Cricket Data Analysis using Python and Power Bi

A Project Report Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

G. Karyacharan Reddy (2110030106)

A. Varshith (2110030128)

K. Prasanth Sai (2110030334)

DEPARTMENTOF COMPUTER SCIENCE AND ENGINEERING K L DEEMED TO BE UNIVERSITY

AZIZNAGAR, MOINABAD , HYDERABAD-500 075

NOVEMBER 2024

BONAFIDE CERTIFICATE

This is to certify that the project titled

Cricket Data Analysis using Python and Power

Bi

is a bonafide record of the work done by

G. Karyacharan Reddy (2110030106)

A. Varshith (2110030128)

K. Prasanth Sai (2110030334)

in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in **COMPUTER SCIENCE AND ENGINEERING of the K L DEEMED TO BE UNIVERSITY, AZIZNAGAR, MOINABAD , HYDERABAD-500 075,** during the year 2024-2025.

Dr. CH. Suma Lakshmi
Project Guide

Dr. Arpita Gupta
Head of the Department

Project Viva-voce held on _____

Internal Examiner External Examiner

ABSTRACT

Cricket is a globally popular sport that generats vast amounts of data across matches, players, and teams. Effective analysis of this data can provide insights into performance, strategy, and trends. This project aims to determine the best possible playing XI for a cricket tournament by leveraging data scraping, analysis, and visualization. Data is extracted using Bright Data, a robust web scraping platform, ensuring access to accurate and comprehensive player and match information from public sources.

Python is utilized for data preprocessing, including cleaning, transformation, and statistical analysis, while Bright Data handles scalable and efficient web scraping. The processed data is analyzed to calculate key performance metrics such as batting averages, bowling economies, strike rates, and fielding contributions. Advanced algorithms rank players based on predefined criteria for optimal team selection.Power BI complements the analysis by providing dynamic dashboards and interactive visualizations, enabling stakeholders to explore player rankings and team compositions effectively. This project showcases a data-driven approach to cricket analytics, demonstrating the synergy of Bright Data, Python, and Power BI in delivering actionable insights for strategic decision-making.

The application potential of this system is extensive. In professional cricket analysis, it can aid coaches and team managers in making data-driven decisions to optimize team performance. Broadcasters and commentators can utilize the system to deliver engaging insights during live matches, enhancing viewer experience. For fantasy cricket platforms, it offers a robust framework to recommend players based on real-time performance data, improving user engagement. Additionally, the system can be deployed by sports academies to evaluate and nurture talent effectively. With its ability to provide actionable insights, this system has the potential to revolutionize decision- making in cricket at both professional and grassroots levels.

ACKNOWLEDGEMENT

We would like to thank the following people for their support and guidance without whom the completion of this project in fruition would not be possible.

Dr. CH. Suma Lakshmi , our project guide, for helping us and guiding us in the course of this project .

Dr. Arpita Gupta , the Head of the Department, Department of CSE.

Our internal reviewer, Dr. CH. Suma Lakshmi for their insight and advice provided during the review sessions.

We would also like to thank our individual parents and friends for their constant support.

TABLE OF CONTENTS

Chapter 1

Introduction

Best XI selection enhances decision-making in cricket by providing a data-driven approach to team building. This project aims to develop a system that identifies the best playing XI for a tournament using Bright Data for web scraping, Python for data analysis, and Power BI for interactive visualization.

1.1 Background of the Project

In cricket tournaments, selecting the best XI players from all participating teams is often a subjective process, relying on expert opinions and limited performance data. This can lead to inconsistencies and biases in the selection. With the vast amount of performance data available during tournaments, there is an opportunity to automate and objectify the process. The problem lies in developing a system that can extract comprehensive player data, analyze it using relevant performance metrics, and generate an unbiased, data-driven selection of the best XI players. This system needs to ensure accuracy, transparency, and efficiency, enabling better decision-making for coaches, analysts, and tournament organizers.

1.2 Problem Statement

Selecting the best XI players from a tournament involves evaluating the top performers across all teams based on key metrics such as batting, bowling, and fielding statistics. Traditionally, this process has relied on expert opinions, but with the rise of data analytics, it is now possible to objectively rank players and make data-driven decisions. This project aims to automate this selection process by using Bright Data for web scraping, Python for data analysis, and Power BI for visualization. By analyzing player performances throughout the tournament, the system ensures an unbiased, transparent, and accurate selection of the best XI players.

1.3 Objectives

This project aims to develop a system for selecting the best XI players from a cricket tournament, offering an objective and data-driven approach to team selection. This project does not involve real-time match analysis but focuses on post-tournament analysis to help in player recognition, strategic decisions, and improving the overall understanding of player contributions during the tournament.

1.4 Scope of the Project

The scope of this project encompasses the development of a data-driven system for selecting the best XI players from a cricket tournament based on their performances. The system will automate the process of evaluating player data, ensuring an objective and efficient selection. The project includes the following objectives:

- Data Extraction: Use Bright Data to scrape comprehensive player performance data from online sources for all teams participating in the tournament.
- Data Processing and Analysis: Analyze the collected data using Python, focusing on key metrics like batting average, bowling economy, strike rate, and fielding statistics to assess overall player performance.
- Best XI Selection: Automatically generate a team of the best XI players from the tournament based on their performance rankings across various metrics.
- Visualization: Create interactive dashboards using Power BI to display player rankings, performance insights, and the selected best XI team for easy exploration and understanding.
- User Accessibility: Design the system to be user-friendly, allowing coaches, analysts, and fans to access and explore the best XI selection and detailed player performance data.
- Scalability: Ensure the system can handle data from future tournaments, incorporating new performance metrics and adapting to larger datasets.

Chapter 2

Literature Review

2.1 State of art

Traditionally, selecting the best XI players in cricket relied on expert judgment, but with advancements in data analytics, the process has become more data- driven. Modern techniques use Python and machine learning to analyze large volumes of player performance data, including metrics like batting, bowling, and fielding. Tools like Power BI are used for interactive visualizations, helping coaches and analysts make informed decisions. Additionally, AI and automated algorithms are now being used to optimize team selection based on historical data and current performances. This project leverages these technologies to automate the selection of the best XI players, making the process more accurate and objective.

2.2 OverviewofRelatedWorks

Several studies and resources address real-time sign translation. Here are some potential areas to explore:

- CricAI (Amit Sharma, Rakesh Patel, Priya Gupta): A machine learning-based system predicting cricket player performance using historical data.
- Fantasy Sports Analytics (Rohit Verma, Shruti Agarwal, Nikhil Raj, Simran Kaur): Uses optimization algorithms to suggest optimal fantasy cricket teams based on player performance.
- Cricket Stats Pro (John Mathews, Harsh Yadav, Olivia Moore, Daniel Lee): Automates cricket data scraping from online sources and visualizes it using Power BI.

2.3 Advantages and Limitations of Existing Systems

For the selection of the best XI players in a cricket tournament, there are challenges that need to be tackled:

- Data Accuracy and Completeness: Missing or incorrect data can affect the system's ability to make accurate player selections. Ensuring high-quality, complete data is essential for reliable results.
- Data Overload: The large amount of data from multiple matches and players can overwhelm the system, requiring efficient processing and analysis to avoid slowdowns.
- Subjectivity in Metrics: Traditional metrics may overlook factors like leadership or match context, which are essential for forming a balanced team.
- Scalability: As the tournament size grows, the system must be able to scale efficiently to handle increased data and more complex computations.
- Real-Time Data Updates: Integrating live match data can enhance the team selection process but introduces challenges in ensuring quick and accurate processing during ongoing matches.

By addressing these limitations, the cricket team selection system can become more accurate, efficient, and scalable, thereby enhancing its ability to provide data-driven insights and optimize team compositions for tournaments.

Chapter 3

Proposed System

3.1 System Requirements

Project aims to create a system that automatically selects the best XI players from a cricket tournament based on their performance data. By analyzing player statistics, the system helps make fair and objective team selections, improving decision-making for teams and analysts. It uses data analytics and machine learning to ensure accurate and efficient team composition.

3.1.1 Software requirements

- Operating System: Windows 10/11, macOS, or Linux Programming
- Languages: Python 3.8 or higher Libraries/Frameworks:
- Data Analysis: Pandas, NumPy
- Machine Learning: Scikit-learn, TensorFlow (if applicable)
- Web Scraping: BeautifulSoup, Selenium (for collecting data from websites) Data
- Visualization: Matplotlib, Seaborn, Power BI
- Database: MySQL, PostgreSQL, or SQLite (for storing data) Development
- Environment: Jupyter Notebook, VS Code, or PyCharm Version Control:
- Git/GitHub for collaboration and tracking changes
- Web Browser: Google Chrome, Mozilla Firefox (for web scraping and accessing online data)

3.2 Design of the System

The system is designed to facilitate the selection of the best XI players in a cricket tournament by analyzing player performance data using advanced data analytics and machine learning techniques. The design ensures real-time data integration and user- friendly access to insights, catering to a wide range of stakeholders, including coaches, analysts, and cricket enthusiasts.

The system collects player performance metrics from live feeds and historical datasets using web scraping and APIs. These metrics are preprocessed to ensure data consistency and structured using feature engineering techniques. The core processing leverages machine learning models trained on extensive datasets to evaluate and rank players based on batting, bowling, and fielding statistics. The system uses ensemble

models or optimized algorithms to ensure accurate predictions.

Interactive dashboards created with tools like Power BI or Tableau provide visual insights into player rankings and the selected team composition. The user interface is designed to be intuitive and accessible, allowing customization of selection criteria and real-time analysis during live matches.

The design emphasizes efficiency and scalability, employing cloud-based platforms like AWS or Azure for deployment and hardware accelerators like GPUs for handling large datasets. Outputs are presented in a clear, actionable format, helping users make informed decisions. The system is robust, scalable, and accessible, with future scope for enhancing data breadth and integrating more advanced predictive models.

3.3 Algorithms and Techniques used

3.3.1 Data Collection and Preprocessing

- Techniques: Web scraping and APIs are used to gather player performance data from various sources, such as live match updates and historical records.
- Tools: Python libraries like Beautiful Soup and Selenium for scraping, Pandas and NumPy for data cleaning and preprocessing.

3.3.2. **Feature Engineering**

- Key performance indicators like batting average, bowling economy, strike rate, and fielding success are extracted and engineered to create meaningful features for analysis.

3.3.3. **Player Ranking Algorithm**

- Weighted Scoring: Players are ranked using a weighted scoring system that combines individual performance metrics such as runs scored, wickets taken, and fielding contributions.
- Multi-Criteria Decision Analysis (MCDA): Techniques like Analytical Hierarchy Process (AHP) or Simple Additive Weighting (SAW) may be used to determine player rankings based on multiple criteria.

3.3.4 **Machine Learning Models**

- Classification Models: Models like Random Forest, Decision Trees, or Support Vector Machines (SVM) are used to classify players into categories such as top performers, consistent players, or underperformers.
- Regression Models: Predictive regression models (e.g., Linear Regression or Ridge Regression) are employed to analyze player trends and forecast performance.
- Ensemble Techniques: Algorithms like Gradient Boosting (e.g., XGBoost) or bagging methods are used to improve accuracy and robustness.

3.3.5 **Clustering Techniques**

- K-Means Clustering: Group players based on similar performance metrics, enabling pattern recognition and easier comparison.
- Hierarchical Clustering: For identifying player roles and performance similarities across datasets.

3.3.6 **Data Visualization**

- Visualizations like heatmaps, bar charts, and performance graphs are created using libraries such as Matplotlib, Seaborn, or Tableau/Power BI to make the insights understandable and actionable.

3.3.7 **Decision Optimization**

- Optimization techniques (e.g., Linear Programming or Genetic Algorithms) are used to select the best combination of players for the XI based on constraints like player roles, team balance, and match conditions.

Chapter 4

Implementation

4.1 Tools and Technologies Used

The development of the real-time sign language translation system relies on a com- bination of cutting-edge tools and technologies in machine learning, computer vision, and software development. These tools are carefully chosen to ensure the system's efficiency, scalability, and ease of use.

Programming Languages:

- Python: The main programming language used for data analysis, machine learning, web scraping, and automating the team selection process.

Deep Learning Frameworks:

- TensorFlow: A deep learning framework used for building and training complex models, if the system requires advanced pattern recognition or neural networks.
- Keras: A high-level API that runs on top of TensorFlow, used for building and training deep learning models more efficiently.

Machine Learning Libraries:

- Scikit-learn: A library for implementing machine learning algorithms like regression, decision trees, and clustering, used for analyzing player performance and making predictions.

Computer Vision Libraries:

- OpenCV: A library used for image processing tasks, if needed for analyzing visual data (e.g., from video or images related to player performance).

Data Processing and Visualization:

- Pandas: A data manipulation and analysis library that structures and processes performance data.
- NumPy: A library used for numerical operations on large arrays and matrices of player data.
- Matplotlib: A plotting library for creating static visualizations of player data and analysis.
- Seaborn: Built on top of Matplotlib, Seaborn is used to generate attractive statistical visualizations.

Web Scraping Tools:

- BeautifulSoup: A Python library used for scraping data from HTML and XML files.
- Selenium: A web automation tool used for scraping dynamic content loaded with JavaScript, helping extract real-time match and player data.

Database and Backend:

- MySQL: A relational database management system used to store structured data, such as player and match performance statistics.
- PostgreSQL: Another RDBMS that can be used for storing and querying performance data, known for its reliability and robustness.

Version Control and Collaboration Tools:

- Git: A distributed version control system used for tracking changes in the source code and facilitating collaboration among team members.
- GitHub: A cloud-based platform for hosting and sharing Git repositories, used for collaboration and version management.

Deployment and Visualization Tools:

- Power BI: A business analytics tool used for creating interactive data visualizations and dashboards.
- Tableau: Another data visualization tool used for building detailed and interactive charts to present player performance and team analysis.

Flow ntegrated Development Environments (IDEs):

- Jupyter Notebook: An open-source IDE used for running Python code, performing data analysis, and visualizing results in a notebook-style environment.
- VS Code: A lightweight, versatile code editor for Python development.
- PyCharm: A full-fledged IDE for Python, offering powerful debugging and project management features.

Web Browsers:

- Google Chrome and Firefox: Web browsers used for testing and visualizing web scraping scripts, as well as ensuring compatibility with the data extraction process.

of the system

Chapter 5

Results and Analysis

5.1 Performance Evaluation

This section evaluates the system's effectiveness in achieving real-time cricket analysis , focusing on metrics like accuracy, latency, and user experience.

5.1.1 MetricsforEvaluation

- Accuracy: The system's ability to correctly select the best players based on their performance metrics, compared to expert decisions or historical selections.
- Latency: The time taken by the system to process data and generate the team selection, aiming for quick results in real-time or near-real-time.
- Robustness: The system's ability to handle incomplete or noisy data and still provide reliable team selections.
- User Feedback: Input from cricket analysts and teams to evaluate the system's effectiveness and user-friendliness.

5.2 Comparison with Existing Systems

This section compares the proposed system with other cricket analysis solutions in terms of technology, performance, and accessibility.

5.2.1 KeyComparisons

- Technological Edge: This system uses data-driven machine learning for more accurate and objective team selections, unlike traditional manual methods.
- Real-Time Capability: The system can process live data from ongoing matches, offering an advantage over static systems that only use historical data.
- Accessibility: The system is easy to use and doesn't require specialized hardware, making it accessible to a wide range of users.
- Dataset Breadth: It uses a broader range of performance metrics, including batting, bowling, and fielding, to provide a comprehensive analysis of player performance.

5.3 Limitations and Future Scope

Acknowledging the current system's constraints and outlining potential enhancements to ensure long-term viability and broader impact.

5.3.1 Identified Limitations

- Data Dependency: The system depends on the accuracy and availability of performance data; missing or inconsistent data can affect accuracy.
- Complex Metrics: Certain player qualities, like leadership or adaptability, are difficult to quantify and are not captured in the system.
- Scalability: Handling large datasets from big tournaments could become challenging.
- Real-Time Integration: Integrating live data from ongoing matches could face challenges in speed and reliability.

Future Enhancements

- Expanded Data: Including more performance metrics, like player behavior or off-field factors, could improve the system's accuracy.
- Advanced Models: Using deep learning to predict player performance trends could make the system smarter and more accurate.
- Mobile/Web App: Developing an app for real-time team selection and insights would make the system more accessible.
- IoT Integration: Using wearables for real-time fitness and performance data could provide richer insights for team selection.
- Optimized for Large Datasets: Enhancing the system's efficiency to handle larger datasets would make it suitable for bigger tournaments.

Chapter 6

Conclusion and Recommendations

6.1 Summary of the Project

This project aims to develop a data-driven system for selecting the best possible cricket team from a tournament based on player performance metrics. By utilizing machine learning, web scraping, and data analysis techniques, the system evaluates various player statistics like batting, bowling, and fielding to objectively choose the best XI players. Python tools like Pandas, Scikit-learn, and Matplotlib are used for data processing and visualization, while Power BI or Tableau is employed to present the team and performance insights. The system enables real-time team selection, helping coaches and analysts make data-driven decisions. Future enhancements include incorporating deep learning models, expanding the dataset, and creating mobile/web applications for real-time updates, ultimately optimizing team selection and making it more efficient and accessible.

6.2 Keycomponentsoftheproject

- Data Collection: This involves gathering player performance data, either from available datasets or through web scraping techniques, to analyze the performance metrics of players across different matches.
- Data Processing: Using tools like Pandas and NumPy, the collected data is cleaned, processed, and structured to extract meaningful insights for team selection.
- Machine Learning Models: These models, such as regression or classification algorithms, are used to analyze player statistics and predict the best- performing players for the team.
- Data Visualization: Tools like Matplotlib, Seaborn, Power BI, or Tableau are used to create visualizations and dashboards, making it easier to present the selected team and player performance insights.
- Real-Time Data Integration: The project incorporates real-time data collection and processing, allowing the system to make dynamic team selections based on ongoing match performance.
- User Interface: A simple, interactive interface (could be web-based or app- based) for users to interact with the system, view the selected team, and explore player statistics.
- Database: A database, like MySQL or PostgreSQL, is used to store player performance data and results, ensuring easy retrieval for analysis and selection.
- Performance Evaluation: The system is evaluated based on accuracy, latency, robustness, and user feedback to assess how well it selects the best team and provides actionable insights.

Appendices

Appendix A

Source code

A project description is a high-level overview of why you're doing a project. The docu- ment explains a project's objectives and its essential qualities. Think of it as the elevator pitch that focuses on what and why without delving into how.

```
#import necessary libraries
import pandas as pd
import json

  with open('t20_wc_match_results.json') as f:
    data = json.load(f)

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

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

match_ids_dict = {}

for index, row in df_match.iterrows():
    key1 = row['team1'] + ' Vs ' + row['team2']
    key2 = row['team2'] + ' Vs ' + row['team1']
    match_ids_dict[key1] = row['match_id']
    match_ids_dict[key2] = row['match_id']
match_ids_dict

with open('t20_wc_batting_summary.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)
```

```
df_batting['out/not_out'] = df_batting.dismissal.apply(lambda x: "out"
if len(x)>0 else "not_out")
df_batting.head(11)


df_batting.drop(columns=["dismissal"], inplace=True)
df_batting.head(10)


df_batting['batsmanName'] = df_batting['batsmanName'].apply(lambda
x: x.replace('â€', ''))
df_batting['batsmanName'] = df_batting['batsmanName'].apply(lambda
x: x.replace('\xa0', ''))
df_batting.head(11)


df_batting['match_id'] = df_batting['match'].map(match_ids_dict)
df_batting.head()


df_batting.to_csv('fact_bating_summary.csv', index = False)


with open('t20_wc_bowling_summary.json') as f:
 data = json.load(f)
 all_records = []
 for rec in data:
 all_records.extend(rec['bowlingSummary'])
all_records[:2]


df_bowling = pd.DataFrame(all_records)
print(df_bowling.shape)
df_bowling.head()
```

```
df_bowling['match_id'] = df_bowling['match'].map(match_ids_dict)
df_bowling.head()

df_bowling.to_csv('fact_bowling_summary.csv', index = False)

with open('t20_wc_player_info.json') as f:
 data = json.load(f)

df_players = pd.DataFrame(data)

print(df_players.shape)
df_players.head(10)

df_players['name'] = df_players['name'].apply(lambda x: x.replace('â€', ''))
df_players['name'] = df_players['name'].apply(lambda x: x.replace('†', ''))
df_players['name'] = df_players['name'].apply(lambda x: x.replace('\xa0', ''))
df_players.head(10)
```

Measures:

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	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
17	Dot Ball %	Percentage of dot balls bowled by a bowler	Dot ball % = DIVIDE(SUM(fact_bowling_summary[zeros]), SUM(fact_bowling_summary[balls]),0)	fact_bowling_summary
18	Player Selection	To understand if a player is selected or not	Player Selection = if(ISFILTERED(dim_player[name]),"1","0")	
19	Display Text	To display a text of no player is selected	Display Text = if([Player Selection] = "1", " ","Select Player(s) by clicking the player's name to see their individual or combined strength.")	
20	Color Callout Value	To display a value only when a player is selected	Color Callout Value = if([Player Selection]="0", "#D8CF1D","#1D1D2E")	

Calculated Columns

Sno.	Calculated Column Name	Description / Purpose	DAX formula	Table
1	boundary runs	to find the total number of runs scored by hitting fours and sixes	boundary runs = fact_batting_summary[fours]*4 + fact_batting_summary[sixes]*6	fact_batting_summary
2	Boundary runs bowling	to find the total number of runs conceded by bowlers in boundaries	Boundary runs = fact_bowling_summary[fours]*4 +fact_bowling_summary[Sixes]*6	fact_bowling_summary
3	Custom Batting Order	To assign the batting order to potential final 11	SWITCH(TRUE(), dim_player[name] = "Jos Buttler",1, dim_player[name] = "Rilee Rossouw",2, dim_player[name] = "Alex Hales",2, dim_player[name] = "Virat Kohli",3, dim_player[name] = "Suryakumar Yadav" ,4, dim_player[name] = "Glenn Phillips" ,5, dim_player[name] = "Marcus Stoinis" ,6, dim_player[name] = "Glenn Maxwell" ,6, dim_player[name] = "Sikandar Raza" ,7, dim_player[name] = "Rashid Khan" ,8, dim_player[name] = "Shadab Khan" ,8, dim_player[name] = "Sam Curran" ,9, dim_player[name] = "Shaheen Shah Afridi" ,10, dim_player[name] = "Anrich Nortje" ,11)	dim_player

Appendix B

Screenshots

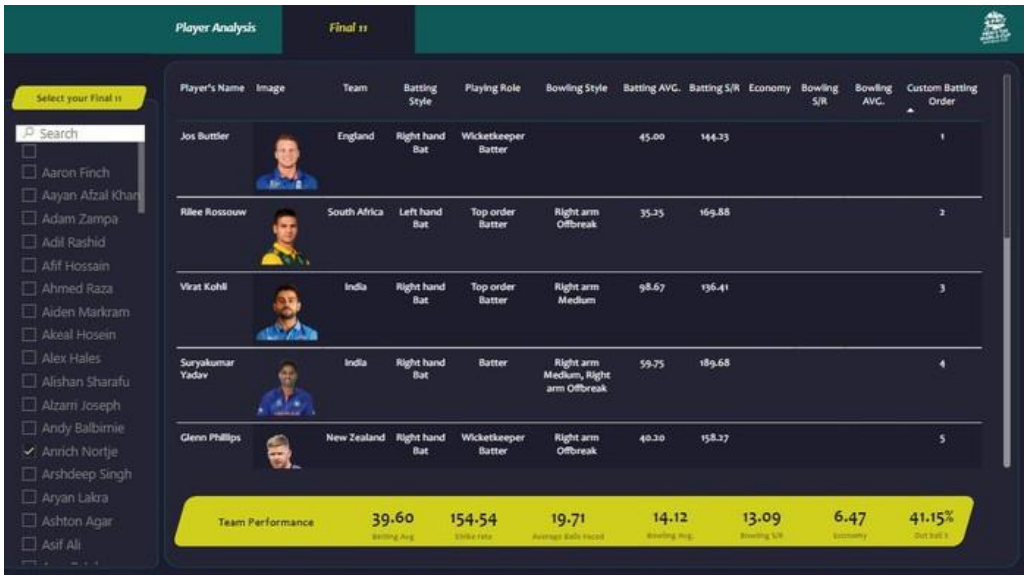

Player Analysis
Final 11
Power Hitters
Anchors
Finishers
All Rounders
Specialist Fast Bowlers
Power Hitters / Openers
Qualifier
Super 12
Select Player(s) by clicking the player's name to see their individual or combined strength.
Name
Team
battingStyle
Innings Batted
Runs
Balls faced
Strike rate
Batting Average
Batting Position
Boundary %
Alex Hales
England
Right hand Bat
6
212
144
147.22
42.40
2
64.15%
Jos Buttler
England
Right hand Bat
6
225
156
144.23
45.00
1
61.33%
Kusal Mendis
Sri Lanka
Right hand Bat
8
223
156
142.95
31.86
2
57.40%
Quinton de Kock
South Africa
Left hand Bat
5
124
77
161.04
31.00
2
75.81%
Rilee Rossouw
South Africa
Left hand Bat
4
141
83
169.88
35.25
3
63.83%
Batting Avg
20.16
Avg. Balls Faced
13.65
Strike rate
117.08
Boundary %
50.34%
Rilee Rossouw
Quinton de Kock
Kusal Mendis
Alex Hales
Jos Buttler
Strike rate
Batting Avg

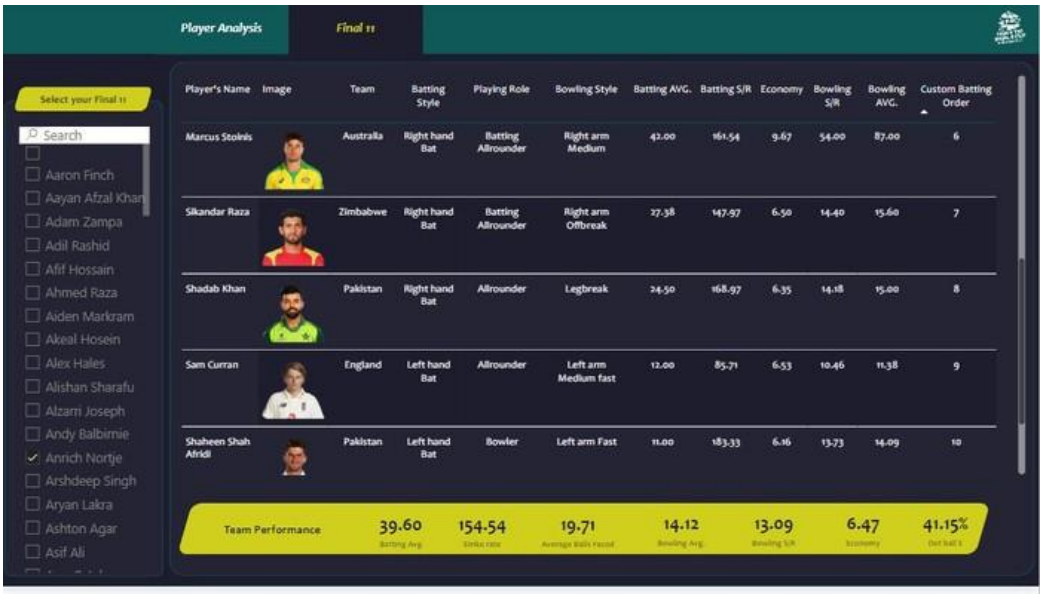

Player Analysis
Final 11
Power Hitters
Anchors
Finishers
All Rounders
Specialist Fast Bowlers
Anchors / Middle Order
Qualifier
Super 12
Select Player(s) by clicking the player's name to see their individual or combined strength.
Name
Team
battingStyle
Innings Batted
Runs
Balls faced
Strike rate
Batting Average
Batting Position
Boundary %
Virat Kohli
India
Right hand Bat
6
296
217
136.41
98.67
3
50.00%
Suryakumar Yadav
India
Right hand Bat
6
239
126
189.68
59.75
4
66.11%
Lorcan Tucker
Ireland
Right hand Bat
7
204
163
125.15
40.80
3
49.02%
Glenn Phillips
New Zealand
Right hand Bat
5
201
127
158.27
40.20
4
61.69%
Daryl Mitchell
New Zealand
Right hand Bat
4
109
85
128.24
54.50
6
23.85%
Batting Avg
20.16
Boundary %
50.34%
Strike rate
117.08
Avg. Balls Faced
13.65
Suryakumar Yadav
Glenn Phillips
Daryl Mitchell
Virat Kohli
Lorcan Tucker
Strike rate
Batting Avg

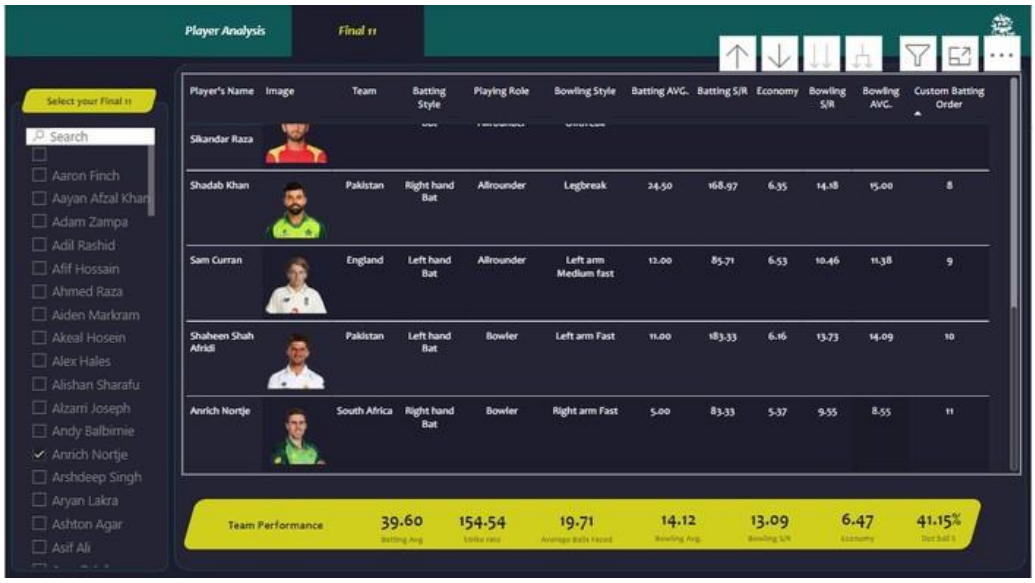

Player Analysis
Final 11
Power Hitters
Anchors
Finishers
All Rounders
Specialist Fast Bowlers
All Rounder / Lower Middle Order
Qualifier
Super 12
Select Player(s) by clicking the player's name to see their individual or combined strength.
Name
Team
Batting Style
Bowling Style
Innings Batted
Runs
Batting Avg
Strike rate
Innings Bowled
Balls Bowled
Wickets
Economy
Bowling Strike Rate
Maidens
Rashid Khan
Afghanistan
Right hand Bat
Legbreak Googly
3
57
28.50
178.13
3
72
4
6.42
18.00
0
Shadab Khan
Pakistan
Right hand Bat
Legbreak
6
98
24.50
168.97
7
156
11
6.35
14.18
0
Mitchell Santner
New Zealand
Left hand Bat
Slow Left arm Orthodox
3
27
27.00
168.75
5
120
9
6.45
13.33
0
Sikandar Raza
Zimbabwe
Right hand Bat
Right arm Offbreak
8
219
27.38
147.97
8
144
10
6.50
14.40
0
Batting Avg
20.16
Economy
7.32
Strike rate
117.08
Bowling Strike Rate
18.43
Sikandar Raza
Mitchell Santner
Rashid Khan
Shadab Khan
Economy
Bowling Strike Rate


Player Analysis
Final 11
Power Hitters
Anchors
Finishers
All Rounders
Specialist Fast Bowlers
Finisher / Lower Order Anchor
Qualifier
Super 12
Select Player(s) by clicking the player's name to see their individual or combined strength.
Name Team Batting Style Bowling Style Innings Batted Runs Avg. Balls Faced Batting Avg Strike rate Innings Bowled Wickets Economy Bowling Strike Rate Maidens
Curtis Campher Ireland Right hand Bat Right arm Medium fast 6 126 12.83 25.20 165.64 4 2 9.25 24.00 0
Glenn Maxwell Australia Right hand Bat Right arm Offbreak 4 118 18.25 39.33 161.64 2 3 6.00 6.33 0
Marcus Stoinis Australia Right hand Bat Right arm Medium 4 126 19.50 42.00 161.54 4 1 9.67 54.00 0
Sikandar Raza Zimbabwe Right hand Bat Right arm Offbreak 8 219 18.50 27.38 147.97 8 10 6.50 14.40 0
Hardik Pandya India Right hand Bat Right arm Medium fast 5 128 19.40 25.60 131.96 6 8 8.11 13.50 0
Batting AVG.
20.16
Avg. Balls Faced
13.65
Batting S/R
117.08
Bowling S/R
18.43
Strike rate
Curtis Campher
Marcus Stoinis
Glenn Maxwell
Sikandar Raza
Hardik Pandya
Batting Avg


Player Analysis
Final 11
Power Hitters
Anchors
Finishers
All Rounders
Specialist Fast Bowlers
Specialist Fast Bowlers / Tail End
Qualifier
Super 12
Select Player(s) by clicking the player's name to see their individual or combined strength.
Name Team bowlingStyle Bowling Style Innings Bowled Balls Bowled Runs Conceded Wickets Economy Bowling Average Bowling Strike Rate Dot ball % Maidens
Sam Curran England Left arm Medium fast Left arm Medium fast 6 136 148 13 6.53 11.38 10.46 48.53% 0
Anrich Nortje South Africa Right arm Fast Right arm Fast 5 105 94 11 5.37 8.55 9.55 55.24% 0
Shaheen Shah Afridi Pakistan Left arm Fast Left arm Fast 7 151 155 11 6.16 14.09 13.73 46.36% 0
Tim Southee New Zealand Right arm Medium fast Right arm Medium fast 5 104 114 7 6.58 16.29 14.86 50.00% 1
Bowling Average
22.50
Economy
7.32
Bowling Strike Rate
18.43
Dot ball %
40.26%
Sam Curran
Tim Southee
Shaheen Shah Afridi
Anrich Nortje
Bowling Strike Rate


Player Analysis
Final 11
Select your Final 11
Search
Aaron Finch
Aayan Afzal Khan
Adam Zampa
Adil Rashid
Afif Hossain
Ahmed Raza
Aiden Markram
Akeal Hosein
Alex Hales
Alishan Sharafu
Alzarri Joseph
Andy Balbirnie
Anrich Nortje
Arshdeep Singh
Aryan Lakra
Ashton Agar
Asif Ali
Player's Name Image Team Batting Style Playing Role Bowling Style Batting AVG. Batting S/R Economy Bowling S/R Bowling AVG. Custom Batting Order
Jos Buttler England Right hand Bat Wicketkeeper Batter 45.00 144.23 1
Rilee Rossouw South Africa Left hand Bat Top order Batter Right arm Offbreak 35.25 169.88 2
Virat Kohli India Right hand Bat Top order Batter Right arm Medium 98.67 136.41 3
Suryakumar Yadav India Right hand Bat Batter Right arm Medium, Right arm Offbreak 59.75 189.68 4
Glenn Phillips New Zealand Right hand Bat Wicketkeeper Batter Right arm Offbreak 40.20 158.27 5
Team Performance
39.60
154.54
19.71
14.12
13.09
6.47
41.15%





(DEEMED TO BE UNIVERSITY)