Cricket Fielding Performance Analysis

Of course. Based on the provided sample data for Match No. IPL2367, Innings 1.0, Team: Delhi Capitals, here is a detailed fielding performance analysis for the three selected players: Yash Dhull, Axar Patel, and Kuldeep Yadav.

Fielding Performance Analysis Report

Match: IPL2367 (Delhi Capitals vs. [Opponent]) Innings:1 (Delhi Capitals Fielding) Venue:Arun Jaitly Stadium, Delhi

1. Performance Summary Table

Player Name Clean Picks (CP) Good Throws (GT) Catches (C) Dropped Catches (DC) Run Outs (RO) Missed Run Outs (MRO) Direct Hits (DH) Runs Saved (RS) Performance Score (PS)

Yash Dhull 3 1 2 0 0 1 0 +3 11

Axar Patel 2 3 1 0 0 0 0 0 11

Kuldeep Yadav 3 0 1 1 0 0 1 +4 9

2. Individual Player Analysis

a) Yash Dhull (Performance Score: 11)

· Strengths: Yash Dhull was the most consistent and impactful fielder in the ring. His 3 clean picks and 2 catches demonstrate exceptional hand-eye coordination and safe hands under pressure. Taking two catches is a high-value contribution.

· Area for Improvement: The data shows 1 missed run out (MRO). This suggests that while his gathering and throwing were generally good (1 Good Throw), there was one instance where a run-out opportunity was not capitalized on, possibly due to a slight inaccuracy or delay in the throw.

· Overall Impact: His positive run saving (+3) and high-value actions (catches) make him the joint-top performer. He was a reliable asset in the inner ring.

b) Axar Patel (Performance Score: 11)

· Strengths: Axar Patel was the engine of the fielding unit in terms of volume and efficiency. A remarkable 3 good throws indicate his arm was a constant weapon, putting pressure on the batsmen and preventing extra runs. His all-round contribution (2 CP, 1 C) was very balanced.

· Area for Improvement: There are no significant negatives recorded. His challenge would be to maintain this high level of consistency throughout the tournament. Converting one of his good throws into a direct hit or run-out would be the next level.

· Overall Impact: Joint-top scorer. His performance is a model of high-efficiency fielding: consistent, reliable, and constantly pressuring the opposition without making errors.

c) Kuldeep Yadav (Performance Score: 9)

· Strengths: Kuldeep was very active, evidenced by a high number of 3 clean picks. He also effected a direct hit (DH), which is a game-changing play that often leads to a run-out or creates immense pressure. He also saved 4 runs, showing his effort and efficiency in the field.

· Area for Improvement: The major blemish on his performance was a dropped catch (DC). Dropping a chance is the most costly fielding error, as it gives a batsman a second life. This negative outweighs the positive of his direct hit in the points system.

· Overall Impact: A mixed bag. He made some of the most impactful plays (DH, run saves) but also committed the most significant error (DC). His score reflects this duality—good, but could have been excellent with safer hands.

3. Comparative Analysis & Team Impact

· Top Performer: Yash Dhull and Axar Patel are tied with a PS of 11, but they achieved it differently. Dhull was impactful with dismissals (catches), while Axar was consistently pressuring with his throwing.

· Most Aggressive Fielder: Kuldeep Yadav attempted the most high-risk, high-reward plays. He got a direct hit but also dropped a catch.

· Team Contribution: The data shows a strong fielding performance from Delhi Capitals. The fact that all three analyzed players have positive scores indicates a good overall team effort. The dropped catch and missed run-out are key areas the team must address to convert good fielding into excellent, match-winning fielding.

4. Recommendations for Strategic Fielding Placements

1. Yash Dhull: Should be placed in key catching positions like slip, short extra cover, or short mid-wicket where his safe hands can be best utilized. His minor issue with run-outs suggests he might be better suited to positions where quick throws to the stumps are less frequent than quick throws to the keeper.

2. Axar Patel: Is ideal for point or cover region. These positions require a sharp thrower to prevent singles and put pressure on batsmen trying to steal runs. His 3 good throws prove he is perfect for this role.

3. Kuldeep Yadav: His mix of brilliance and error suggests he should be placed in the deep (long-on/long-off) or at mid-wicket. In the deep, his powerful arm can create run-out opportunities from the boundary, and the consequences of a fumble are often just an extra run instead of a dropped catch. This mitigates his risk while leveraging his throwing strength.

This analysis confirms that fielding is not just about avoiding mistakes but about making positive contributions. All three players contributed positively to the team's defensive effort, with Axar and Yash being the most valuable on the day.

Cricket Fielding Performance Analysis - Python Code

import pandas as pd  
import numpy as np  
  
# Define the weights for each fielding action as per the provided formula  
WEIGHTS = {  
 'CP': 1, # Clean Picks  
 'GT': 1, # Good Throws  
 'C': 3, # Catches  
 'DC': -3, # Dropped Catches  
 'ST': 3, # Stumpings  
 'RO': 3, # Run Outs  
 'MRO': -2, # Missed Run Outs  
 'DH': 2, # Direct Hits  
 'RS': 1 # Runs Saved (added as is)  
}  
  
def calculate\_performance\_score(player\_data):  
 """  
 Calculate the performance score for a player based on the formula:  
 PS = (CP×WCP) + (GT×WGT) + (C×WC) + (DC×WDC) + (ST×WST) +  
 (RO×WRO) + (MRO×WMRO) + (DH×WDH) + RS  
 """  
 cp = player\_data.get('Clean Picks (CP)', 0)  
 gt = player\_data.get('Good Throws (GT)', 0)  
 c = player\_data.get('Catches (C)', 0)  
 dc = player\_data.get('Dropped Catches (DC)', 0)  
 st = player\_data.get('Stumpings (S)', 0)  
 ro = player\_data.get('Run Outs (RO)', 0)  
 mro = player\_data.get('Missed Run Outs (MR)', 0)  
 dh = player\_data.get('Direct Hits (DH)', 0)  
 rs = player\_data.get('Runs Saved (RS)', 0)  
  
 ps = (cp \* WEIGHTS['CP']) + (gt \* WEIGHTS['GT']) + (c \* WEIGHTS['C']) + \   
 (dc \* WEIGHTS['DC']) + (st \* WEIGHTS['ST']) + (ro \* WEIGHTS['RO']) + \   
 (mro \* WEIGHTS['MRO']) + (dh \* WEIGHTS['DH']) + (rs \* WEIGHTS['RS'])   
   
 return ps  
  
def analyze\_fielding\_performance():  
 # Create a DataFrame with the sample data  
 data = {  
 'Player Name': ['Rilee russouw', 'Phil Salt', 'Yash Dhull', 'Axer Patel',  
 'Lalit yadav', 'Aman Khan', 'Kuldeep yadav'],  
 'Clean Picks (CP)': [2, 1, 3, 2, 1, 4, 3],  
 'Good Throws (GT)': [1, 2, 1, 3, 2, 1, 0],  
 'Catches (C)': [1, 0, 2, 1, 1, 0, 1],  
 'Dropped Catches (DC)': [0, 1, 0, 0, 0, 0, 1],  
 'Stumpings (S)': [0, 0, 0, 1, 0, 0, 0],  
 'Run Outs (RO)': [0, 1, 0, 0, 0, 1, 0],  
 'Missed Run Outs (MR)': [0, 0, 1, 0, 0, 0, 0],  
 'Direct Hits (DH)': [1, 0, 0, 0, 1, 0, 1],  
 'Runs Saved (RS)': [2, -1, 3, 0, -2, 1, 4],  
 'Performance Score (PS)': [10, 2, 11, 11, 6, 9, 9]  
 }  
  
 df = pd.DataFrame(data)   
   
 # Filter for the three players of interest   
 players\_of\_interest = ['Axer Patel', 'Kuldeep yadav', 'Yash Dhull']   
 filtered\_df = df[df['Player Name'].isin(players\_of\_interest)].copy()   
   
 # Calculate performance score using our function   
 filtered\_df['Calculated PS'] = filtered\_df.apply(calculate\_performance\_score, axis=1)   
   
 return filtered\_df  
  
def generate\_detailed\_analysis(df):  
 """  
 Generate a detailed analysis of the fielding performance for each player  
 """  
 analysis = {}  
  
 for \_, row in df.iterrows():   
 player\_name = row['Player Name']   
 analysis[player\_name] = {   
 'Clean Picks': row['Clean Picks (CP)'],   
 'Good Throws': row['Good Throws (GT)'],   
 'Catches': row['Catches (C)'],   
 'Dropped Catches': row['Dropped Catches (DC)'],   
 'Stumpings': row['Stumpings (S)'],   
 'Run Outs': row['Run Outs (RO)'],   
 'Missed Run Outs': row['Missed Run Outs (MR)'],   
 'Direct Hits': row['Direct Hits (DH)'],   
 'Runs Saved': row['Runs Saved (RS)'],   
 'Performance Score': row['Performance Score (PS)'],   
 'Strengths': [],   
 'Areas for Improvement': []   
 }   
   
 # Identify strengths   
 if row['Clean Picks (CP)'] >= 2:   
 analysis[player\_name]['Strengths'].append('Clean fielding pick-ups')   
 if row['Good Throws (GT)'] >= 2:   
 analysis[player\_name]['Strengths'].append('Accurate throwing')   
 if row['Catches (C)'] >= 1:   
 analysis[player\_name]['Strengths'].append('Safe hands in catching')   
 if row['Direct Hits (DH)'] >= 1:   
 analysis[player\_name]['Strengths'].append('Direct hit capability')   
 if row['Runs Saved (RS)'] > 0:   
 analysis[player\_name]['Strengths'].append('Run-saving fielding')   
   
 # Identify areas for improvement   
 if row['Dropped Catches (DC)'] > 0:   
 analysis[player\_name]['Areas for Improvement'].append('Catching consistency')   
 if row['Missed Run Outs (MR)'] > 0:   
 analysis[player\_name]['Areas for Improvement'].append('Run-out execution')   
 if row['Runs Saved (RS)'] < 0:   
 analysis[player\_name]['Areas for Improvement'].append('Preventing extra runs')   
   
 return analysis  
  
def main():  
 print("Cricket Fielding Performance Analysis")  
 print("=" \* 50)  
 print("Players: Axar Patel, Kuldeep Yadav, Yash Dhull")  
 print("=" \* 50)  
  
 # Analyze fielding performance   
 results\_df = analyze\_fielding\_performance()   
   
 # Display the results   
 print("\nFielding Performance Metrics:")   
 print(results\_df[['Player Name', 'Clean Picks (CP)', 'Good Throws (GT)',   
 'Catches (C)', 'Dropped Catches (DC)', 'Stumpings (S)',   
 'Run Outs (RO)', 'Missed Run Outs (MR)', 'Direct Hits (DH)',   
 'Runs Saved (RS)', 'Performance Score (PS)', 'Calculated PS']])   
   
 # Generate detailed analysis   
 detailed\_analysis = generate\_detailed\_analysis(results\_df)   
   
 print("\n" + "=" \* 50)   
 print("Detailed Performance Analysis:")   
 print("=" \* 50)   
   
 for player, analysis in detailed\_analysis.items():   
 print(f"\n{player}:")   
 print(f" Performance Score: {analysis['Performance Score']}")   
 print(f" Strengths: {', '.join(analysis['Strengths']) if analysis['Strengths'] else 'None identified'}")   
 print(f" Areas for Improvement: {', '.join(analysis['Areas for Improvement']) if analysis['Areas for Improvement'] else 'None identified'}")   
   
 # Comparative analysis   
 print("\n" + "=" \* 50)   
 print("Comparative Analysis:")   
 print("=" \* 50)   
   
 best\_performer = results\_df.loc[results\_df['Performance Score (PS)'].idxmax()]   
 print(f"Best performer: {best\_performer['Player Name']} (PS: {best\_performer['Performance Score (PS)']})")   
   
 most\_clean\_picks = results\_df.loc[results\_df['Clean Picks (CP)'].idxmax()]   
 print(f"Most clean picks: {most\_clean\_picks['Player Name']} ({most\_clean\_picks['Clean Picks (CP)']})")   
   
 most\_runs\_saved = results\_df.loc[results\_df['Runs Saved (RS)'].idxmax()]   
 print(f"Most runs saved: {most\_runs\_saved['Player Name']} ({most\_runs\_saved['Runs Saved (RS)']})")  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

Output

