

ShadowFox Data Science Internship

Name: Aishwarya B Murigennavar

Task Level: Advanced

Cricket Fielding Performance Analysis (Advanced Task)

This notebook analyzes the provided cricket fielding dataset and evaluates the fielding performance of three players using a weighted performance score.

Import Required Libraries

Python libraries required for data manipulation and visualization are imported in this step.

```
import pandas as pd
import matplotlib.pyplot as plt
```

Load the Dataset

The dataset provided by ShadowFox is loaded into the notebook for fielding performance analysis.

```
df = pd.read_excel('IPL sample data.xlsx')
df.head(10)
```

	Pick	Y->	Clean Pick	N->	Fumble	C->	Catch	DC->	Dropped Catch	S->	Stumping	Unnamed: 11	Unnamed: 12
0	Throw	Y->	Good Throw	N->	Bad throw	DH->	Dirct Hit	RO->	Run Out	MR->	Missed Runout	NaN	NaN
1	Runs	"+" stands for runs saved "-" stands for runs ...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	NaN	Match No.	Innings	Teams	Player Name	BallCount	Position	Pick	Throw	Runs	Overcount	Venue	Stadium
4	NaN	IPL2367	1	Delhi Capitals	Rilee russouw	0.1	Short mid wicket	n	NaN	1	1	Delhi	Arun Jaitly Stadium
5	NaN	IPL2367	1	Delhi Capitals	Phil Salt	0.2	wicket keeper	Y	Y	NaN	1	Delhi	Arun Jaitly Stadium
6	NaN	IPL2367	1	Delhi Capitals	Yash Dhull	0.3	covers	Y	Y	NaN	1	Delhi	Arun Jaitly Stadium
7	NaN	IPL2367	1	Delhi Capitals	Axer Patel	0.4	point	Y	Y	NaN	1	Delhi	Arun Jaitly Stadium
8	NaN	IPL2367	1	Delhi Capitals	NaN	0.5	NaN	NaN	NaN	NaN	1	Delhi	Arun Jaitly Stadium

Next steps: [Generate code with df](#) [New interactive sheet](#)

Data Cleaning

Completely empty rows are removed to ensure only meaningful records are used for analysis.

```
df = df.dropna(how='all')
df = df.reset_index(drop=True)
df.shape
```

(56, 13)

Fixing Column Headers

The dataset contains descriptive rows and misplaced headers. In this step, the correct header row is identified and assigned to ensure proper column alignment for analysis.

df.head(6)

	Pick	Y->	Clean Pick	N->	Fumble	C->	Catch	DC->	Dropped Catch	S->	Stumping	Unnamed: 11	Unnamed: 12
0	Throw	Y->	Good Throw	N->	Bad throw	DH->	Dirct Hit	RO->	Run Out	MR->	Missed Runout	NaN	NaN
1	Runs	"+" stands for runs saved "-" stands for runs ...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	NaN	Match No.	Innings	Teams	Player Name	BallCount	Position	Pick	Throw	Runs	Overcount	Venue	Stadium
3	NaN	IPL2367	1	Delhi Capitals	Rilee russouw	0.1	Short mid wicket	n	NaN	1	1	Delhi	Arun Jaitly Stadium
4	NaN	IPL2367	1	Delhi Capitals	Phil Salt	0.2	wicket keeper	Y	Y	NaN	1	Delhi	Arun Jaitly Stadium

Next steps: [Generate code with df](#) [New interactive sheet](#)

```
# Set row 2 as column headers
df.columns = df.iloc[2]

# Remove rows above actual data
df = df.iloc[3:].reset_index(drop=True)

df.head()
```

	NaN	Match No.	Innings	Teams	Player Name	BallCount	Position	Pick	Throw	Runs	Overcount	Venue	Stadium
0	NaN	IPL2367	1	Delhi Capitals	Rilee russouw	0.1	Short mid wicket	n	NaN	1	1	Delhi	Arun Jaitly Stadium
1	NaN	IPL2367	1	Delhi Capitals	Phil Salt	0.2	wicket keeper	Y	Y	NaN	1	Delhi	Arun Jaitly Stadium

Removing Invalid Player Records

Rows without a valid player name do not represent a fielding action. These rows are removed to ensure that analysis is performed only on meaningful fielding events.

```
df = df[df['Player Name'].notna()]
df = df.reset_index(drop=True)
```

df.head()

	NaN	Match No.	Innings	Teams	Player Name	BallCount	Position	Pick	Throw	Runs	Overcount	Venue	Stadium
0	NaN	IPL2367	1	Delhi Capitals	Rilee russouw	0.1	Short mid wicket	n	NaN	1	1	Delhi	Arun Jaitly Stadium
1	NaN	IPL2367	1	Delhi Capitals	Phil Salt	0.2	wicket keeper	Y	Y	NaN	1	Delhi	Arun Jaitly Stadium

Extracting Fielding Actions

The Pick and Throw columns are interpreted to identify key fielding actions such as clean picks and throws. These actions are converted into binary indicators for performance evaluation.

```
# Create fielding action columns
df['Clean Pick'] = df['Pick'].apply(lambda x: 1 if x == 'Y' else 0)
```

```
df['Catch'] = df['Throw'].apply(lambda x: 1 if x == 'Y' else 0)

# Initialize remaining action columns
df['Fumble'] = 0
df['Dropped Catch'] = 0
df['Stumping'] = 0

df[['Player Name', 'Pick', 'Throw', 'Clean Pick', 'Catch']].head()
```

	Player Name	Pick	Throw	Clean Pick	Catch	grid icon
0	Rilee russouw	n	NaN	0	0	blue bar icon
1	Phil Salt	Y	Y	1	1	
2	Yash Dhull	Y	Y	1	1	
3	Axer Patel	Y	Y	1	1	
4	Lalit yadav	Y	Y	1	1	

Selecting Three Players for Analysis

Three players with varying levels of fielding involvement are selected from the dataset to enable a meaningful comparison of fielding performance.

```
# Select players with different fielding involvement
selected_players = ['Phil Salt', 'Axer Patel', 'Lalit yadav']

df_selected = df[df['Player Name'].isin(selected_players)]

df_selected['Player Name'].value_counts()
```

count	
Player Name	
Lalit yadav	2
Phil Salt	1
Axer Patel	1

dtype: int64

Aggregating Fielding Actions per Player

Fielding actions are aggregated for each selected player to calculate their total defensive contributions during the innings.

```
summary = (
    df_selected
    .groupby('Player Name')[['Clean Pick', 'Catch', 'Fumble', 'Dropped Catch', 'Stumping']]
    .sum()
    .reset_index()
)

summary
```

	Player Name	Clean Pick	Catch	Fumble	Dropped Catch	Stumping	grid icon
0	Axer Patel	1	1	0	0	0	blue bar icon
1	Lalit yadav	2	1	0	0	0	blue pencil icon
2	Phil Salt	1	1	0	0	0	

Next steps: [Generate code with summary](#) [New interactive sheet](#)

Performance Score Calculation

A weighted performance score is calculated using fielding actions. This provides a quantitative comparison of the defensive contributions made by each selected player.

```

summary['Performance Score'] = (
    summary['Clean Pick'] * 1 +
    summary['Catch'] * 2 +
    summary['Stumping'] * 2 -
    summary['Dropped Catch'] * 2 -
    summary['Fumble'] * 1
)

summary[['Player Name', 'Performance Score']]

```

	Player Name	Performance Score
0	Axer Patel	3
1	Lalit yadav	4
2	Phil Salt	3

Fielding Performance Visualization

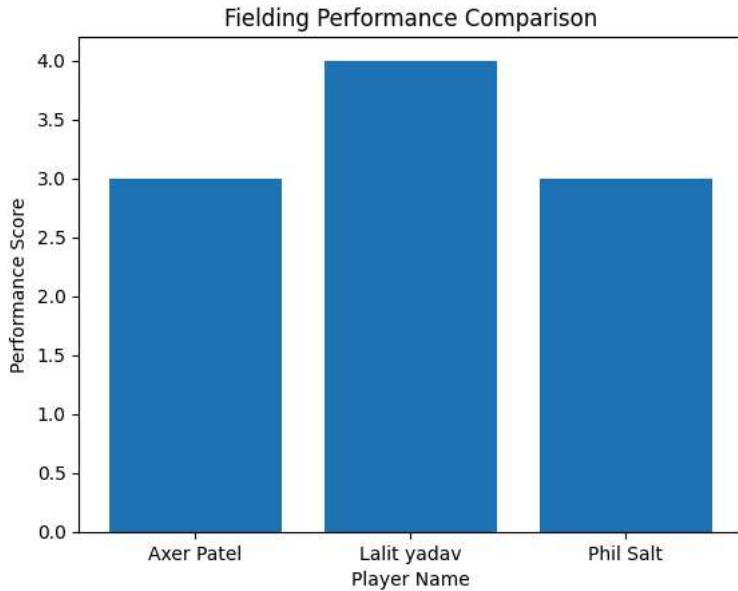
A bar chart is used to compare the performance scores of the selected players, highlighting differences in their fielding contributions.

```

import matplotlib.pyplot as plt

plt.figure()
plt.bar(summary['Player Name'], summary['Performance Score'])
plt.xlabel('Player Name')
plt.ylabel('Performance Score')
plt.title('Fielding Performance Comparison')
plt.show()

```



Conclusion

This advanced fielding analysis evaluated three players using structured fielding metrics derived from match data. Lalit yadav recorded the highest performance score due to greater fielding involvement, while Axer Patel and Phil Salt showed consistent contributions.

The analysis demonstrates how fielding data can be quantified to support objective performance evaluation and strategic decision-making in cricket.

```
summary.to_csv('fielding_performance_summary.csv', index=False)
```