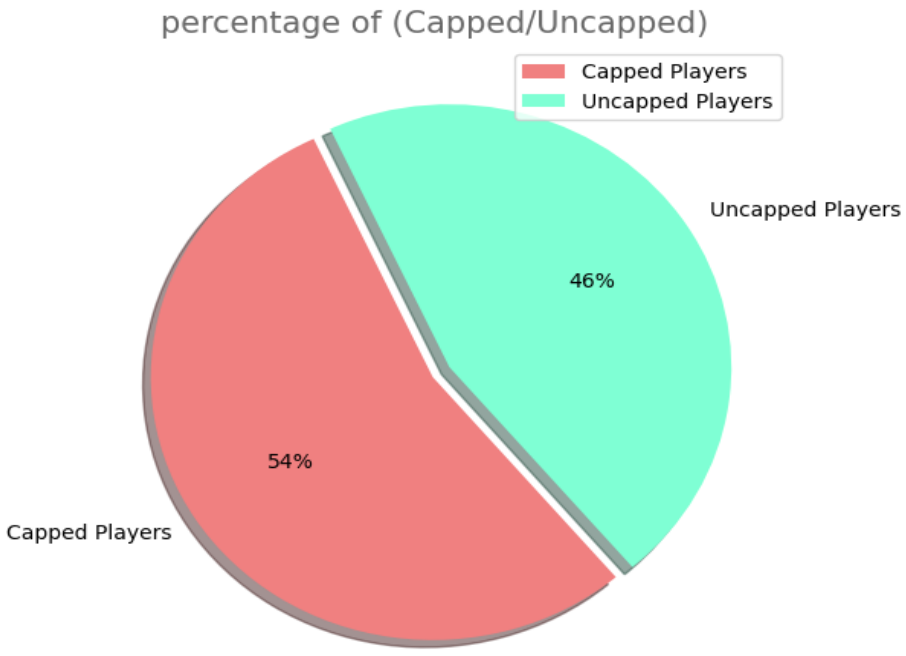
Total Number of Sold (Capped/Uncapped Players in Auction 2025

[72]:
```
Sold = Dataset.groupby('Capped/Uncapped')[['Winning Bid']].count()
Sold
```

[72]:

Capped/Uncapped	Winning Bid
Capped	101
Uncapped	85

[73]:
```
import matplotlib.pyplot as plt
explode = [0.03, 0.04]
plt.figure(figsize=(6,6))
color=['lightcoral','aquamarine']
plt.pie(Sold['Winning Bid'], labels=['Capped Players','Uncapped Players']
        , colors=color, autopct='%.0f%%', explode=explode,shadow = True,startangle = 115)
plt.title('percentage of (Capped/Uncapped) ',size=15,color='dimgrey')
plt.legend(loc='upper right')
plt.show()
```



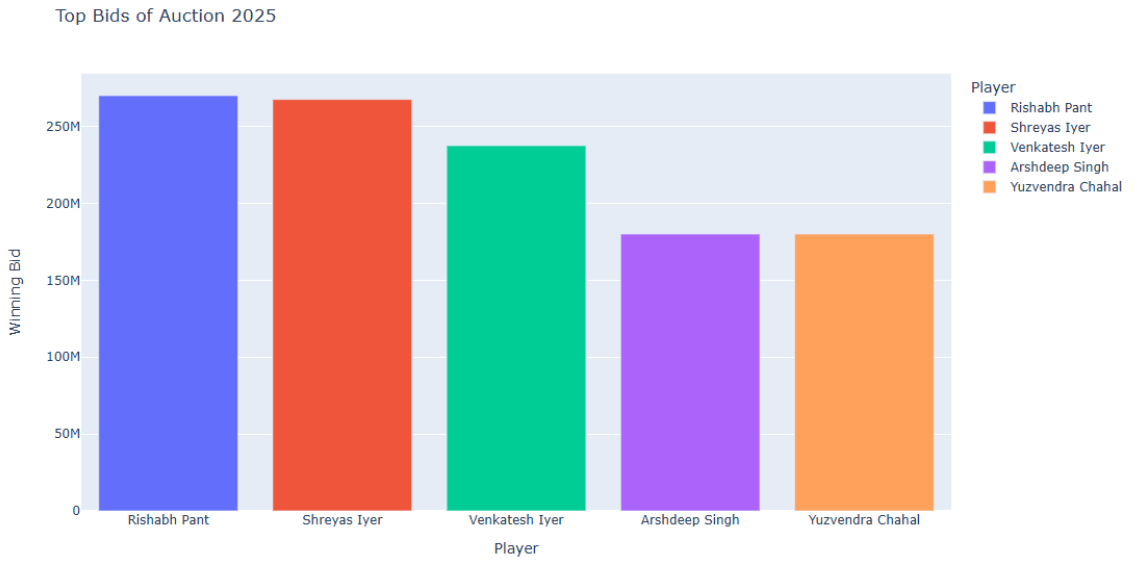
Top Bids of Auction 2025 ¶

[75]:
```
Data = Dataset.sort_values(by=['Winning Bid'],ascending=False).head(5)
Data
```

[75]:

	Sr. No.	Player	Base Price	Winning Bid	Capped/Uncapped
74	1	Rishabh Pant	20000000	270000000	Capped
111	1	Shreyas Iyer	20000000	267500000	Capped
59	1	Venkatesh Iyer	20000000	237500000	Capped
113	3	Arshdeep Singh	20000000	180000000	Capped
112	2	Yuzvendra Chahal	20000000	180000000	Capped

```
[76]: import plotly.express as px
      fig = px.bar(Data, x='Player', y='Winning Bid'
                   ,title='Top Bids of Auction 2025',color='Player')
      fig.update_layout(height=600)
      fig.show()
```



Dashboard

Future Scope

The IPL Auction 2025 project has significant potential for future improvements and expansions. Below are some key areas where the project can be enhanced:

1. Real-Time Data Updates

- Automate the web scraping process to fetch real-time auction updates and refresh the dataset dynamically.
- Implement a scheduled data pipeline to update the dashboard with live auction data.

2. Advanced Data Analytics

- Use machine learning models to predict player auction prices based on historical trends.
- Perform sentiment analysis using social media data to gauge player popularity before auctions.
- Analyze team performance vs. auction spending to evaluate investment effectiveness.

3. Enhanced Visualization

- Integrate Power BI or Tableau for more interactive and customizable dashboards.
- Add geo-mapping features to analyze player origins and their impact on auction demand.

4. Comparative Analysis

- Compare previous IPL auctions (e.g., 2023, 2024) with IPL 2025 to identify trends and changes in player demand.
- Study foreign vs. domestic player impact on team performance.

5. Web Application Development

- Develop a web-based application where users can explore auction trends interactively.
- Add search and filter options to let users query specific players, teams, or categories.

6. Expansion to Other Leagues

- Extend the project to cover other cricket leagues like Big Bash League (BBL), Pakistan Super League (PSL), and The Hundred.
- Analyze global auction trends in T20 leagues.

Conclusion

The IPL Auction 2025 project successfully integrates web scraping, data analysis, and visualization to provide valuable insights into the player auction process. By extracting real-time data from the official IPL website, the project efficiently collects and processes auction details, including player names, base price, sold price, teams, and player categories.

Using Python libraries like pandas, numpy, matplotlib, and seaborn, the project analyzes trends such as top purchases, team-wise spending, and player category distributions. The findings are then visualized through an interactive dashboard in Google Looker Studio, offering a clear and dynamic representation of auction trend

Learning Outcomes

The IPL Auction 2025 project provided valuable insights into various data science, analytics, and visualization techniques. The key learning outcomes from this project are:

1. Web Scraping & Data Extraction

- Learned how to use requests and BeautifulSoup to scrape real-time data from websites.
- Understood HTML parsing and extracting relevant information from web pages.

2. Data Processing & Cleaning

- Gained hands-on experience with pandas for data structuring, cleaning, and conversion into CSV format.
- Applied techniques to handle missing values, duplicates, and data inconsistencies.

3. Data Analysis & Visualization

- Used numpy, matplotlib, and seaborn to analyze and visualize auction trends.
- Explored team-wise spending, highest-paid players, and player demand through data insights.

4. Dashboard Development

- Built an interactive dashboard using Google Looker Studio.
- Learned how to create charts, filters, and data-driven visualizations for better insights.