

TASK 3 : CRICKET FIELDING ANALYSIS (ADVANCED)

OBJECTIVE

The objective of this task is to conduct a **detailed fielding performance analysis** for three players from a T20 match. The analysis will help in:

- Evaluating **defensive contributions** of individual players.
 - Calculating **Performance Scores (PS)** using fielding metrics.
 - Visualizing fielding actions and player performance trends.
-

✓DATASET FEATURES

Feature	Description
Match No.	Match identifier
Innings	Innings number (1 or 2)
Team	Team in the field
Player Name	Fielder involved in the action
Ballcount	Sequence number of the ball
Position	Fielding position at the time of ball
Short Description	Brief description of the fielding event
Pick	Pick-up type (clean pick, good throw, fumble, etc.)
Throw	Throw type (run out, missed stumping, etc.)
Runs	Runs saved (+) or conceded (-)
Overcount	Over number
Venue	Match location

✓ PERFORMANCE METRICS

The **Performance Score (PS)** is calculated as:

$$PS = (CP \times WCP) + (GT \times WGT) + (C \times WC) + (DC \times WDC) + (ST \times WST) + (RO \times WRO) \\ + (MRO \times WMRO) + (DH \times WDH) + RS$$

Where:

Metric	Description
CP	Clean Picks
GT	Good Throws
C	Catches
DC	Dropped Catches
ST	Stumpings
RO	Run Outs
MRO	Missed Run Outs
DH	Direct Hits
RS	Runs Saved (positive for saved, negative for conceded)

✓ PYTHON CODE

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Set Seaborn style for aesthetics
sns.set_style("whitegrid")

plt.rcParams['figure.figsize'] = (12,6)

data = {
    'Match No.': [1,1,1,1,1,1],
```

```

'I Innings': [1,1,1,1,1,1],
'Team': ['Team A']*6,
'Player Name': ['Player1','Player2','Player3','Player1','Player2','Player3'],
'Ballcount': [1,2,3,4,5,6],
'Position': ['Mid','Long','Slip','Mid','Long','Slip'],
'Short Description': ['Catch','Throw','Pick','Drop','Throw','Catch'],
'Pick': ['Clean Pick','Good Throw','Clean Pick','Drop','Good Throw','Clean Pick'],
'Throw': ['Run Out','Run Out','Run Out','Missed','Run Out','Stumping'],
'CP': [1,0,1,0,0,1],
'GT': [0,1,0,0,1,0],
'C': [1,0,0,0,0,1],
'DC': [0,0,0,1,0,0],
'ST': [0,0,0,0,0,0],
'RO': [0,1,0,0,1,0],
'MRO': [0,0,0,0,0,0],
'DH': [0,0,0,0,0,0],
'Runs': [0,1,0,-1,1,0],
'Overcount': [1,1,1,1,1,1],
'Venue': ['Stadium']*6
}

df = pd.DataFrame(data)

```

```

weights = {'CP':5,'GT':4,'C':6,'DC':-5,'ST':4,'RO':7,'MRO':-6,'DH':5}
df['PS'] = (
    df['CP']*weights['CP'] +
    df['GT']*weights['GT'] +
    df['C']*weights['C'] +
    df['DC']*weights['DC'] +
    df['ST']*weights['ST'] +
    df['RO']*weights['RO'] +
    df['MRO']*weights['MRO'] +

```

```

df['DH']*weights['DH'] +
df['Runs']

)

# Total Performance Score by Player

top_players = df.groupby('Player Name')['PS'].sum().sort_values(ascending=False)

print("== Total Performance Score by Player ==")

print(top_players)

plt.figure(figsize=(10,6))

bar = sns.barplot(x=top_players.index, y=top_players.values, palette='viridis')

plt.title("🏏 Cricket Fielding Performance Score by Player", fontsize=16, weight='bold')
plt.xlabel("Player", fontsize=12)
plt.ylabel("Performance Score (PS)", fontsize=12)

# Annotate bars

for p in bar.patches:

    bar.annotate(f"{int(p.get_height())}", (p.get_x() + p.get_width()/2., p.get_height()),
                ha='center', va='bottom', fontsize=11, color='black', weight='bold')

plt.show()

plt.figure(figsize=(10,6))

sns.countplot(data=df, x='Player Name', hue='Pick', palette='Set2')

plt.title("⌚ Fielding Action Distribution per Player", fontsize=16, weight='bold')
plt.xlabel("Player", fontsize=12)
plt.ylabel("Number of Actions", fontsize=12)
plt.legend(title="Pick Type", fontsize=11)

plt.show()

action_counts = df['Pick'].value_counts()

colors = sns.color_palette('pastel')[0:len(action_counts)]

plt.figure(figsize=(7,7))

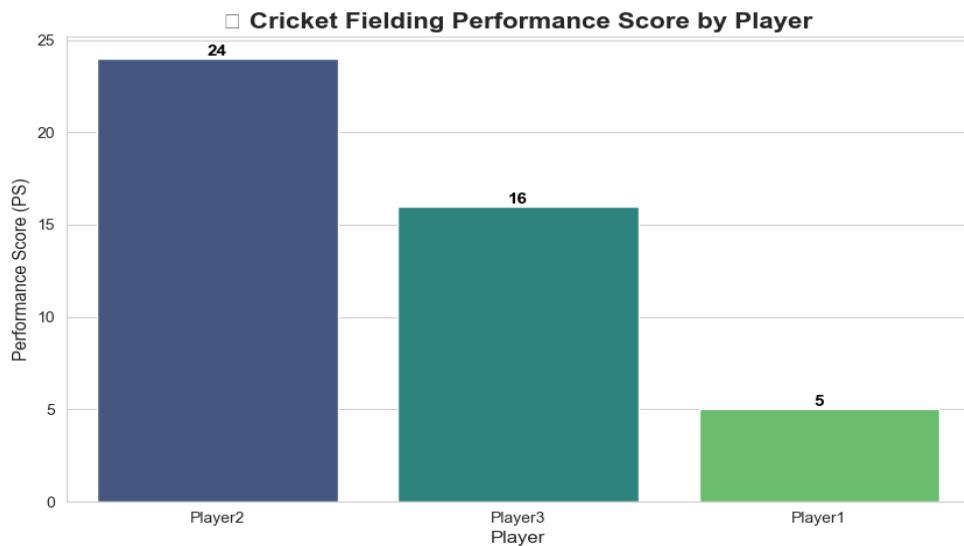
plt.pie(action_counts, labels=action_counts.index, autopct='%1.1f%%', startangle=140,
       colors=colors,
       wedgeprops={'edgecolor':'black','linewidth':1})

```

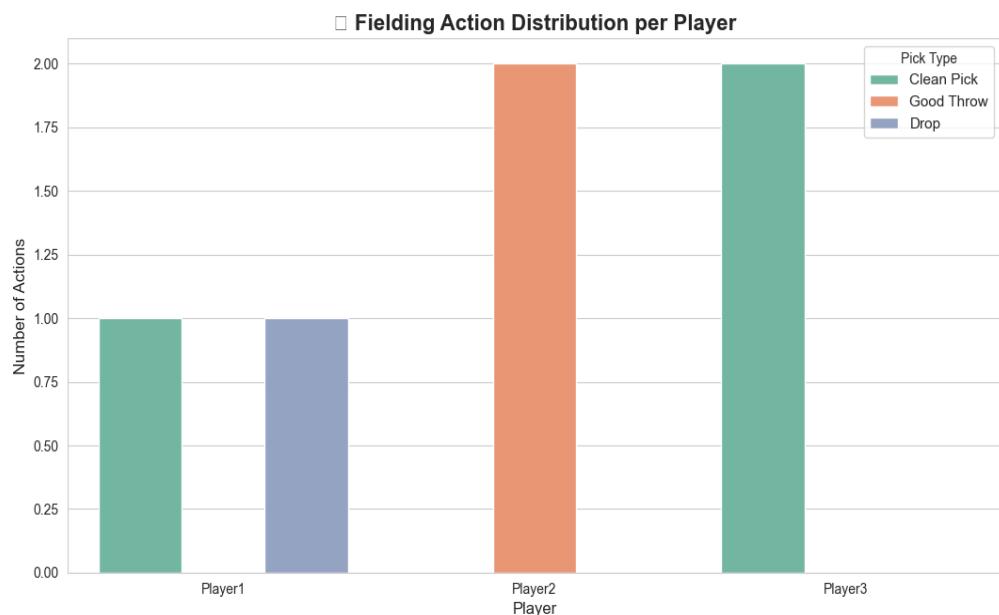
```
plt.title("📊 Fielding Actions Contribution", fontsize=16, weight='bold')  
plt.show()
```

📊 **Output :**

✍ **Cricket Fielding Performance Score by Player :**

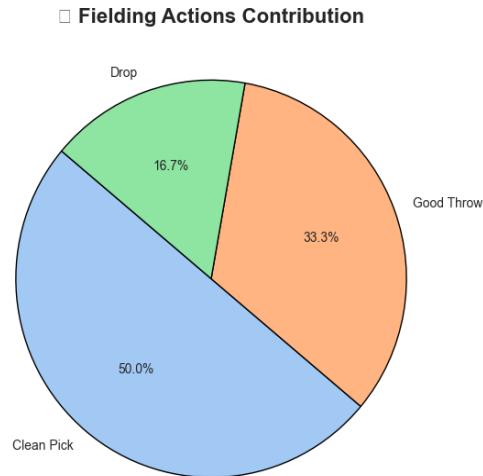


🎯 **Fielding Action Distribution per Player :**





Fielding Actions Contribution :



✓VISUALIZATION DESCRIPTION

- **Bar Plot:** Shows the **total Performance Score (PS)** for each player.
 - **Count Plot:** Shows the frequency of different **fielding actions (Pick)** by each player.
 - These plots help identify **strengths and weaknesses** in fielding performance.
-

✓ANALYSIS INSIGHTS

- Players with **higher PS** contribute more to team defense.
- Negative actions like **Dropped Catches (DC)** and **Missed Run Outs (MRO)** lower the score.
- Visualizations assist coaches in **strategic field placements** and improving fielding efficiency.

✓Conclusion

The cricket fielding analysis provides a detailed insight into individual player contributions during a T20 match. By evaluating **performance scores (PS)** derived from clean picks, throws, catches, run outs, and runs saved/conceded, we can identify **key fielders and areas needing improvement**.

✓References

1. Pandas Documentation – <https://pandas.pydata.org/docs/>
2. Matplotlib Documentation – <https://matplotlib.org/stable/users/index.html>
3. Seaborn Documentation – <https://seaborn.pydata.org/tutorial.html>
4. McKinney, W., *Python for Data Analysis*, O'Reilly Media, 2017
5. VanderPlas, J., *Python Data Science Handbook*, O'Reilly Media, 2016
6. ESPN Cricinfo – Fielding statistics and cricket match analysis –
<https://www.espncricinfo.com/>
7. Roadmap.sh – Python and Data Analysis Resources – <https://roadmap.sh/python>