Mentorness ML Internship

T20 World Cup Data Analysis

SAI VARMA PALOJI

Google Colab Link

Datasets Link

Contents

Introduction

In this project, we analyze a dataset containing cricket match statistics. The dataset includes information such as match details, player performances, team scores, and match events. The objective of this analysis is to uncover insights into player and team performances, identify key match events, and provide insights based on the findings.

Data Exploration

We explore the dataset to understand its structure and characteristics. We analyze key statistics and distributions to gain insights into the dataset's content.

Data set Overview

No of Columns: 44
No of Rows: 9814
Description:
Match details: Match ID, names of the home and away teams, innings details, over-by-over progress.
- **Player statistics:** Runs, wickets, and other player-specific details.
- **Match events**: Boundaries, wickets, retired hurt scenarios, and more.
- **Commentary text**: Pre-match, in-match, and post-match commentary snippets.

Player Statistics

Batsman runs per match

```
[18] # Explore Batsman performances
     max_runs_per_match = df.groupby(['match_name', 'batsman1_name']).agg({
           'match_id': 'nunique',
           'batsman1_runs': 'max',
           'batsman1_balls': 'max'
       }).reset_index()
```

Total runs by batsman in entire tournament

```
[19] total_runs_by_batsman = max_runs_per_match.groupby('batsman1_name').agg({
           'match_id': 'sum',
           'batsman1_runs': 'sum',
           'batsman1_balls': 'sum'
       }).reset_index()
     total_runs_by_batsman = total_runs_by_batsman.sort_values(by='batsman1_runs', ascending=False)

     # Calculate strike rate
     total_runs_by_batsman['strike_rate'] = (total_runs_by_batsman['batsman1_runs'] / total_runs_by_batsman['batsman1_balls']) * 100
```

Bowlers wickets per match

```
wkts_per_bowler_per_match = df.groupby(['match_name', 'bowler1_name']).agg({
      'match_id': 'nunique',
      'bowler1_overs': 'max',
      'bowler1_wkts': 'max',
      'bowler1_maidens': 'max',
      'bowler1_runs': 'max'
  }).reset_index()
wkts_per_bowler_per_match.head()
```

Over by Over Progress

```
# Analyze over-by-over progress
over_by_over_progress = df.groupby(['match_name', 'current_innings', 'over'])['runs'].agg(['sum','mean']).reset_index()
over_by_over_progress = over_by_over_progress.rename(columns={'sum': 'total_runs', 'mean': 'avg_runs'})
print(over_by_over_progress.head())
```

```
  match_name current_innings  over  total_runs  avg_runs
0  AFG v ENG             AFG     1           2  0.333333
1  AFG v ENG             AFG     2           9  1.500000
2  AFG v ENG             AFG     3           4  0.666667
3  AFG v ENG             AFG     4           5  0.833333
4  AFG v ENG             AFG     5           7  1.166667
```

Event Interface

```
# Analyze boundary events
boundary_events = df[df['isBoundary'] == True]
boundaries_per_match = boundary_events.groupby('match_name').size().reset_index(name='num_boundaries')
print(boundaries_per_match.head())
```

```
  match_name  num_boundaries
0  AFG v ENG              18
1   AFG v SL              29
2  AUS v AFG              41
3  AUS v IRE              35
4   AUS v NZ              37
```

```
[34] # Analyze wicket events per match
wicket_events = df[df['wicket_id'].notnull()]
wickets_per_match = wicket_events.groupby('match_name').size().reset_index(name='num_wickets')
print(wickets_per_match.head())
```

```
  match_name  num_wickets
0  AFG v ENG           15
1   AFG v SL           11
2  AUS v AFG           15
3  AUS v IRE           15
4   AUS v NZ           13
```

Connected to Python 3 Google Compute Engine backend

Event Interface

```python
# Analyze the distribution of boundary types
boundary_types = df[df['isBoundary'] == True]['shortText'].value_counts()
print(boundary_types)
```

```
shortText
Starc to Tucker, FOUR          7
Rajitha to Hales, FOUR         5
Ngarava to de Kock, FOUR       4
Chatara to O'Dowd, FOUR        4
Davey to Campher, FOUR         4
                              ..
Maxwell to Tector, FOUR        1
Cummins to Tucker, FOUR        1
Starc to Delany, FOUR          1
Maxwell to Delany, FOUR        1
Frylinck to Theekshana, SIX    1
Name: count, Length: 947, dtype: int64
```

```python
event_impact.head()
```

	match_id	shortText
1	1298179	Stokes to Mohammad Rizwan, 1 wide
2	1298179	Stokes to Mohammad Rizwan, 1 no ball
9	1298179	Woakes to Babar Azam, 1 wide
21	1298179	Woakes to Mohammad Rizwan, SIX
34	1298179	Woakes to Babar Azam, FOUR

Performance Evaluation

1.Batsman

```
total_runs_by_batsman = max_runs_per_match.groupby('batsman1_name').agg({
      'match_id': 'sum',
      'batsman1_runs': 'sum',
      'batsman1_balls': 'sum'
  }).reset_index()
total_runs_by_batsman = total_runs_by_batsman.sort_values(by='batsman1_runs', ascending=False)

# Calculate strike rate
total_runs_by_batsman['strike_rate'] = (total_runs_by_batsman['batsman1_runs'] / total_runs_by_batsman['batsman1_balls']) * 100
```

```
# Calculate average
# For average, we need the number of times each batsman got out
# We can get this from the original dataset where 'wicket_id' is not null
batsman_out = df[df['wicket_id'].notnull()]
# Group by 'batsman1_name' and count the occurrences to get the number of times each batsman got out
batsman_out_count = batsman_out.groupby('batsman1_name')['wkt_batsman_name'].count().reset_index()

#batsman_out_count = batsman_out_count.rename(columns={'wkt_batsman_name': 'batsman1_name'})
# Merge with the total_runs_balls_by_batsman DataFrame
total_runs_by_batsman = pd.merge(total_runs_by_batsman, batsman_out_count, on='batsman1_name', how='left')

# Calculate average
total_runs_by_batsman['average'] = total_runs_by_batsman['batsman1_runs'] / total_runs_by_batsman['wkt_batsman_name']
```

Performance Evaluation

2.Bowler

```
total_wickets_by_bowler = wkts_per_bowler_per_match.groupby('bowler1_name').agg({
      'match_id': 'sum',
      'bowler1_overs': 'sum',
      'bowler1_wkts': 'sum',
      'bowler1_maidens':'sum',
      'bowler1_runs':'sum'
  }).reset_index()
total_wickets_by_bowler = total_wickets_by_bowler.sort_values(by='bowler1_wkts', ascending=False)

# Calculate Economy
total_wickets_by_bowler['economy'] = (total_wickets_by_bowler['bowler1_runs'] / total_wickets_by_bowler['bowler1_overs'])

#Rename columns
total_wickets_by_bowler = total_wickets_by_bowler.rename(columns={
    'bowler1_name': 'bowler_name',
    'match_id': 'matches',
    'bowler1_overs': 'overs',
    'bowler1_wkts': 'wickets',
    'bowler1_maidens': 'maidens',
    'bowler1_runs': 'runs_conceded'
})
total_wickets_by_bowler.head(5)
```

Top 10 Batsman in T20 World Cup 2022 by Runs

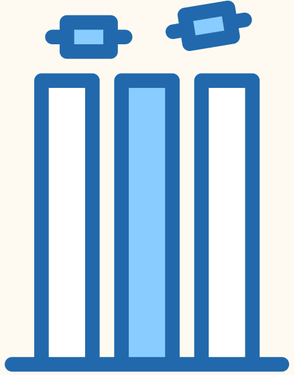
```
total_runs_by_batsman.head(10)
```

	batsman	matches	runs	balls_played	strike_rate	average	num_boundaries	not_out_count
0	Virat Kohli	6	296	217	136.405530	74.000000	33	2.0
1	Max O'Dowd	8	242	213	113.615023	26.888889	30	0.0
2	Suryakumar Yadav	6	233	125	186.400000	58.250000	34	2.0
3	Jos Buttler	6	225	156	144.230769	56.250000	31	2.0
4	Kusal Mendis	8	223	156	142.948718	31.857143	27	1.0
5	Sikandar Raza	8	219	147	148.979592	31.285714	27	1.0
6	Pathum Nissanka	7	214	196	109.183673	26.750000	21	0.0
7	Alex Hales	6	212	144	147.222222	42.400000	29	1.0
8	Lorcan Tucker	7	204	163	125.153374	51.000000	23	3.0
9	Glenn Phillips	5	201	126	159.523810	50.250000	27	1.0

Top 10 Bowler in T20 World Cup 2022

	bowler_name	matches	overs	wickets	maidens	runs_conceded	economy
126	Wanindu Hasaranga de Silva	8	30.4	14	1	198	6.513158
15	Bas de Leede	7	20.9	13	0	163	7.799043
109	Sam Curran	6	22.4	13	0	148	6.607143
22	Blessing Muzarabani	7	25.4	12	0	198	7.795276
100	Paul van Meekeren	8	31.0	11	0	198	6.387097
60	Josh Little	7	26.5	11	0	188	7.094340
111	Shadab Khan	7	26.0	11	0	165	6.346154
112	Shaheen Shah Afridi	7	25.1	11	0	155	6.175299
8	Anrich Nortje	5	17.3	11	0	94	5.433526
116	Sikandar Raza	8	24.0	10	0	156	6.500000

Top Performers Each Match

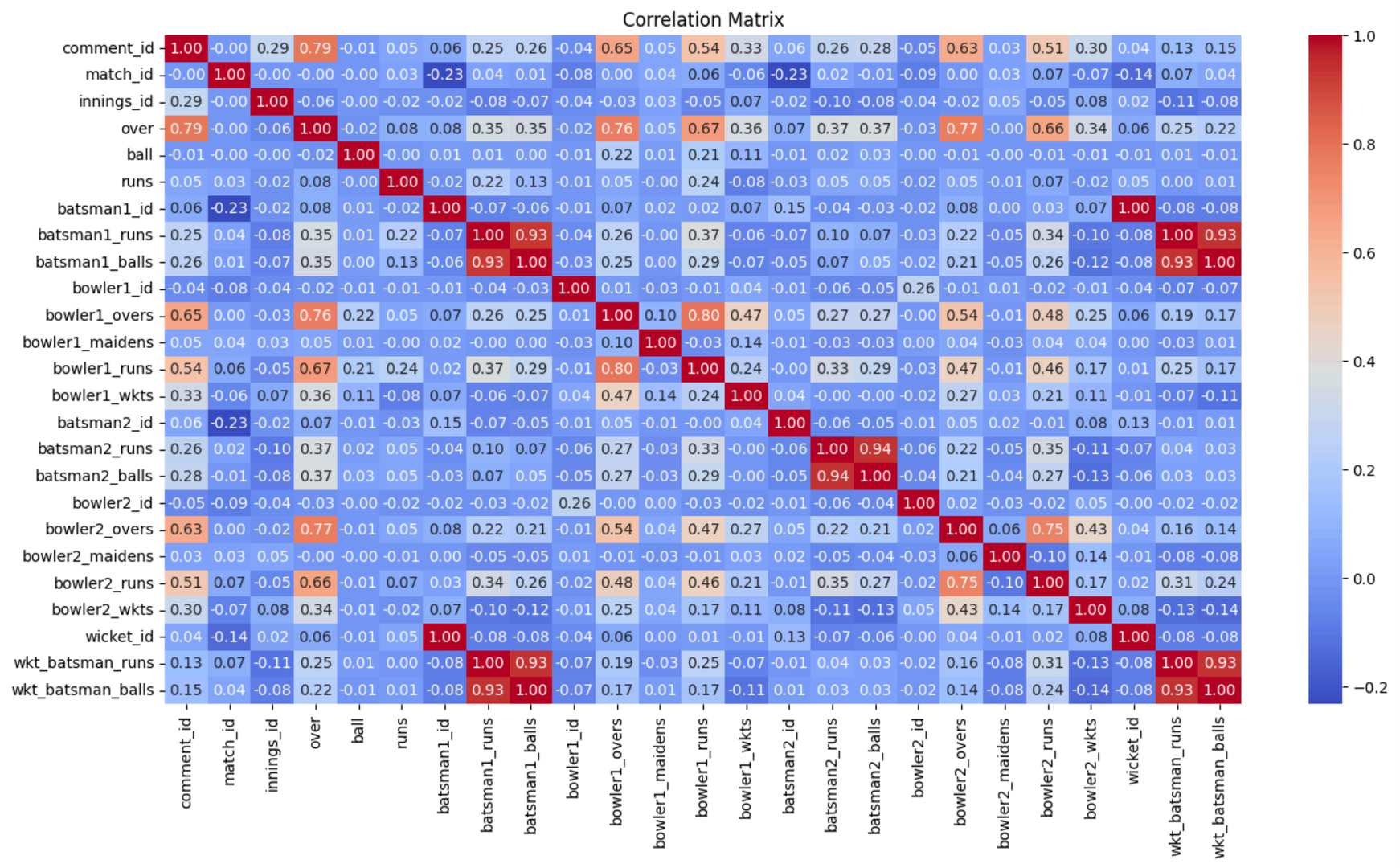


Top Performers per match

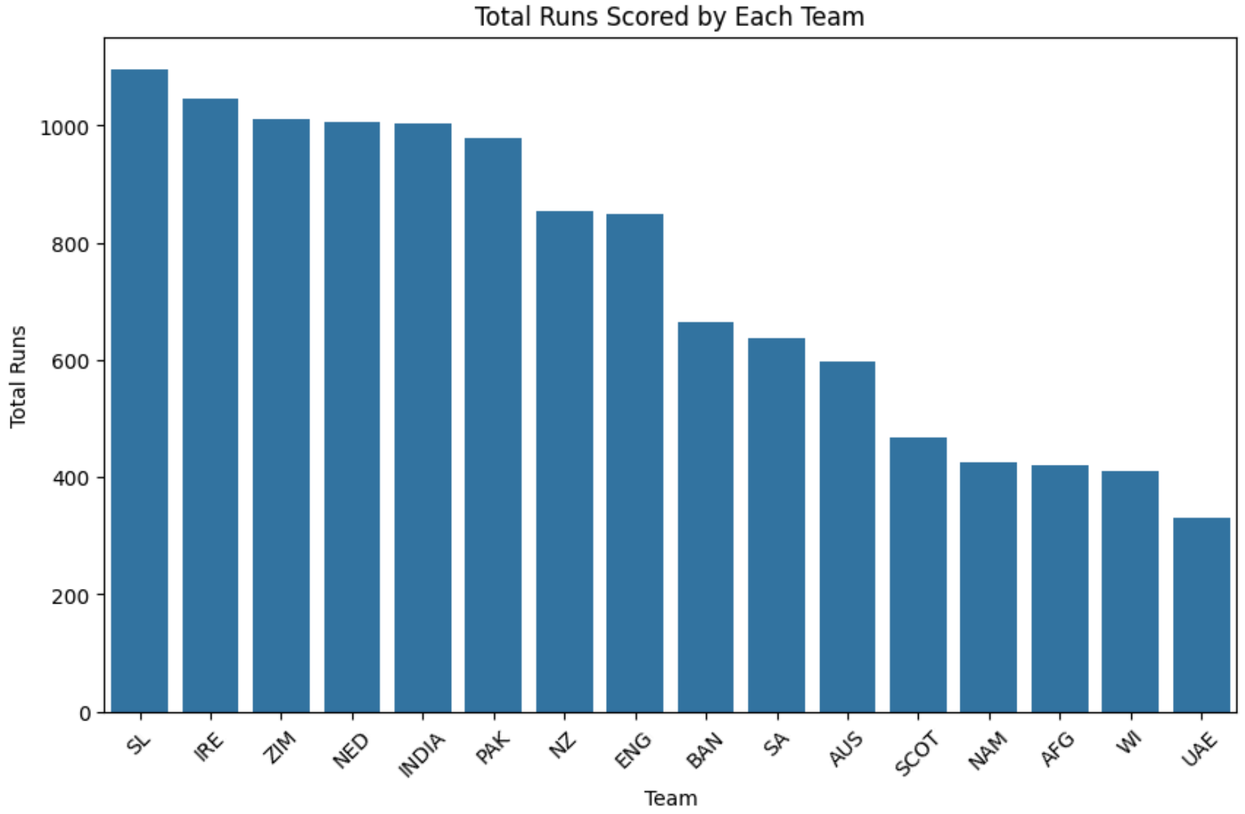
```
# For batsmen
top_batsman_per_match = df.groupby(['match_name', 'batsman1_name'])['batsman1_runs'].max().reset_index()
top_batsman_per_match = top_batsman_per_match.sort_values(by=['match_name', 'batsman1_runs'], ascending=[True, False])
top_batsman_per_match = top_batsman_per_match.groupby('match_name').first().reset_index()
top_batsman_per_match = top_batsman_per_match.rename(columns={'batsman1_name': 'top_batsman'})

# For bowlers
top_bowler_per_match = df.groupby(['match_name', 'bowler1_name'])['bowler1_wkts'].max().reset_index()
top_bowler_per_match = top_bowler_per_match.sort_values(by=['match_name', 'bowler1_wkts'], ascending=[True, False])
top_bowler_per_match = top_bowler_per_match.groupby('match_name').first().reset_index()
top_bowler_per_match = top_bowler_per_match.rename(columns={'bowler1_name': 'top_bowler'})

# Merge top batsman and top bowler DataFrames on 'match_name'
top_performers_per_match = pd.merge(top_batsman_per_match, top_bowler_per_match, on='match_name', how='outer')

#Rename columns
top_performers_per_match = top_performers_per_match.rename(columns={
    'batsman1_runs': 'runs',
    'bowler1_wkts': 'wickets'
})

top_performers_per_match.head(10)
```

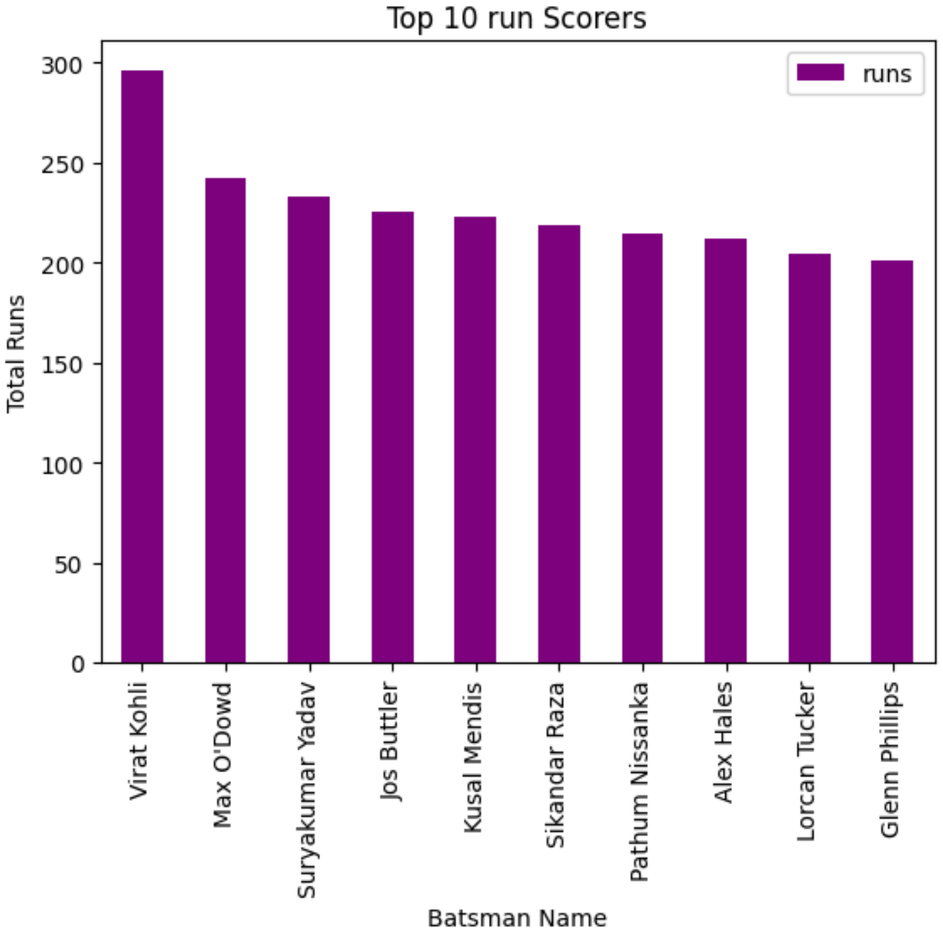
	match_name	top_batsman	runs	top_bowler	wickets
0	AFG v ENG	Ibrahim Zadran	32	Sam Curran	5
1	AFG v SL	Dhananjaya de Silva	66	Lahiru Kumara	2
2	AUS v AFG	Glenn Maxwell	54	Naveen-ul-Haq	3
3	AUS v IRE	Lorcan Tucker	71	Barry McCarthy	3
4	AUS v NZ	Devon Conway	92	Mitchell Santner	3
5	AUS v SL	Marcus Stoinis	59	Ashton Agar	1
6	BAN v INDIA	Virat Kohli	64	Hasan Mahmud	3
7	BAN v NED	Colin Ackermann	62	Taskin Ahmed	4
8	BAN v PAK	Najmul Hossain Shanto	54	Shaheen Shah Afridi	4
9	BAN v SA	Rilee Rossouw	109	Anrich Nortje	4

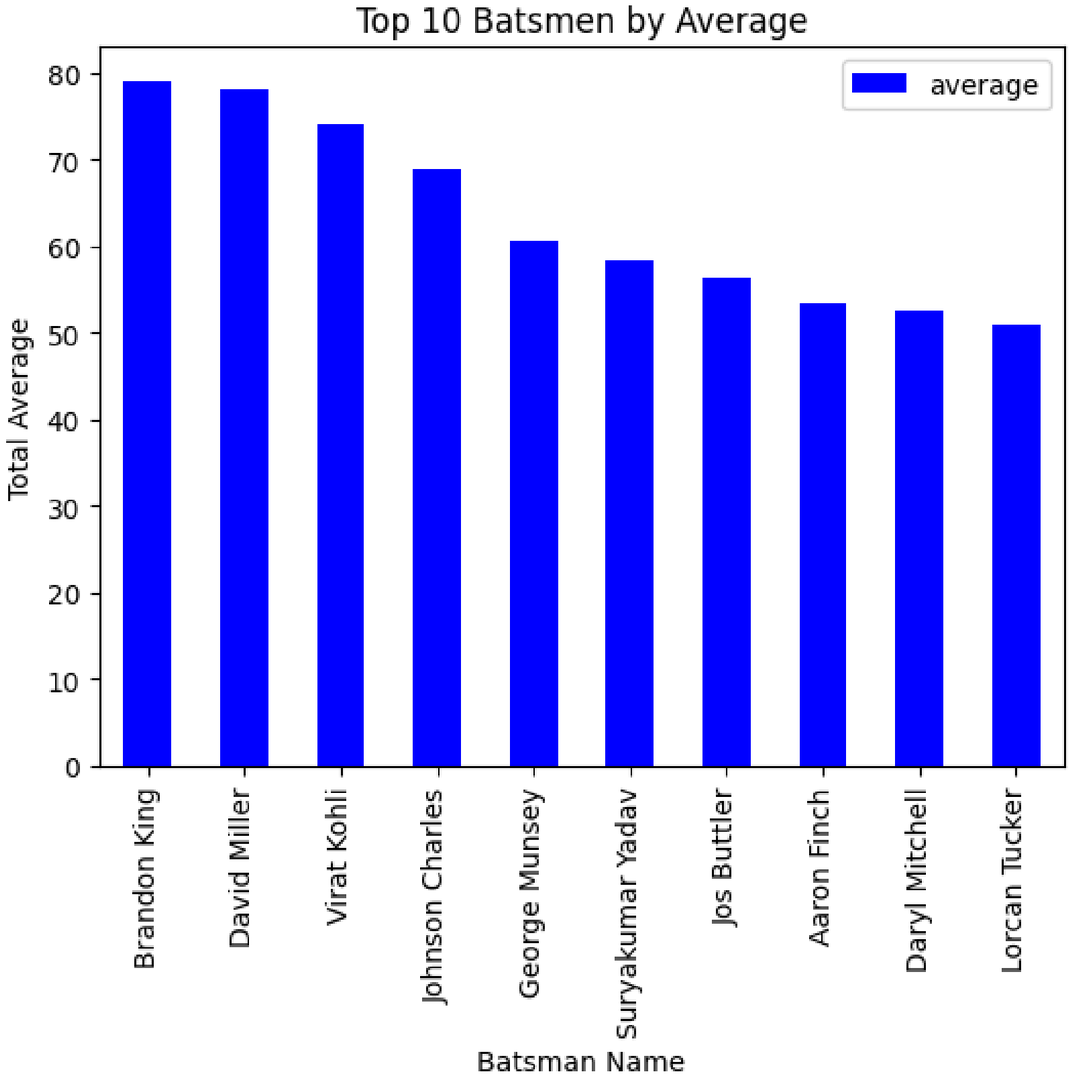
Statistical Insights

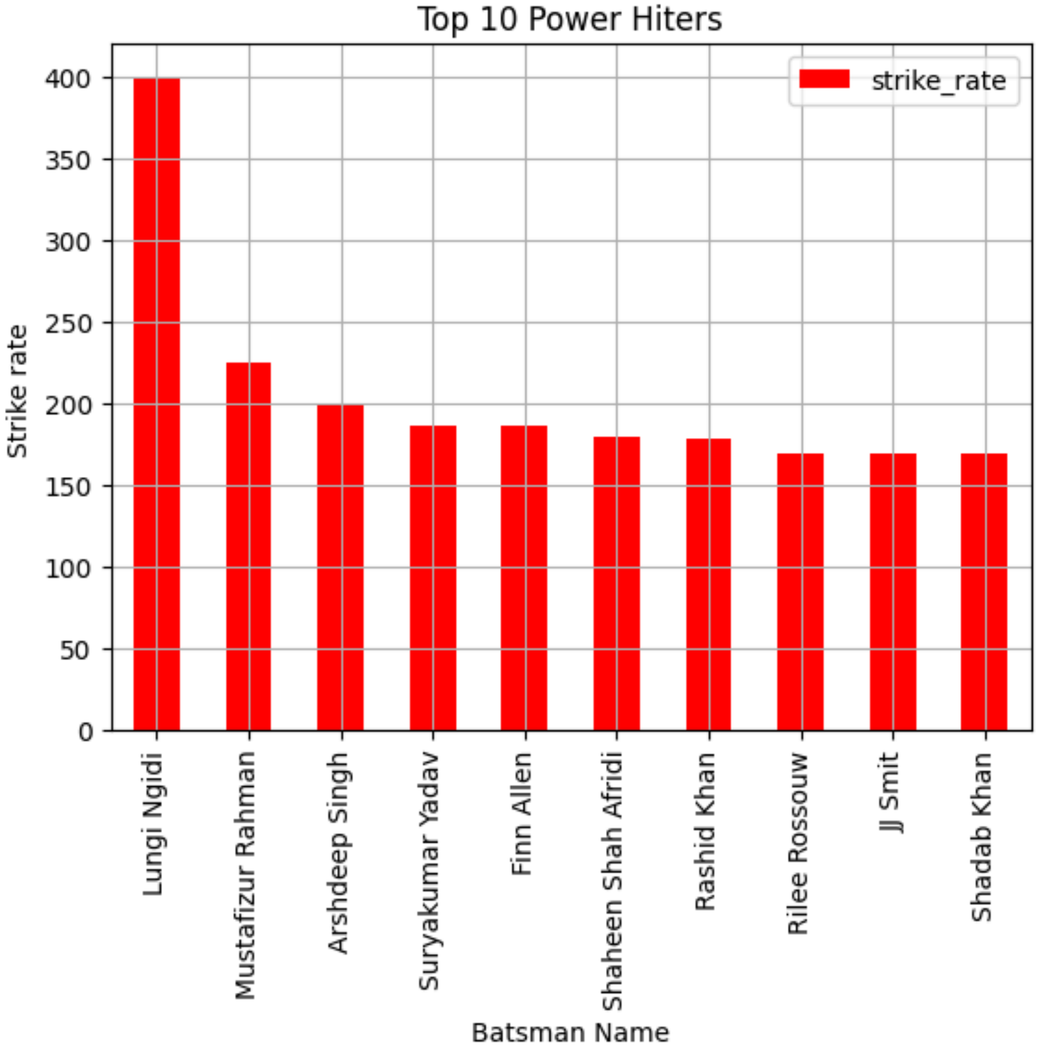


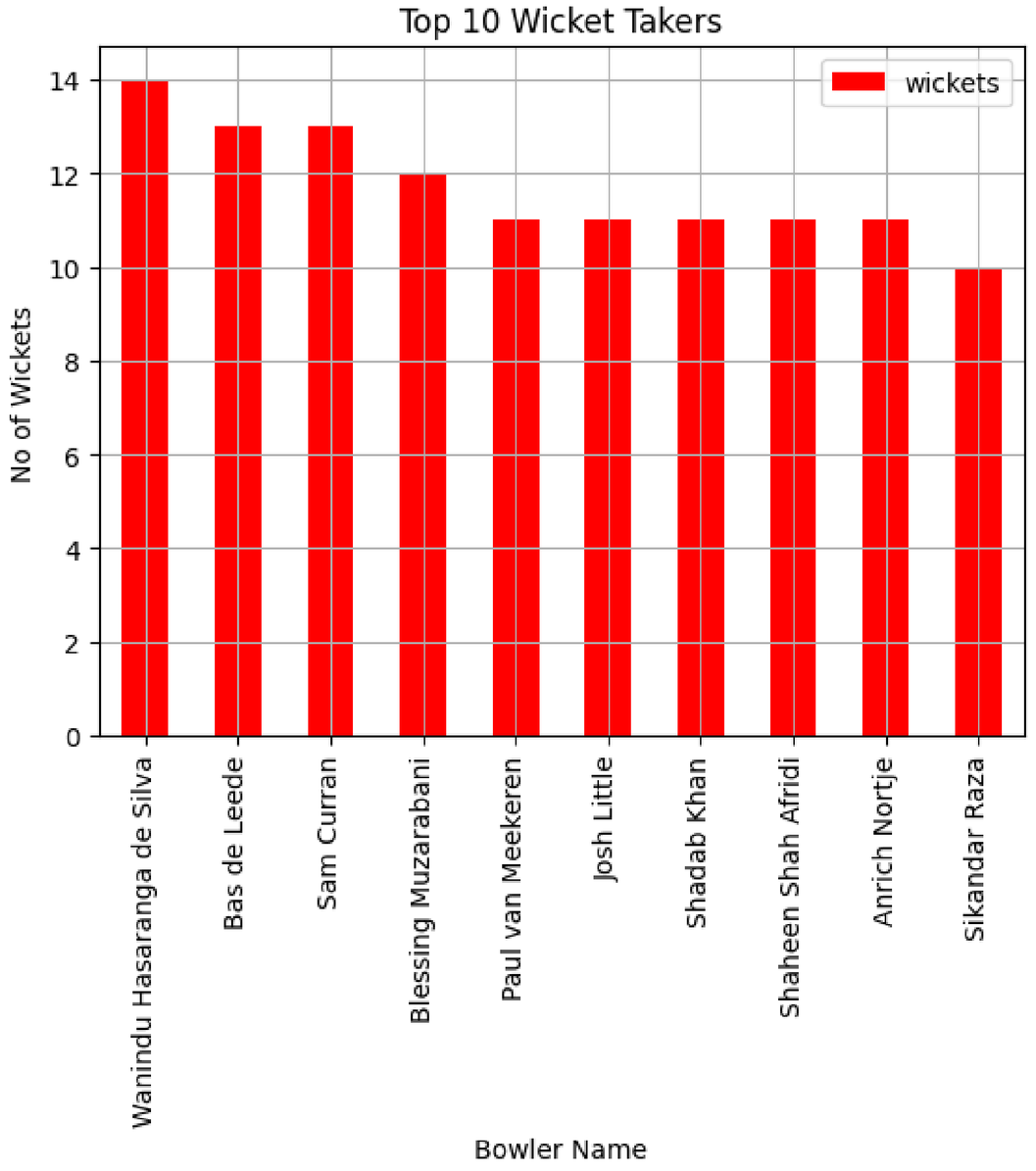
Visualizations

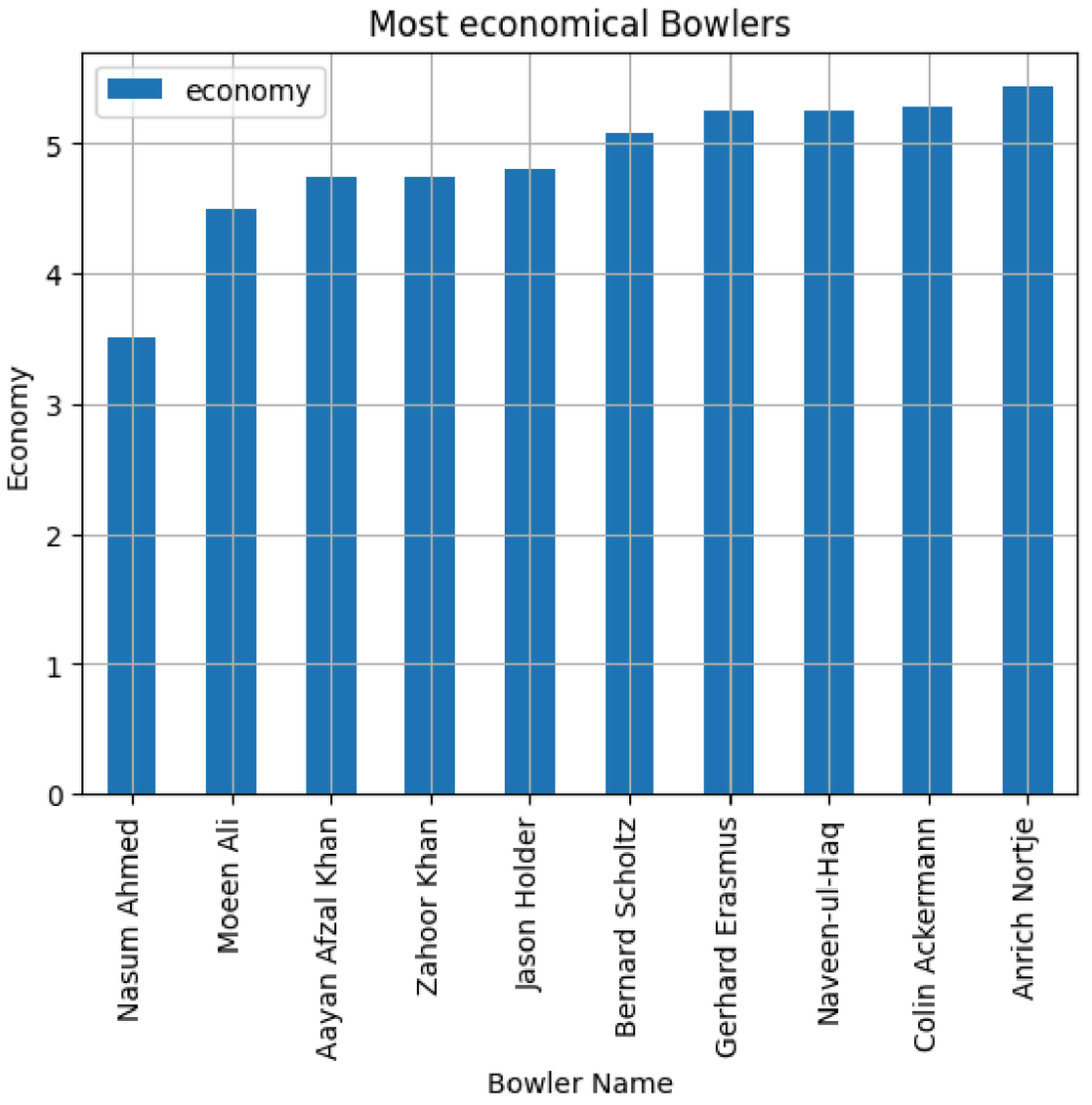





Top 10 Batsmen by Average
average
Total Average
0
10
20
30
40
50
60
70
80
Brandon King
David Miller
Virat Kohli
Johnson Charles
George Munsey
Suryakumar Yadav
Jos Buttler
Aaron Finch
Daryl Mitchell
Lorcan Tucker
Batsman Name


Top 10 Power Hiters
strike_rate
Strike rate
0
50
100
150
200
250
300
350
400
Lungi Ngidi
Mustafizur Rahman
Arshdeep Singh
Suryakumar Yadav
Finn Allen
Shaheen Shah Afridi
Rashid Khan
Rilee Rossouw
JJ Smit
Shadab Khan
Batsman Name


Top 10 Wicket Takers
wickets
No of Wickets
0
2
4
6
8
10
12
14
Wanindu Hasaranga de Silva
Bas de Leede
Sam Curran
Blessing Muzarabani
Paul van Meekeren
Josh Little
Shadab Khan
Shaheen Shah Afridi
Anrich Nortje
Sikandar Raza
Bowler Name


Most economical Bowlers
economy
Economy
0
1
2
3
4
5
Nasum Ahmed
Moeen Ali
Aayan Afzal Khan
Zahoor Khan
Jason Holder
Bernard Scholtz
Gerhard Erasmus
Naveen-ul-Haq
Colin Ackermann
Anrich Nortje
Bowler Name

Conclusion

In this analysis, we conducted a comprehensive exploration of the cricket match dataset, aiming to uncover insights into player and team performances, identify critical match events, and derive actionable recommendations. Our analysis revealed several key findings:

- **Player Performance**: Top batsmen and bowlers emerged based on their total runs scored and wickets taken, respectively. Additionally, the strike rate and average metrics provided insights into the effectiveness of players during matches.
- **Team Performance**: We observed variations in team performances, with some teams consistently scoring high totals across matches, while others struggled to maintain consistency. The determination of winning and losing teams shed light on the competitive nature of the matches and highlighted the significance of runs scored in determining match outcomes.
- **Event Inference:** Critical match events such as boundaries and wickets were found to have a significant impact on match dynamics. Strategic moments during matches, influenced by these events, often played a decisive role in shaping match outcomes.

Our analysis underscores the importance of a balanced approach to both batting and bowling aspects of the game. Teams that effectively capitalized on scoring opportunities while maintaining discipline in bowling emerged as frontrunners in match results. Furthermore, the ability to adapt to match situations and capitalize on key moments proved to be crucial for success.


Mentorness ML Internship
THANK YOU
SAI VARMA PALOJI
upstox
NISSAN