VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science Engineering and Information Systems

Fall Semester 2024-25 MTech (Software Engineering)

SWE2011 – Big Data Analytics – B1+TB1

Project Based Component

FINAL REVIEW

IPL DATA ANALYSIS

Team Members:

MOHAMMAD ABUBAKAR G (21MIS0149)

MOGITH P (21MIS0204)

MOTHISWAR T B G (21MIS0258)

Faculty:

Prof. JAGANNATHAN J

Assistant Professor Sr.Grade I

TABLE OF CONTENTS

DECLARATION

We hereby declare that the report entitled **IPL DATA ANALYSIS** submitted by us for the **SWE2011-Big Data Analytics** to Vellore Institute of Technology is a record of Bonafide work carried out by me under the supervision of **Dr.J.Jagannathan**.

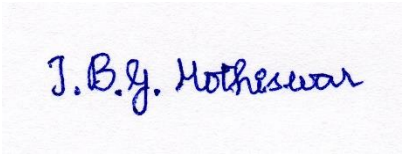
We further declare that the work reported in this report has not been submitted and will not be submitted, either in part or full, for any other courses in this institute or any other institute or university.

Place: Vellore

Date: 14.11.2024

Signature of the Candidate 1

Signature of the Candidate 2



Signature of the Candidate 3

1. ABSTRACT

The Indian Premier League (IPL) is a highly popular cricket tournament that attracts millions of fans globally. This study focuses on analyzing IPL data using Exploratory Data Analysis (EDA) techniques to uncover hidden patterns, trends, and insights. EDA is a critical step in data analysis that involves summarizing the main characteristics of the data, often with visual methods. By applying EDA to IPL data, we aim to explore various aspects such as team performance, player statistics, match outcomes, and venue influences. The analysis will involve data cleaning, transformation, and visualization to draw meaningful conclusions. Key performance indicators (KPIs) like win percentages, player averages, and run distributions will be examined. The findings from this analysis can provide valuable insights for teams, players, and fans, enhancing understanding and decision-making in future IPL seasons. This study highlights the power of EDA in sports analytics, particularly in a dynamic and competitive environment like the IPL.

Keywords: Indian Premier League (IPL), Exploratory Data Analysis (EDA), Cricket Analytics, Player statistics, Match outcomes, win percentages, run distributions.

2. INTRODUCTION

Big data analytics has become an integral part of businesses and organizations around the world. The ability to collect, store, and analyse massive amounts of data has provided companies with valuable insights into customer behaviour, market trends, and operational efficiencies. In recent years, the world of sports has also recognized the importance of big data analytics. Cricket has become a sport where data analytics can provide teams with a competitive edge.

In this project, we will explore the use of web scraping, Python, Pandas, and Power BI to perform data analytics on cricket data. Web scraping is the process of automatically collecting data from websites, and it is a powerful tool for gathering large amounts of data quickly. Python 3 is a popular programming language for data analysis due to its simplicity and ease of use. Pandas is a Python library that provides powerful tools for data manipulation and analysis. Power BI is a data visualization tool that allows users to create interactive reports and dashboards.

Cricket is a sport that generates a lot of data, from ball-by-ball records to player statistics. By collecting and analysing this data, we can gain insights into player performance, team strategies, and overall game trends. This project will involve collecting cricket data from a variety of sources, cleaning and processing the data using Pandas, and then visualizing the data using Power BI.

Overall, the aim of this project is to demonstrate the power of big data analytics in the world of cricket. By leveraging web scraping, Python, Pandas, and Power BI, we can gain valuable insights into the sport and help teams make data-driven decisions.

3. OBJECTIVE

This project harnesses big data analytics in cricket, using web scraping, Python, Pandas, and Power BI to gather, process, analyze, and visualize cricket data. By collecting extensive data, such as ball-by-ball records and player statistics from reliable sources, and cleaning it with Python and Pandas for consistency, we aim to uncover valuable insights into player performance, team strategies, and game trends. These insights will be presented in interactive Power BI dashboards, allowing for intuitive exploration and interpretation. By supporting data-driven decision-making, this project enables cricket teams and analysts to enhance strategies and performance, showcasing the impact of big data on the sport.

4. PROBLEM STATEMENT

In cricket, teams and analysts are increasingly seeking data-driven insights to enhance player performance, refine team strategies, and gain a competitive edge. However, the vast amount of cricket data available ranging from ball-by-ball match records to player and team statistics is often scattered across various sources and lacks structure, making it challenging to extract and interpret meaningful information efficiently. This project addresses the need for a systematic approach to collect, clean, analyse, and visualize cricket data, enabling stakeholders to make informed decisions. By leveraging web scraping, data manipulation tools in Python, and interactive visualizations in Power BI, this project aims to bridge the gap between raw cricket data and actionable insights that can drive team performance and strategic planning.

5. LITERATURE REVIEW

The study on IPL Data Analysis employed classification models to predict IPL match outcomes, focusing on the effectiveness of algorithms like Random Forest and Decision Trees. While the model demonstrated reasonable accuracy, its limitations include overfitting [1]. Another study used machine learning techniques, specifically Support Vector Machines (SVM) and Logistic Regression, to predict match outcomes. [2]. A separate work analysed and estimated IPL winners using machine learning, primarily utilizing ensemble methods like Gradient Boosting. Although the approach showed promise, the accuracy was sensitive to input feature selection, limiting its generalizability [3].

Another study explored Twitter sentiment analysis as a predictor of match outcomes, using Natural Language Processing (NLP) techniques. Despite capturing public sentiment trends, the model's accuracy was impacted by the noisy nature of social media data [4]. Research focused on data analysis and visualization using Microsoft Power BI revealed that while visualization tools effectively presented data insights, they were limited in predictive capabilities [5].

An additional study applied machine learning models such as Naïve Bayes and K-Nearest Neighbors (KNN) for match prediction, achieving moderate accuracy. However, the models struggled with complex patterns in the data [6]. Optimization techniques were also explored, with a study using a modified NSGA-II algorithm to create balanced IPL squads. Although the method provided a systematic approach to team selection, its effectiveness was limited by the assumptions made during model construction [7]. Another investigation analyzed cricket bowlers' kinematics using wearable sensors and machine learning. The study's main limitation was the variability in sensor data, which affected model accuracy [8].

Moreover, a study on measuring and predicting IPL team popularity using machine learning highlighted the challenge of accurately predicting popularity trends due to the dynamic nature of fan engagement and external factors [9]. Lastly, research on T20 World Cup winner prediction using machine learning techniques, while relevant, faced limitations in model accuracy due to the unpredictable nature of T20 matches and the small dataset size used for training [10].

6. DATA EXPLORATION

DATASET:

Dataset is collected by scraping from website

https://www.espncricinfo.com/records/tournament/team-match-results/indian- premier-league-2024-15940

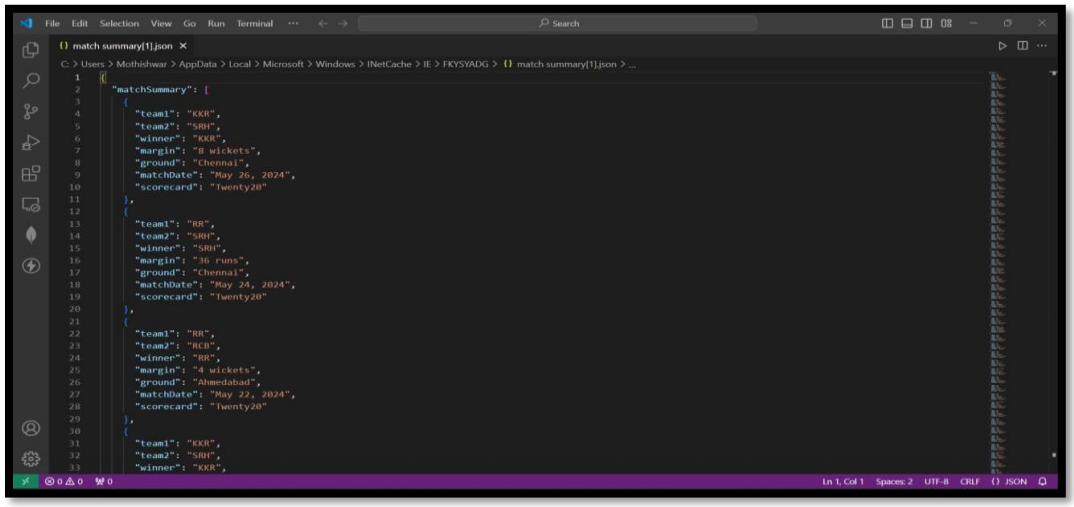
It is done using Bright Data, an online tool that uses proxy networks for building scrapers that work seamlessly. Unlike other scraping methods, Bright Data uses JavaScript to crawl websites and can scrap datasets in a customized way. The results can be stored in JSON file which can be used for preprocessing.

BrightData: https://brightdata.com/

INTERACTION CODE:

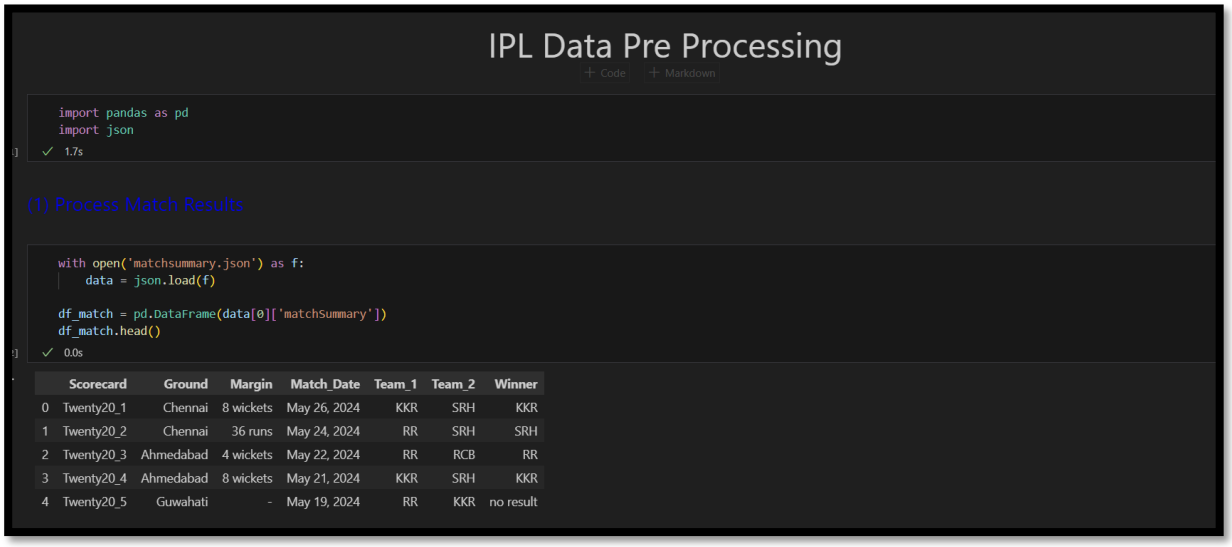
```
navigate('https://www.espncricinfo.com/records/tournament/team-match-results/indian-premier-league-2024-15940');
collect(parse());
```

Match summary.json (output json file):



DATA PROCESSING:

- Applying some transformations such as
- New column for Out/Not_Out
- Replacing special characters in player's name
- Adding match_id to the column in which it is absent using team name in both column
- Converting to .csv



IPL Data Pre Processing

+ Code + Markdown

```
import pandas as pd
import json
```

✓ 1.7s

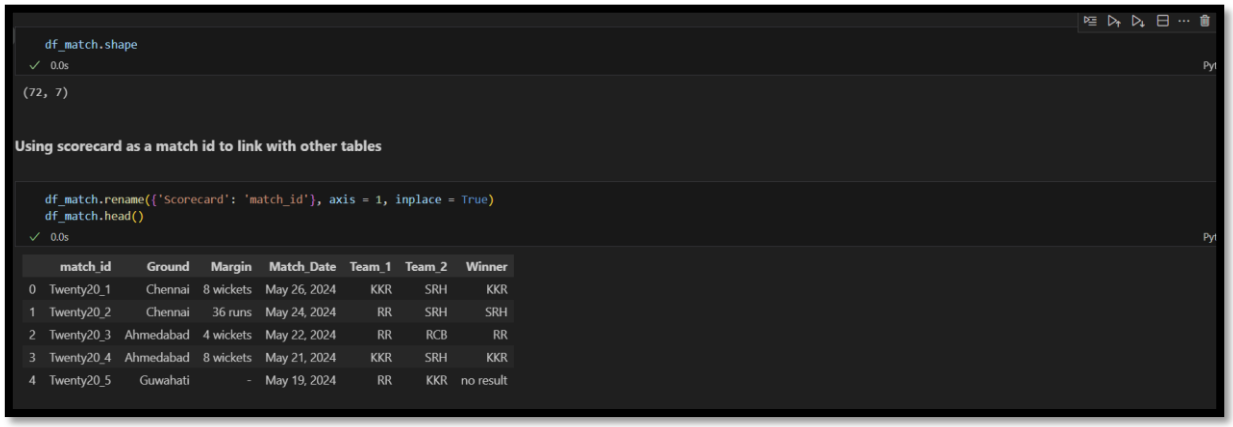
(1) Process Match Results

```
with open('matchsummary.json') as f:
    data = json.load(f)

df_match = pd.DataFrame(data[0]['matchSummary'])
df_match.head()
```

✓ 0.0s

	Scorecard	Ground	Margin	Match_Date	Team_1	Team_2	Winner
0	Twenty20_1	Chennai	8 wickets	May 26, 2024	KKR	SRH	KKR
1	Twenty20_2	Chennai	36 runs	May 24, 2024	RR	SRH	SRH
2	Twenty20_3	Ahmedabad	4 wickets	May 22, 2024	RR	RCB	RR
3	Twenty20_4	Ahmedabad	8 wickets	May 21, 2024	KKR	SRH	KKR
4	Twenty20_5	Guwahati	-	May 19, 2024	RR	KKR	no result



```
df_match.shape
```

✓ 0.0s

```
(72, 7)
```

Using scorecard as a match id to link with other tables

```
df_match.rename({'Scorecard': 'match_id'}, axis = 1, inplace = True)
df_match.head()
```

✓ 0.0s

	match_id	Ground	Margin	Match_Date	Team_1	Team_2	Winner
0	Twenty20_1	Chennai	8 wickets	May 26, 2024	KKR	SRH	KKR
1	Twenty20_2	Chennai	36 runs	May 24, 2024	RR	SRH	SRH
2	Twenty20_3	Ahmedabad	4 wickets	May 22, 2024	RR	RCB	RR
3	Twenty20_4	Ahmedabad	8 wickets	May 21, 2024	KKR	SRH	KKR
4	Twenty20_5	Guwahati	-	May 19, 2024	RR	KKR	no result

Using scorecard as a match id to link with other tables

```
df_match.rename({'Scorecard': 'match_id'}, axis = 1, inplace = True)
df_match.head()
```

✓ 0.0s

	match_id	Ground	Margin	Match_Date	Team_1	Team_2	Winner
0	Twenty20_1	Chennai	8 wickets	May 26, 2024	KKR	SRH	KKR
1	Twenty20_2	Chennai	36 runs	May 24, 2024	RR	SRH	SRH
2	Twenty20_3	Ahmedabad	4 wickets	May 22, 2024	RR	RCB	RR
3	Twenty20_4	Ahmedabad	8 wickets	May 21, 2024	KKR	SRH	KKR
4	Twenty20_5	Guwahati	-	May 19, 2024	RR	KKR	no result

(2) Process Batting Summary

```
with open('BattingSummary.json') as f:
    data = json.load(f)
    all_records = []
    for rec in data:
        all_records.extend(rec['battingSummary'])

df_batting = pd.DataFrame(all_records)
df_batting.head(11)
```

✓ 0.0s

	match	teamInnings	battingPos	batsmanName	dismissal	runs	balls	4s	6s	SR
0	KKR Vs SRH	SRH	1	AbhishekSharma	b Starc	2	5	0	0	40.00
1	KKR Vs SRH	SRH	2	TravisHead	c Rahmanullah Gurbaz b Arora	0	1	0	0	0.00
2	KKR Vs SRH	SRH	3	RahulTripathi	c Ramandeep Singh b Starc	9	13	1	0	69.23
3	KKR Vs SRH	SRH	4	AidenMarkram	c Starc b Russell	20	23	3	0	86.95
4	KKR Vs SRH	SRH	5	NitishKumar Reddy	c Rahmanullah Gurbaz b Harshit Rana	13	10	1	1	130.00
5	KKR Vs SRH	SRH	6	HeinrichKlaasen	b Harshit Rana	16	17	1	0	94.11
6	KKR Vs SRH	SRH	7	ShahbazAhmed	c Narine b Varun	8	7	0	1	114.28
7	KKR Vs SRH	SRH	8	AbdulSamad	c Rahmanullah Gurbaz b Russell	4	4	0	0	100.00
8	KKR Vs SRH	SRH	9	PatCummins(c)	c Starc b Russell	24	19	2	1	126.31
9	KKR Vs SRH	SRH	10	JaydevUnadkat	lbw b Narine	4	11	0	0	36.36
10	KKR Vs SRH	SRH	11	BhuvneshwarKumar		0	1	0	0	0.00

(3) Process Bowling Summary

```
with open('BlowingSummary.json') as f:
    data = json.load(f)
    all_records = []
    for rec in data:
        all_records.extend(rec['bowlingSummary'])
all_records[:2]
```

✓ 0.0s


```
[{'match': 'KKR Vs SRH',
  'bowlingTeam': 'KKR',
  'bowlerName': 'Mitchell Starc',
  'overs': '3',
  'maiden': '0',
  'runs': '14',
  'wickets': '2',
  'economy': '4.66',
  '0s': '11',
  '4s': '2',
  '6s': '0',
  'wides': '0',
  'noBalls': '0'},
 {'match': 'KKR Vs SRH',
  'bowlingTeam': 'KKR',
  'bowlerName': 'Vaibhav Arora',
  'overs': '3',
  'maiden': '0',
```

```
(4) Process Players Information

with open('playerInfo.json') as f:
    data = json.load(f)
✓ 0.0s

df_players = pd.DataFrame(data)

print(df_players.shape)
df_players.head(10)
✓ 0.0s
(123, 6)
```

	name	team	battingStyle	bowlingStyle	playingRole	description
0	Abhishek Sharma	SRH	Left hand Bat	Slow Left arm Orthodox	Allrounder	
1	Travis Head	SRH	Left hand Bat	Right arm Offbreak	Top order Batter	A talented, aggressive left-hand batter earmar...
2	Rahul Tripathi	SRH	Right hand Bat	Right arm Medium	Top order Batter	A technically correct middle-order batter, Rah...
3	Aiden Markram	SRH	Right hand Bat	Right arm Offbreak	Opening Batter	The first South Africa captain to win a World ...
4	Nitish Kumar Reddy	SRH	Right hand Bat	Right arm Medium fast	Batting Allrounder	Nitish Kumar Reddy first made headlines when h...
5	Heinrich Klaasenâ€	SRH	Right hand Bat	Right arm Offbreak	Wicketkeeper Batter	A wicketkeeper-batter known for his belligeren...
6	Shahbaz Ahmed	SRH	Left hand Bat	Slow Left arm Orthodox	Allrounder	Shahbaz Ahmed was born in Mewat, Haryana, the ...
7	Abdul Samad	SRH	Right hand Bat	Legbreak	Batter	

7. METHODS

1. Data Collection through Web Scraping:

- Gathering cricket-related data from multiple online sources using web scraping techniques.
- Python libraries such as **BeautifulSoup** and **Scrapy** will be used to automate the data extraction process, pulling relevant information such as match details, player statistics, and ball-by-ball records.

This approach enables the collection of extensive data rapidly, ensuring a rich dataset for analysis.

2. Data Preprocessing and Cleaning with Pandas:

Raw data often contains inconsistencies, missing values, and irregular formats.

- Using the Pandas library in Python, we will clean and preprocess the data to ensure consistency and accuracy.
- Handling null values, correcting data types, and transforming data into a structured format.
- Cleaning and organizing the data will allow for effective analysis in later stages.

3. Data Analysis Using Python:

After preprocessing, Python will be used to analyse the data and derive insights. **Statistical and exploratory data analysis techniques** will be applied to understand player and team performance trends, player strengths, and overall game patterns. Python's analytical libraries, such as **NumPy, Pandas, and Matplotlib** will aid in performing these calculations and creating initial visual representations.

4. Data Visualization with Power BI:

The analyzed data will be imported into Power BI for advanced visualization. Using Power BI, we will design **interactive dashboards** and reports that **showcase key insights** in an accessible format. These visualizations will allow users to explore performance trends, team strategies, and other critical factors through interactive charts, graphs, and filters.

5. Insight Generation and Reporting:

Finally, the findings will be compiled into a comprehensive report that highlights the main insights derived from the data. The report will include data-driven recommendations for enhancing team strategies and player performance, emphasizing the value of big data analytics in cricket.

8. MODELING AND RESULTS

Requirement Scoping:

The requirements for a player to be in the best 11 based on the category of the player is as follows

OPENERS:

PARAMETERS	DESCRIPTION	CRITERIA
Batting Average	Average runs scored in an innings	> 30
Strike Rate	No of runs scored per 100 balls	> 140
Innings Batted	Total Innings batted	> 3
Boundary %	% of runs scored in boundaries	> 50
Batting Position	Order in which the batter played	< 4

MIDDLE ORDERS:

PARAMETERS	DESCRIPTION	CRITERIA
Batting Average	Average runs scored in an innings	> 40
Strike Rate	No of runs scored per 100 balls	> 125
Innings Batted	Total Innings batted	> 3
Avg. Balls Faced	Average balls faced by the batter in an innings	> 20
Batting Position	Order in which the batter played	> 2

FINISHERS:

PARAMETERS	DESCRIPTION	CRITERIA
Batting Average	Average runs scored in an innings	> 25
Strike Rate	No of runs scored per 100 balls	> 130
Innings Batted	Total Innings batted	> 3
Avg. Balls Faced	Average balls faced by the batter in an innings	> 12
Batting Position	Order in which the batter played	> 4
Innings Bowled	Total Innings Bowled by the bowler	> 1

ALL – ROUNDERS:

PARAMETERS	DESCRIPTION	CRITERIA
Batting Average	Average runs scored in an innings	> 15
Strike Rate	No of runs scored per 100 balls	> 140
Innings Batted	Total Innings batted	> 2
Batting Position	Order in which the batter played	> 4
Innings Bowled	Total Innings bowled	> 2
Bowling Economy	Average runs allowed per over	< 7
Bowling Strike Rate	Average no. of balls required to take a wicket	< 20

FAST BOWLERS:

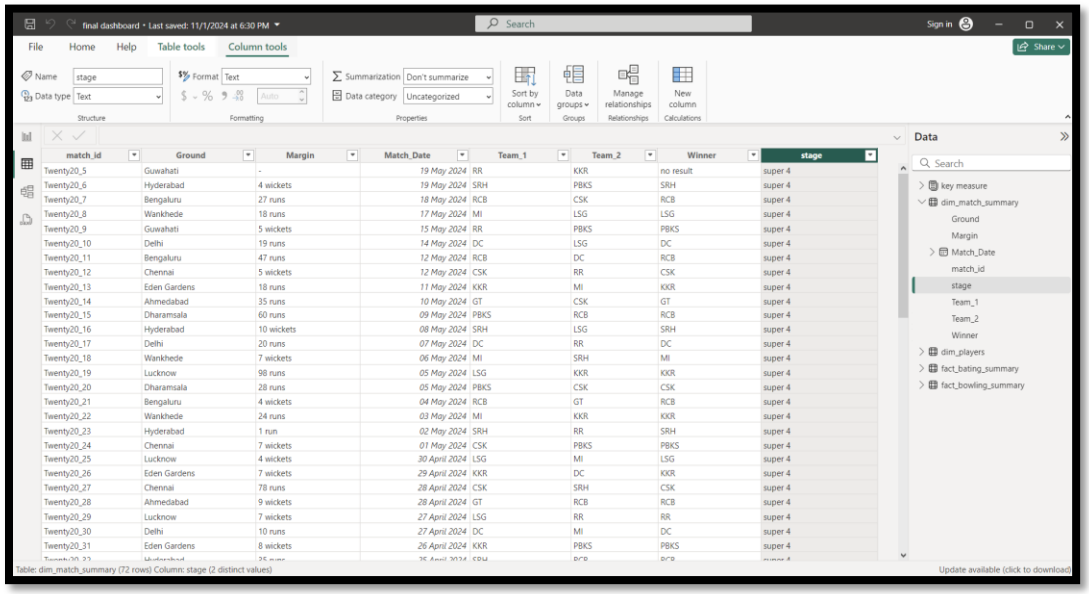
PARAMETERS	DESCRIPTION	CRITERIA
Innings Bowled	Total Innings bowled	> 4
Bowling Economy	Average runs allowed per over	< 7
Bowling Strike Rate	Average no. of balls required to take a wicket	< 16
Bowling Style	Bowling style of the player	= "%Fast%"
Bowling Average	No. of runs allowed per wicket	< 20
Dot Ball %	% of dot balls bowled	> 40

Data Transformation:

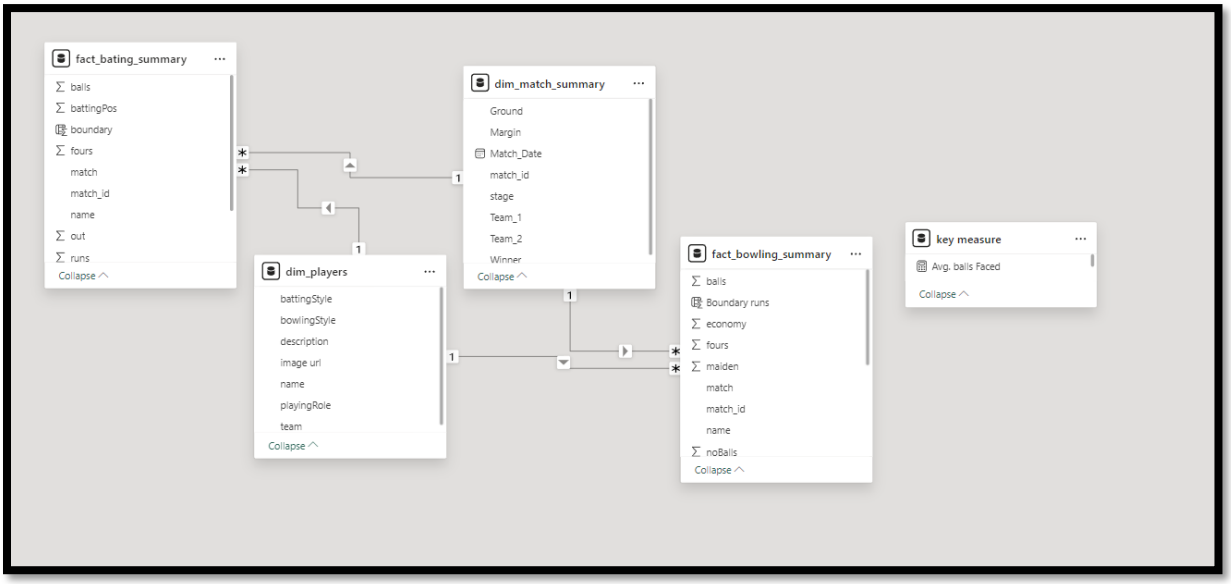
- Power query is a component inside power bi to do data transformation
- Using power query some transformations are applied.

Examples:

- 1st row as header
- Remove (c) after captain name
- Removing duplicates
- New column for indicating super 12 or qualifier based on date
- New column for no.of.balls from overs



Data Modeling:



Dax Measures:

- DAX refers to data analysis expression used to build visuals with measures
- Created DAX measures according to the requirements we scoped previously for each category of players

Examples:

- Total_runs = sum(batting_summary[runs])
- Total innings batted = count(fact_batting_summary[match_id])
- Total balls faced = sum(fact_batting_summary[balls])

Sno	Measures	Description / Purpose	DAX FORMULA	TABLE
1	Total Runs	Total number of runs scored by the batsman	Total Runs = SUM(fact_batting_summary[runs])	fact_batting_summary
2	Total Innings Batted	Total number of innings a batsman got a chance to bat	Total Innings Batted = COUNT(fact_batting_summary[match_id])	fact_batting_summary
3	Total Innings Dismissed	To find the number of innings batsman got out	SUM(fact_batting_summary[out])	fact_batting_summary
4	**Batting Average**	**Average runs scored in an innings**	**Batting Avg = DIVIDE([Total Runs],[Total Innings Dismissed],0)**	**fact_batting_summary**
5	Total balls Faced	Total number of balls faced by the batsman	total balls faced = SUM(fact_batting_summary[balls])	fact_batting_summary
6	**Strike Rate**	**No of runs scored per 100 balls**	**Strike rate = DIVIDE([Total Runs],[total balls faced],0)*100**	**fact_batting_summary**
7	Batting Position	Batting position of a player	Batting Position = ROUNDUP(AVERAGE(fact_batting_summary[batting_pos]),0)	fact_batting_summary
8	Boundary %	Percentage of boundaries scored by the Batsman	Boundary % = DIVIDE(SUM(fact_batting_summary[Boundary runs]),[Total Runs],0)	fact_batting_summary
9	Avg. balls Faced	Average balls faced by the batter in an innings	AVERAGE(fact_batting_summary[balls])	fact_batting_summary
10	Wickets	Total number of wickets taken by a bowler	wickets = SUM(fact_bowling_summary[wickets])	fact_bowling_summary
11	balls Bowled	Total number of balls bowled by the bowler	balls Bowled = SUM(fact_bowling_summary[balls])	fact_bowling_summary
12	Runs Conceded	Total runs conceded by the bowler	Runs Conceded = SUM(fact_bowling_summary[runs])	fact_bowling_summary
13	**Bowling Economy**	**Average number of runs conceded in an over**	**Economy = DIVIDE([Runs Conceded], ([balls Bowled]/6),0)**	**fact_bowling_summary**
14	**Bowling Strike Rate**	**Number of balls bowled per wicket**	**Bowling Strike Rate = DIVIDE([balls Bowled], [wickets],0)**	**fact_bowling_summary**
15	Bowling Average	No. of runs allowed per wicket	Bowling Average = DIVIDE([Runs Conceded],[wickets],0)	fact_bowling_summary
16	Total Innings Bowled	Total number of innings bowled by a bowler	Total Innings Bowled = DISTINCTCOUNT(fact_bowling_summary[match_id])	fact_bowling_summary

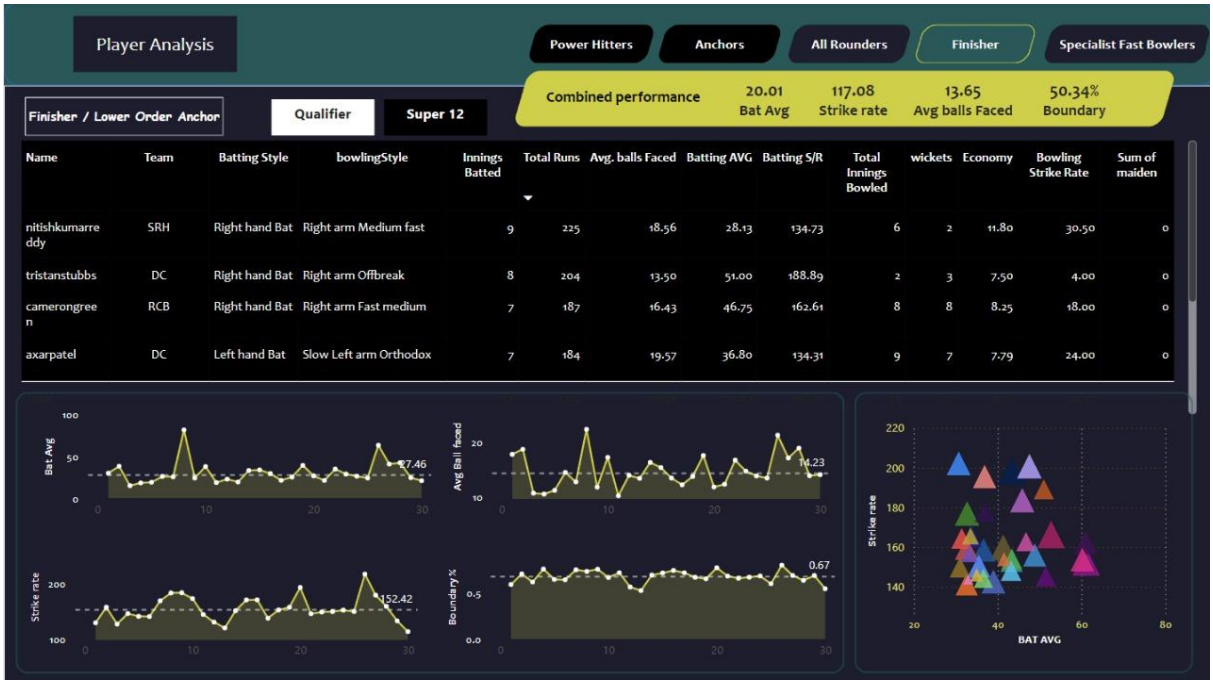
Results:

The following images are the screenshots of the dashboard created using IPL 2024 data.

POWER HITTERS / OPENERS:

MIDDLE ORDERS:

FINISHERS:



ALL ROUNDERS:

BOWLERS:

9. CONCLUSION AND FUTURE DEVELOPMENT

Conclusion:

In this project, we explored the use of web scraping, Python, Pandas, and Power BI to perform data analytics on cricket data. We collected data from various sources, including ESPN Cricinfo and used Python and Pandas to preprocess and analyze the data. We then visualized the results using Power BI, which allowed us to gain insights into player and team performance.

Through our analysis, we were able to identify key performance indicators for cricket, including batting averages, bowling strike rates, and fielding efficiency. We also identified trends and patterns in player and team performance, which can be used to make informed decisions about strategy and tactics.

Overall, our project demonstrates the potential of big data analytics for cricket and provides a useful framework for future research in this area.

Future Work:

There are several avenues for future work in this area. One potential direction is to expand our analysis to include additional data sources, such as video data or biometric data from wearable sensors. This could provide a more comprehensive view of player and team performance and allow for more detailed analysis of specific skills and techniques.

Another potential direction is to apply machine learning techniques to our data, such as regression analysis or decision trees, to predict outcomes and identify key factors that contribute to success in cricket matches.

Finally, we could explore the use of big data analytics for other sports, such as football, basketball, or baseball, to identify similarities and differences in performance analysis across different sports. This could provide valuable insights into the use of big data analytics for performance analysis in sports more broadly.

10. REFERENCES

[1] Paul, P., Datta, P. R., & Kar, A. (2023, April). Classification Model to Predict the Outcome of an IPL Match. In Analytics Global Conference (pp. 83-111). Cham: Springer Nature Switzerland.

[2] Avinash, M., & Vishwakarma, H. (2024). IPL MATCH PREDICTION USING MACHINE LEARNING TECHNIQUES. Journal of Research Administration, 6(1).

[3] Kian, G. S., Rafi, S. K., Bhagyaraju, G., & Subramanyam, D. S. Analyzing and Estimating the IPL Winner using Machine Learning.

[4] Sharma, K., Hota, L., Tikkiwal, V. A., & Kumar, A. (2024). Exploring Twitter Sentiments for Predicting Match Outcomes in The Game of Cricket. Procedia Computer Science, 235, 152-162.

[5] Jaipurkar, M., Ragit, N., Tambuskar, C., & Saraf, P. D. (2023, July). IPL Data Analysis and Visualization Using Microsoft Power BI Tool. In 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT) (pp. 1-6). IEEE.

[6] Singhal, A., Agarwal, D., Singh, E., Valecha, R., & Malik, R. (2022). IPL Analysis and Match Prediction. In Intelligent System Design: Proceedings of INDIA 2022 (pp. 29-38). Singapore: Springer Nature Singapore.

[7] Verma, S., Pandey, V., Pant, M., & Snasel, V. (2022). A balanced squad for Indian premier league using modified NSGA-II. IEEE Access, 10, 100463-100477.

[8] Rana, M., & Mittal, V. (2022). Analysis of front leg kinematics of cricket bowler using wearable sensors and machine learning. IEEE Sensors Journal, 22(22), 22053-22061.

[9] Kumar, Y., Sharma, H., & Pal, R. (2021, September). Popularity Measuring and Prediction Mining of IPL Team Using Machine Learning. In 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO) (pp. 1-5). IEEE.

[10] Basit, A., Alvi, M. B., Jaskani, F. H., Alvi, M., Memon, K. H., & Shah, R. A. (2020, November). ICC T20 Cricket World Cup 2020 winner prediction using machine learning techniques. In 2020 IEEE 23rd International Multitopic Conference (INMIC) (pp. 1-6). IEEE.