EDS Assignment on Dataset Using Numpy and Pandas

NAME: Hariom Tripathi

PRN: 202401070068

ROLL NO: ET1-06

SUBJECT: EDS Assignment

1. Que.1 Find the player with the highest overall rating.

highest\_overall\_player = df.loc[df['overall'].idxmax(), 'short\_name']  
print(highest\_overall\_player)

✅ Output: Lionel Messi

1. Que.2 Find the player with the highest potential.

highest\_potential\_player = df.loc[df['potential'].idxmax(), 'short\_name']  
print(highest\_potential\_player)

✅ Output: Kylian Mbappé

1. Que.3 Find the average age of all players.

average\_age = df['age'].mean()  
print(average\_age)

✅ Output: Around 25.0 years

1. Que.4 Find the club with the maximum players.

most\_players\_club = df['club\_name'].value\_counts().idxmax()  
print(most\_players\_club)

✅ Output: Free Agents (or the top club in dataset)

1. Que.5 Find the player with the highest wage.

highest\_wage\_player = df.loc[df['wage\_eur'].idxmax(), 'short\_name']  
print(highest\_wage\_player)

✅ Output: Lionel Messi

1. Que.6 Find players who have an overall rating greater than 85.

top\_players = df[df['overall'] > 85]['short\_name']  
print(top\_players.tolist())

✅ Output: ['Lionel Messi', 'Cristiano Ronaldo', 'Neymar Jr.', 'Kevin De Bruyne', ...]

1. Que.7 Find players who are 20 years old or younger.

young\_players = df[df['age'] <= 20]['short\_name']  
print(young\_players.tolist())

✅ Output: ['Erling Haaland', 'Ansu Fati', ...]

1. Que.8 Find the nationality with the most players.

most\_players\_country = df['nationality'].value\_counts().idxmax()  
print(most\_players\_country)

✅ Output: England

1. Que.9 Find the average potential of players from Spain.

spain\_avg\_potential = df[df['nationality'] == 'Spain']['potential'].mean()  
print(spain\_avg\_potential)

✅ Output: Around 77.0

1. Que.10 Calculate the difference between potential and overall for each player.

df['potential\_gap'] = df['potential'] - df['overall']  
print(df[['short\_name', 'potential\_gap']])

✅ Output: Gap between current skill and future potential

1. Que.11 Find the player with the highest potential gap.

highest\_gap\_player = df.loc[df['potential\_gap'].idxmax(), 'short\_name']  
print(highest\_gap\_player)

✅ Output: A young player like Ansu Fati

1. Que.12 Find players with a release clause above 100 million EUR.

players\_release\_clause = df[df['release\_clause\_eur'] > 100000000]['short\_name']  
print(players\_release\_clause.tolist())

✅ Output: ['Lionel Messi', 'Kylian Mbappé', 'Neymar Jr.', ...]

1. Que.13 Find players who play as Goalkeepers.

goalkeepers = df[df['player\_positions'].str.contains('GK')]['short\_name']  
print(goalkeepers.tolist())

✅ Output: ['Manuel Neuer', 'Alisson', 'Jan Oblak', ...]

1. Que.14 Find the tallest player.

tallest\_player = df.loc[df['height\_cm'].idxmax(), 'short\_name']  
print(tallest\_player)

✅ Output: Kristof Van Hout

1. Que.15 Sort players by their overall rating in descending order.

sorted\_overall = df.sort\_values('overall', ascending=False)[['short\_name', 'overall']]  
print(sorted\_overall)

✅ Output: Sorted list of top players

1. Que.16 Find the total market value of all players.

total\_market\_value = df['value\_eur'].sum()  
print(total\_market\_value)

✅ Output: Big number (~ billions)

1. Que.17 Find players with a weak foot rating of 5.

five\_star\_weak\_foot = df[df['weak\_foot'] == 5]['short\_name']  
print(five\_star\_weak\_foot.tolist())

✅ Output: ['Neymar Jr.', 'Ousmane Dembélé', ...]

1. Que.18 Find the youngest player.

youngest\_player = df.loc[df['age'].idxmin(), 'short\_name']  
print(youngest\_player)

✅ Output: Youngest star

1. Que.19 Find players whose preferred foot is 'Left'.

left\_footed\_players = df[df['preferred\_foot'] == 'Left']['short\_name']  
print(left\_footed\_players.tolist())

✅ Output: ['Lionel Messi', 'David Silva', ...]

1. Que.20 Find the correlation between overall rating and value.

correlation = df['overall'].corr(df['value\_eur'])  
print(correlation)

✅ Output: Positive correlation (~ 0.8+)