WHO IS THE NEXT HARRY KANE?

The project aims to find similar players to Harry Kane based on attributes from the Premier League data of the 2020/21 season obtained from the FBref website. [Link to Website.]

Two approaches are used for the analysis:

- Comparison of Harry Kane with other Players using Radar Plots.
- Recommendation System to find similar players to Harry Kane using the K-Nearest Neighbor(KNN) Technique.

```
#import required libraries
import io
import pandas as pd
import numpy as np
from copy import deepcopy
from google.colab import files

#import libraries for plotting
import plotly.express as px
```

Read and Explore Data

```
uploaded=files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving epl_general.xlsx to epl_general.xlsx
Saving epl_other.xlsx to epl_other.xlsx
Saving epl_possesion.xlsx to epl_possesion.xlsx
Saving epl_shooting.xlsx to epl_shooting.xlsx
```

```
gen_df=pd.read_excel(io.BytesIO(uploaded['epl_general.xlsx']))
shoot_df=pd.read_excel(io.BytesIO(uploaded['epl_shooting.xlsx']))
poss_df=pd.read_excel(io.BytesIO(uploaded['epl_possesion.xlsx']))
other_df=pd.read_excel(io.BytesIO(uploaded['epl_other.xlsx']))
```

```
gen_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 532 entries, 0 to 531
Data columns (total 33 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Rk             532 non-null    int64
 1   Player         532 non-null    object
 2   Nation         532 non-null    object
 3   Pos            532 non-null    object
 4   Squad          532 non-null    object
 5   Age            532 non-null    int64
 6   Born           532 non-null    int64
 7   MP             532 non-null    int64
 8   Starts         532 non-null    int64
 9   Min            532 non-null    int64
 10  90s            532 non-null    float64
 11  Gls            532 non-null    int64
 12  Ast            532 non-null    int64
 13  G-PK           532 non-null    int64
 14  PK             532 non-null    int64
 15  PKatt          532 non-null    int64
 16  CrdY           532 non-null    int64
 17  CrdR           532 non-null    int64
 18  Gls_per90      532 non-null    float64
 19  Ast_per90      532 non-null    float64
 20  G+A_per90      532 non-null    float64
 21  G-PK_per90     532 non-null    float64
 22  G+A-PK         532 non-null    float64
 23  xG             532 non-null    float64
 24  npxG           532 non-null    float64
 25  xA             532 non-null    float64
 26  npxG+xA        532 non-null    float64
 27  xG_per90       532 non-null    float64
 28  xA_per90       532 non-null    float64
 29  xG+xA_per90    532 non-null    float64
 30  npxG_per90     532 non-null    float64
 31  npxG+xA_per90  532 non-null    float64
 32  Matches        532 non-null    object
dtypes: float64(15), int64(13), object(5)
memory usage: 137.3+ KB
```

```
gen_new=gen_df[["Rk","Player","Pos","Age","Min","Gls","Ast","Ast_per90","Gls_per90"]]  #Obtain only required columns for General Player Statistics
```

```
shoot_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 532 entries, 0 to 531
Data columns (total 26 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   Rk       532 non-null    int64
 1   Player   532 non-null    object
 2   Nation   532 non-null    object
 3   Pos      532 non-null    object
 4   Squad    532 non-null    object
 5   Age      532 non-null    int64
 6   Born     532 non-null    int64
 7   90s      532 non-null    float64
 8   Gls      532 non-null    int64
 9   Sh       532 non-null    int64
 10  SoT      532 non-null    int64
```

```
 11  SoT%     435 non-null    float64
 12  Sh/90    532 non-null    float64
 13  SoT/90   532 non-null    float64
 14  G/Sh     435 non-null    float64
 15  G/SoT    373 non-null    float64
 16  Dist     435 non-null    float64
 17  FK       532 non-null    int64
 18  PK       532 non-null    int64
 19  PKatt    532 non-null    int64
 20  xG       532 non-null    float64
 21  npxG     532 non-null    float64
 22  npxG/Sh  435 non-null    float64
 23  G-xG     532 non-null    float64
 24  np:G-xG  532 non-null    float64
 25  Matches  532 non-null    object
dtypes: float64(12), int64(9), object(5)
memory usage: 108.2+ KB
```

```
shoot_new=shoot_df[["Rk","Sh/90","SoT/90","G/Sh","G/SoT"]]    #Obtain only required columns of  Player Shooting Statistics
```

```
poss_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 532 entries, 0 to 531
Data columns (total 33 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Rk                   532 non-null    int64
 1   Player               532 non-null    object
 2   Nation               532 non-null    object
 3   Pos                  532 non-null    object
 4   Squad                532 non-null    object
 5   Age                  532 non-null    int64
 6   Born                 532 non-null    int64
 7   90s                  532 non-null    float64
 8   Total_Touches        532 non-null    int64
 9   Touches_Def Pen      532 non-null    int64
 10  Def 3rd              532 non-null    int64
 11  Touches_Mid 3rd      532 non-null    int64
 12  Touches_Att 3rd      532 non-null    int64
 13  Touches_Att Pen      532 non-null    int64
 14  Touches_Live         532 non-null    int64
 15  Dribbles_Succ        532 non-null    int64
 16  Dribbles_Att         532 non-null    int64
 17  Dribbles_Succ%       436 non-null    float64
 18  Dribbles_#Pl         532 non-null    int64
 19  Dribbles_Megs        532 non-null    int64
 20  Carries              532 non-null    int64
 21  TotDist              532 non-null    int64
 22  PrgDist              532 non-null    int64
 23  Prog                 532 non-null    int64
 24  2021-03-01 00:00:00  532 non-null    int64
 25  CPA                  532 non-null    int64
 26  Mis                  532 non-null    int64
 27  Dis                  532 non-null    int64
 28  Receiving_Targ       532 non-null    int64
 29  Rec                  532 non-null    int64
 30  Rec%                 528 non-null    float64
 31  Prog.1               532 non-null    int64
 32  Matches              532 non-null    object
dtypes: float64(3), int64(25), object(5)
memory usage: 137.3+ KB
```

```
poss_new=poss_df[["Rk","Touches_Att 3rd","Touches_Att Pen","Dribbles_Succ"]]      #Obtain only required columns of  Player Possesion Statistics
```

```
other_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 532 entries, 0 to 531
Data columns (total 25 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Rk                 532 non-null    int64
 1   Player             532 non-null    object
 2   Nation             532 non-null    object
 3   Pos                532 non-null    object
 4   Squad              532 non-null    object
 5   Age                532 non-null    int64
 6   Born               532 non-null    int64
 7   90s                532 non-null    float64
 8   CrdY               532 non-null    int64
 9   CrdR               532 non-null    int64
 10  2CrdY              532 non-null    int64
 11  Fls                532 non-null    int64
 12  Fld                532 non-null    int64
 13  Off                532 non-null    int64
 14  Crs                532 non-null    int64
 15  Int                532 non-null    int64
 16  TklW               532 non-null    int64
 17  PKwon              532 non-null    int64
 18  PKcon              532 non-null    int64
 19  OG                 532 non-null    int64
 20  Recov              532 non-null    int64
 21  AerialDuels_Won    532 non-null    int64
 22  AerialDuels_Lost   532 non-null    int64
 23  AerialDuels_Won%   464 non-null    float64
 24  Matches            532 non-null    object
dtypes: float64(2), int64(18), object(5)
memory usage: 104.0+ KB
```

```
other_new=other_df[["Rk","AerialDuels_Won"]]    #Obtain only required columns of  Other Player Statistics
```

Merging and Cleaning Dataframes

```
from functools import reduce
```

```
df=[gen_new,shoot_new,poss_new,other_new]                                              # merging datasets
merge_df=reduce(lambda left,right: pd.merge(left,right, on=['Rk'], how='outer'),df)
merge_df
```

	Rk	Player	Pos	Age	Min	Gls	Ast	Ast_per90	Gls_per90	Sh/90	SoT/90	G/Sh	G/SoT	Touches_Att 3rd	Touches_Att Pen	Dribbles_Succ	AerialDuels_Won
0	1	Patrick van Aanholt\Patrick-van-Aanholt	DF	29	1777	0	1	0.05	0.00	0.91	0.30	0.00	0.00	344	38	9	10
1	2	Tammy Abraham\Tammy-Abraham	FW	22	1040	6	1	0.09	0.52	2.77	1.13	0.19	0.46	166	66	6	39
2	3	Che Adams\Che-Adams	FW	24	2667	9	5	0.17	0.30	1.89	1.05	0.16	0.29	492	110	25	47
3	4	Tosin Adarabioyo\Tosin-Adarabioyo	DF	22	2953	0	0	0.00	0.00	0.61	0.15	0.00	0.00	73	34	8	100
4	5	Adrián\Adrian	GK	33	270	0	0	0.00	0.00	0.00	0.00	NaN	NaN	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
527	528	Andi Zeqiri\Andi-Zeqiri	FWDF	21	171	0	0	0.00	0.00	3.68	1.05	0.00	0.00	37	15	1	2
528	529	Oleksandr Zinchenko\Oleksandr-Zinchenko	DF	23	1478	0	0	0.00	0.00	0.97	0.24	0.00	0.00	444	10	5	24
529	530	Hakim Ziyech\Hakim-Ziyech	FWMF	27	1172	2	3	0.23	0.15	2.69	0.92	0.06	0.17	443	37	18	3
530	531	Kurt Zouma\Kurt-Zouma	DF	25	2029	5	0	0.00	0.22	1.20	0.53	0.19	0.42	75	31	5	97
531	532	Martin Ødegaard\Martin-Odegaard	MF	21	866	1	2	0.21	0.10	1.56	0.31	0.07	0.33	289	26	12	1

532 rows × 17 columns

> For Analysis,the 2nd preferred position is removed under the 'Pos' Column keeping only the 1st position as the main role.

> For example 'FWMF' consists of both Forward(FW) as well as Midfielder(MF) positions.The latter characters are deleted to obtain only the former part as the main position for the player concerned.

> Also the first instance before the '' symbol is kept for the 'Player' column.

```
for i in range(len(merge_df)):
  merge_df['Pos'][i]=merge_df['Pos'][i][:2]                              #removing the 2nd playing position of player
  merge_df['Player'][i]= "".join(merge_df['Player'][i].split("\\")[:-1])  #delete the player name after '\' symbol.
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

```
merge_df[["G/Sh","G/SoT"]]=merge_df[["G/Sh","G/SoT"]].replace(np.nan,0)          #replacing NULL values with zero
```

```
#insert per90 stats columns of required attributes

add_Per90=["Touches_Att 3rd","Touches_Att Pen","Dribbles_Succ","AerialDuels_Won"]       # columns to be added

for col in add_Per90:
  merge_df[col + "_Per90"]=merge_df[col].divide(merge_df['Min']).multiply(90).round(2)    # merge column to dataset and calculate Per90 stats
```

> The following are all the Attributes which will be used for further Analysis.

- Pos-->Position
- Min--> Total Minutes Played
- Gls-->Total Goals Scored
- Ast-->Total Assists
- Gls_per90-->Goals scored per match
- Ast-->Assists per match
- Sh/90-->Shots per match
- SoT/90-->Shots on Target per match
- G/Sh-->Goals per Shot
- G/SoT-->Goals per Shot on Target
- Touches_Att 3rd-->Touches taken in the Attacking 3rd of the pitch
- Touches_Att Pen-->Touches taken inside the Attacking Penalty box
- Dribbles_Succ-->Successful Dribbles completed.
- AerialDuels_Won-->Aerial Battles won

```
epl=merge_df
epl
```

	Rk	Player	Pos	Age	Min	Gls	Ast	Ast_per90	Gls_per90	Sh/90	SoT/90	G/Sh	G/SoT	Touches_Att 3rd	Touches_Att Pen	Dribbles_Succ	AerialDuels_Won	Touches_Att 3rd_Per90	Touches_Att Pen_Per90	Dribbles_Succ_Per90	AerialDuels_Won_Per90
0	1	Patrick van Aanholt	DF	29	1777	0	1	0.05	0.00	0.91	0.30	0.00	0.00	344	38	9	10	17.42	1.92	0.46	0.51
1	2	Tammy Abraham	FW	22	1040	6	1	0.09	0.52	2.77	1.13	0.19	0.46	166	66	6	39	14.37	5.71	0.52	3.38
2	3	Che Adams	FW	24	2667	9	5	0.17	0.30	1.89	1.05	0.16	0.29	492	110	25	47	16.60	3.71	0.84	1.59
3	4	Tosin Adarabioyo	DF	22	2953	0	0	0.00	0.00	0.61	0.15	0.00	0.00	73	34	8	100	2.22	1.04	0.24	3.05
4	5	Adrián	GK	33	270	0	0	0.00	0.00	0.00	0.00	0.00	0.00	0	0	0	0	0.00	0.00	0.00	0.00
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
527	528	Andi Zeqiri	FW	21	171	0	0	0.00	0.00	3.68	1.05	0.00	0.00	37	15	1	2	19.47	7.89	0.53	1.05

```
epl.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 532 entries, 0 to 531
Data columns (total 21 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   Rk                     532 non-null    int64  
 1   Player                 532 non-null    object 
 2   Pos                    532 non-null    object 
 3   Age                    532 non-null    int64  
 4   Min                    532 non-null    int64  
 5   Gls                    532 non-null    int64  
 6   Ast                    532 non-null    int64  
 7   Ast_per90              532 non-null    float64
 8   Gls_per90              532 non-null    float64
 9   Sh/90                  532 non-null    float64
 10  SoT/90                 532 non-null    float64
 11  G/Sh                   532 non-null    float64
 12  G/SoT                  532 non-null    float64
 13  Touches_Att 3rd        532 non-null    int64  
 14  Touches_Att Pen        532 non-null    int64  
 15  Dribbles_Succ          532 non-null    int64  
 16  AerialDuels_Won        532 non-null    int64  
 17  Touches_Att 3rd_Per90  532 non-null    float64
 18  Touches_Att Pen_Per90  532 non-null    float64
 19  Dribbles_Succ_Per90    532 non-null    float64
 20  AerialDuels_Won_Per90  532 non-null    float64
dtypes: float64(10), int64(9), object(2)
memory usage: 111.4+ KB
```

```
epl.describe().round(2)       # summary statistics of the epl dataset
```

	Rk	Age	Min	Gls	Ast	Ast_per90	Gls_per90	Sh/90	SoT/90	G/Sh	G/SoT	Touches_Att 3rd	Touches_Att Pen	Dribbles_Succ	AerialDuels_Won	Touches_Att 3rd_Per90	Touches_Att Pen_Per90	Dribbles_Succ_Per90	AerialDuels_Won_Per90
count	532.00	532.00	532.00	532.00	532.00	532.00	532.00	532.00	532.00	532.00	532.00	532.00	532.00	532.00	532.00	532.00	532.00	532.00	532.00
mean	266.50	25.50	1411.44	1.85	1.29	0.07	0.10	1.10	0.36	0.07	0.18	230.70	33.66	13.94	24.65	16.33	2.31	0.91	1.58
std	153.72	4.32	1043.17	3.34	2.10	0.11	0.16	1.15	0.62	0.10	0.24	254.00	44.92	17.97	32.10	19.50	2.90	1.56	2.05
min	1.00	16.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25%	133.75	22.00	426.00	0.00	0.00	0.00	0.00	0.28	0.00	0.00	0.00	25.75	3.00	1.00	3.00	5.27	0.56	0.12	0.44
50%	266.50	26.00	1345.00	1.00	0.00	0.00	0.03	0.73	0.18	0.00	0.00	138.50	17.00	7.00	13.00	15.82	1.48	0.60	1.11
75%	399.25	29.00	2303.50	2.00	2.00	0.11	0.13	1.73	0.53	0.11	0.33	371.00	45.25	20.00	34.00	22.60	3.36	1.34	2.18
max	532.00	38.00	3420.00	23.00	14.00	0.87	1.08	10.00	10.00	1.00	1.00	1339.00	313.00	160.00	228.00	360.00	30.00	30.00	30.00

Data Slicing

```
epl[(epl['Min']>=1400)&(epl['Pos']=="FW")].shape        #Acquiring only epl Forwards data
```

```
(54, 21)
```

- Only Forwards of EPL 2020/21 season are taken into consideration for Analysis since we are comparing similarity between forwards position
- Forwards who played more or equal to the average minutes(i:e 1400) played by players overall are considered.

```
epl_fw=deepcopy(epl[(epl['Min']>=1400)&(epl['Pos']=="FW")])       # Slicing data for only forwards who played at least 1400 minutes of football
epl_fw.reset_index(drop=True,inplace=True)
final_data=epl_fw                                                #Store dataset for ML Analysis
epl_fw.head(49)
```

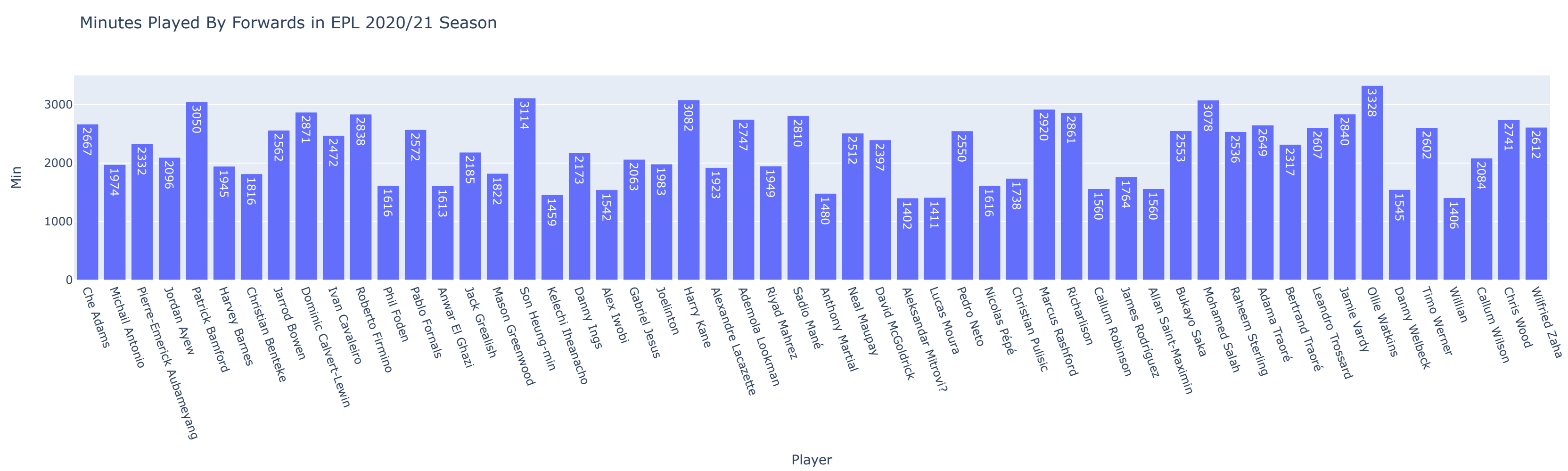
	Rk	Player	Pos	Age	Min	Gls	Ast	Ast_per90	Gls_per90	Sh/90	SoT/90	G/Sh	G/SoT	Touches_Att 3rd	Touches_Att Pen	Dribbles_Succ	AerialDuels_Won	Touches_Att 3rd_Per90	Touches_Att Pen_Per90	Dribbles_Succ_Per90	AerialDuels_Won_Per90
0	3	Che Adams	FW	24	2667	9	5	0.17	0.30	1.89	1.05	0.16	0.29	492	110	25	47	16.60	3.71	0.84	1.59
1	28	Michail Antonio	FW	30	1974	10	5	0.23	0.46	2.92	1.05	0.16	0.43	480	130	47	63	21.88	5.93	2.14	2.87
2	32	Pierre-Emerick Aubameyang	FW	31	2332	10	3	0.12	0.39	2.16	0.73	0.14	0.42	471	126	12	31	18.18	4.86	0.46	1.20
3	35	Jordan Ayew	FW	28	2096	1	3	0.13	0.04	1.07	0.47	0.04	0.09	420	63	45	31	18.03	2.71	1.93	1.33
4	42	Patrick Bamford	FW	26	3050	17	7	0.21	0.50	3.10	1.33	0.14	0.33	408	183	17	54	12.04	5.40	0.50	1.59
5	47	Harvey Barnes	FW	22	1945	9	4	0.19	0.42	2.54	1.16	0.16	0.36	516	109	28	22	23.88	5.04	1.30	1.02
6	56	Christian Benteke	FW	29	1816	10	1	0.05	0.50	3.52	1.64	0.14	0.30	360	127	16	212	17.84	6.29	0.79	10.51
7	67	Jarrod Bowen	FW	23	2562	8	5	0.18	0.28	1.93	0.56	0.15	0.50	579	120	40	20	20.34	4.22	1.41	0.70
8	82	Dominic Calvert-Lewin	FW	23	2871	16	0	0.00	0.50	2.54	1.44	0.20	0.35	545	156	20	143	17.08	4.89	0.63	4.48
9	92	Ivan Cavaleiro	FW	26	2472	3	0	0.00	0.11	1.67	0.58	0.04	0.13	620	92	52	46	22.57	3.35	1.89	1.67
10	156	Roberto Firmino	FW	28	2838	9	7	0.22	0.29	2.63	1.01	0.11	0.28	859	217	50	25	27.24	6.88	1.59	0.79
11	158	Phil Foden	FW	20	1616	9	5	0.28	0.50	2.51	1.11	0.20	0.45	653	144	43	14	36.37	8.02	2.39	0.78
12	160	Pablo Fornals	FW	24	2572	5	4	0.14	0.17	1.75	0.52	0.10	0.33	551	74	21	25	19.28	2.59	0.73	0.87
13	174	Anwar El Ghazi	FW	25	1613	10	0	0.00	0.56	3.35	1.45	0.10	0.23	521	84	28	30	29.07	4.69	1.56	1.67
14	185	Jack Grealish	FW	24	2185	6	10	0.41	0.25	2.02	0.70	0.12	0.35	817	177	61	24	33.65	7.29	2.51	0.99
15	186	Mason Greenwood	FW	18	1822	7	2	0.10	0.35	3.31	1.14	0.10	0.30	528	113	38	7	26.08	5.58	1.88	0.35
16	204	Son Heung-min	FW	28	3114	17	10	0.29	0.49	1.97	1.01	0.24	0.46	813	143	38	24	23.50	4.13	1.10	0.69
17	213	Kelechi Iheanacho	FW	23	1459	12	2	0.12	0.74	3.58	1.54	0.21	0.48	418	81	29	9	25.78	5.00	1.79	0.56
18	214	Danny Ings	FW	28	2173	12	4	0.17	0.50	2.28	0.99	0.18	0.42	405	122	37	21	16.77	5.05	1.53	0.87
19	216	Alex Iwobi	FW	24	1542	1	2	0.12	0.06	0.88	0.18	0.07	0.33	363	46	46	4	21.19	2.68	2.68	0.23
20	224	Gabriel Jesus	FW	23	2063	9	4	0.17	0.39	2.36	0.70	0.17	0.56	601	151	45	14	26.22	6.59	1.96	0.61
21	226	Joelinton	FW	23	1983	4	2	0.09	0.18	2.04	0.86	0.07	0.16	371	85	39	98	16.84	3.86	1.77	4.45
22	236	Harry Kane	FW	27	3082	23	14	0.41	0.67	3.91	1.37	0.14	0.40	616	166	46	90	17.99	4.85	1.34	2.63
23	257	Alexandre Lacazette	FW	29	1923	13	2	0.09	0.61	2.01	1.17	0.23	0.40	350	102	20	30	16.38	4.77	0.94	1.40
24	281	Ademola Lookman	FW	22	2747	4	4	0.13	0.13	2.23	0.69	0.06	0.19	871	144	83	7	28.54	4.72	2.72	0.23
25	290	Riyad Mahrez	FW	29	1949	9	6	0.28	0.42	2.68	1.02	0.16	0.41	876	156	36	15	40.45	7.20	1.66	0.69
26	294	Sadio Mané	FW	28	2810	11	7	0.22	0.35	2.98	1.12	0.12	0.31	1016	283	84	28	32.54	9.06	2.69	0.90
27	299	Anthony Martial	FW	24	1480	4	3	0.18	0.24	2.61	1.28	0.09	0.19	416	121	34	15	25.30	7.36	2.07	0.91
28	307	Neal Maupay	FW	23	2512	8	2	0.07	0.29	2.40	0.79	0.07	0.23	527	171	24	26	18.88	6.13	0.86	0.93
29	313	David McGoldrick	FW	32	2397	8	1	0.04	0.30	2.70	0.90	0.11	0.33	712	101	46	94	26.73	3.79	1.73	3.53
30	329	Aleksandar Mitrovi?	FW	25	1402	3	3	0.19	0.19	3.59	1.09	0.04	0.12	318	117	14	68	20.41	7.51	0.90	4.37
31	335	Lucas Moura	FW	27	1411	3	4	0.26	0.19	1.15	0.38	0.17	0.50	328	46	36	26	20.92	2.93	2.30	1.66
32	346	Pedro Neto	FW	20	2550	5	6	0.21	0.18	2.15	0.74	0.08	0.24	856	111	74	20	30.21	3.92	2.61	0.71
33	370	Nicolas Pépé	FW	25	1616	10	1	0.06	0.56	2.67	1.11	0.19	0.45	519	115	31	35	28.90	6.40	1.73	1.95
34	385	Christian	FW	21	1738	4	2	0.10	0.21	2.23	0.83	0.09	0.25	547	117	46	21	28.33	6.06	2.38	1.00

```
#Finding Harry Kane 'Rk' id
epl_fw[epl_fw['Player'].str.contains("Kane")]
```

	Rk	Player	Pos	Age	Min	Gls	Ast	Ast_per90	Gls_per90	Sh/90	SoT/90	G/Sh	G/SoT	Touches_Att 3rd	Touches_Att Pen	Dribbles_Succ	AerialDuels_Won	Touches_Att 3rd_Per90	Touches_Att Pen_Per90	Dribbles_Succ_Per90	AerialDuels_Won_Per90
22	236	Harry Kane	FW	27	3082	23	14	0.41	0.67	3.91	1.37	0.14	0.4	616	166	46	90	17.99	4.85	1.34	2.63

Minutes Played By Forwards In The EPL 2020/21 Season

```
#Bar Plot of Minutes played by Forwards who played more than average time
fw_min=epl_fw[["Player","Min"]]
fig=px.bar(fw_min,x="Player",y="Min",text="Min",title="Minutes Played By Forwards in EPL 2020/21 Season")
fig.update_xaxes(tickangle=70)
```



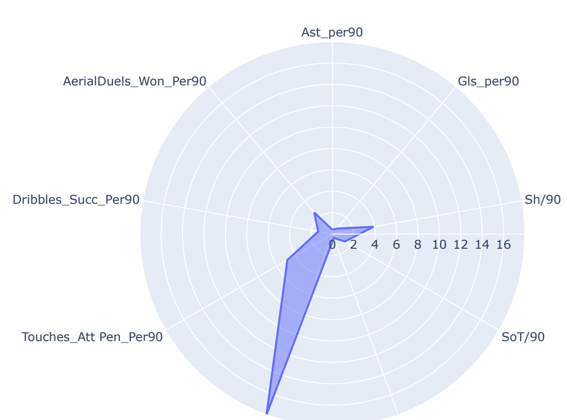
> From the above bar plot, Ollie Watkins, Son Heung-Min, Harry Kane, Mohamed Salah and Patrick Bamford are the forwards that played the most minutes during the EPL 2020/21 season whereas Aleksandar Mitrovic and Willian had the least playing time satisfying the 1400 minutes threshold.

Harry Kane Statistics

```
#Harry Kane Stats
stat_per90=["Ast_per90","Gls_per90","Sh/90","SoT/90","G/SoT","Touches_Att 3rd_Per90","Touches_Att Pen_Per90","Dribbles_Succ_Per90","AerialDuels_Won_Per90"]
epl_fw.loc[(epl_fw["Rk"]== 236), stat_per90].sum()
```

```
Ast_per90                0.41
Gls_per90                0.67
Sh/90                    3.91
SoT/90                   1.37
G/SoT                    0.40
Touches_Att 3rd_Per90   17.99
Touches_Att Pen_Per90    4.85
Dribbles_Succ_Per90      1.34
AerialDuels_Won_Per90    2.63
dtype: float64
```

```
#Harry Kane Radar Plot
fig2=px.line_polar(epl_fw,r=epl_fw.loc[(epl_fw["Rk"]== 236), stat_per90].sum(),
                theta=stat_per90 ,line_close=True)
fig2.update_traces(fill='toself')
fig2.show()
```



Normalize Values

> As can be seen from the above Radar Plot Visualization, not a lot of information can be retrieved as the data values have different scales in each column. Normalization is done to get all the attributes to the same scale for better Analysis.

```
#Normalization Dataset using MinMaxScaler
from sklearn.preprocessing import MinMaxScaler
sc=MinMaxScaler()

epl_fw[stat_per90]= sc.fit_transform(epl_fw[stat_per90])
```

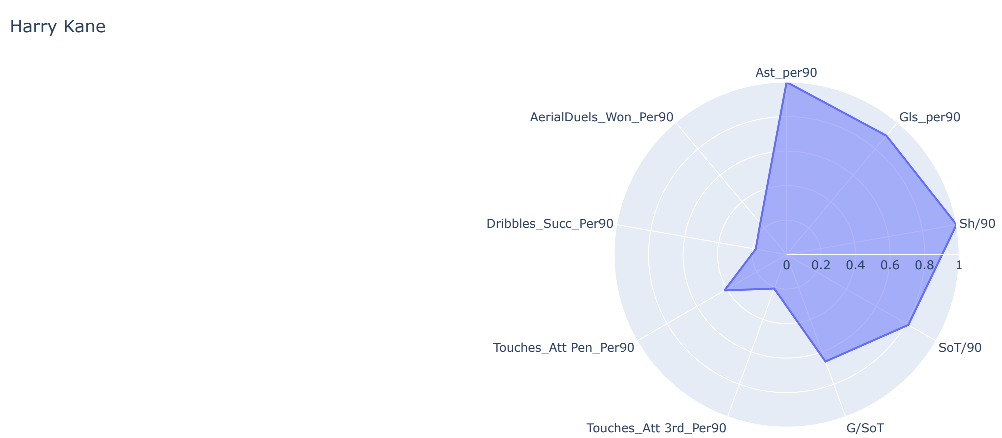
```
epl_fw[stat_per90].max()
```

```
Ast_per90               1.0
Gls_per90               1.0
Sh/90                   1.0
SoT/90                  1.0
G/SoT                   1.0
Touches_Att 3rd_Per90   1.0
Touches_Att Pen_Per90   1.0
Dribbles_Succ_Per90     1.0
AerialDuels_Won_Per90   1.0
dtype: float64
```

```
epl_fw[epl_fw["Rk"]==236]     #Normalized Harry Kane Per90 Statistics
```

	Rk	Player	Pos	Age	Min	Gls	Ast	Ast_per90	Gls_per90	Sh/90	SoT/90	G/Sh	G/SoT	Touches_Att 3rd	Touches_Att Pen	Dribbles_Succ	AerialDuels_Won	Touches_Att 3rd_Per90	Touches_Att Pen_Per90	Dribbles_Succ_Per90	AerialDuels_Won_Per90
22	236	Harry Kane	FW	27	3082	23	14	1.0	0.9	1.0	0.815068	0.14	0.659574	616	166	46	90	0.209433	0.415761	0.181637	0.237173

```
#Normalized Plot for Harry Kane
fig3=px.line_polar(epl_fw,r=epl_fw.loc[(epl_fw["Rk"]== 236), stat_per90].sum(),
                   theta=stat_per90 ,title="Harry Kane",line_close=True)
fig3.update_traces(fill='toself')
fig3.show()
```



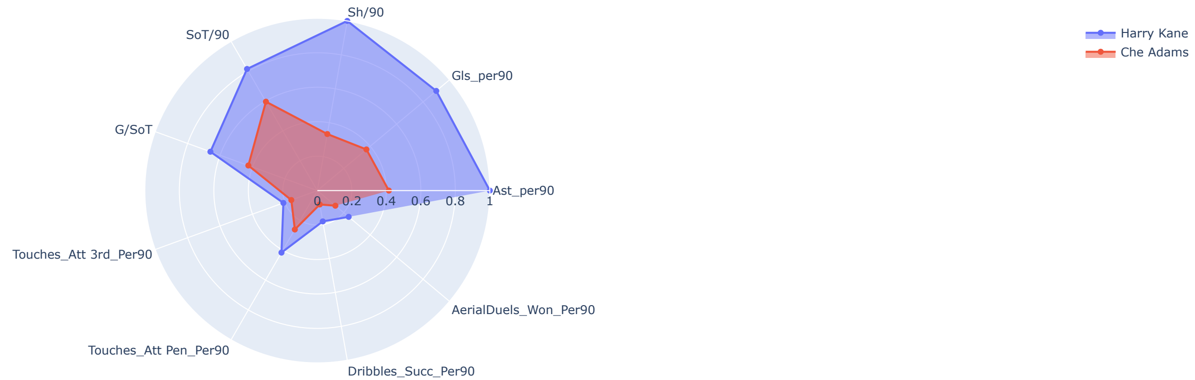
> The above plot is a better visualization which will allow to compare each player to Harry Kane based on each attribute.

1) METHOD I: PLAYER COMPARISONS USING RADAR PLOTS

```
import plotly.graph_objects as go

for i,row in epl_fw.iterrows():
  if row["Rk"]==236:
    continue

  print(row["Player"])

  fig4=go.Figure()

  fig4.add_trace(go.Scatterpolar(r=epl_fw.loc[(epl_fw["Rk"]== 236), stat_per90].sum(),
                 theta=stat_per90 ,name="Harry Kane",fill='toself'))

  fig4.add_trace(go.Scatterpolar(r=epl_fw.loc[(epl_fw["Rk"]== row["Rk"]), stat_per90].sum(),
                 theta=stat_per90 ,name=row["Player"],fill='toself'))

  fig4.update_layout(title="Harry Kane vs "+row["Player"], polar=dict(radialaxis= dict(visible=True, range=[0,epl_max])), showlegend=True)

  fig4.show()
```

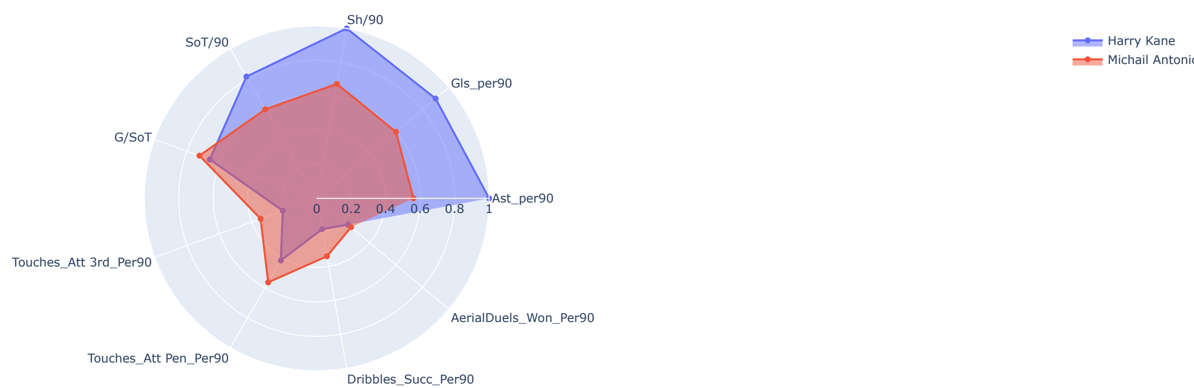
Che Adams

Harry Kane vs Che Adams



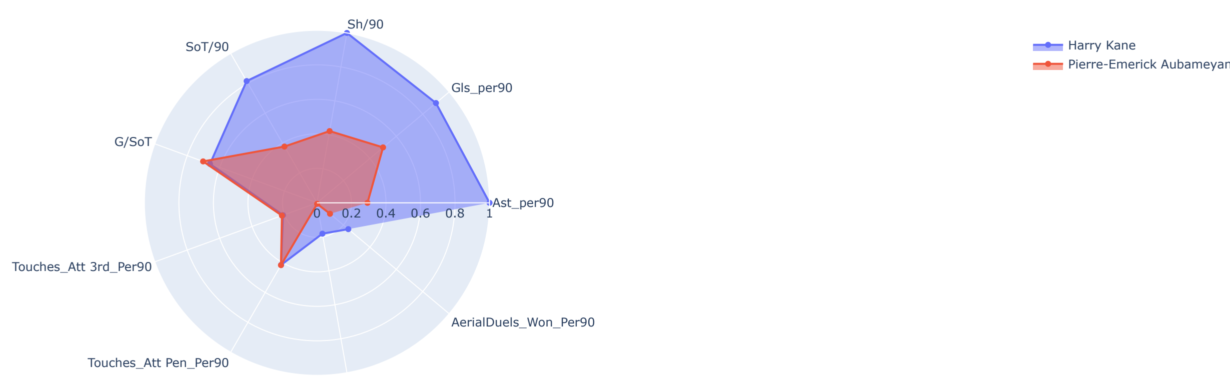
Michail Antonio

Harry Kane vs Michail Antonio



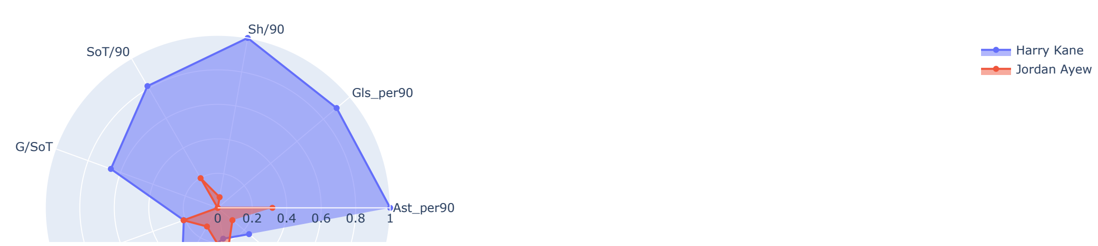
Pierre-Emerick Aubameyang

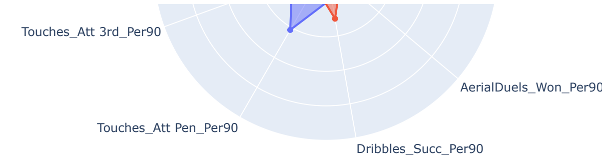
Harry Kane vs Pierre-Emerick Aubameyang



Jordan Ayew

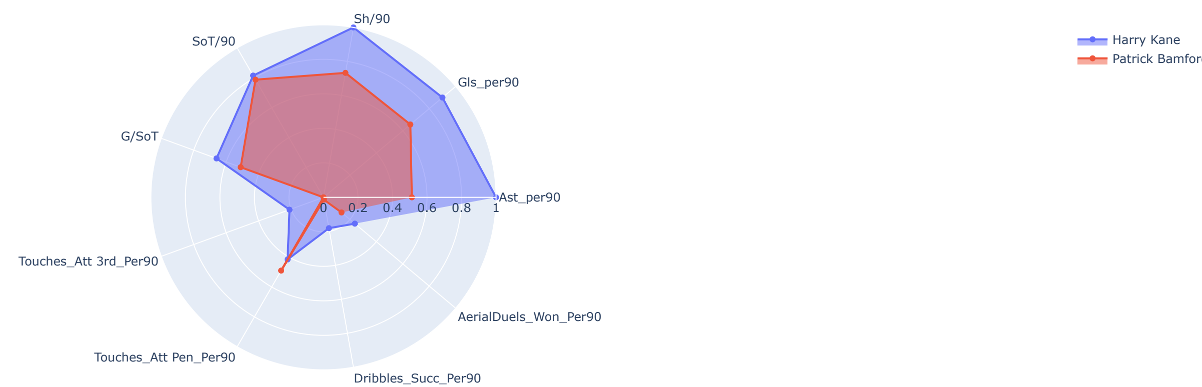
Harry Kane vs Jordan Ayew





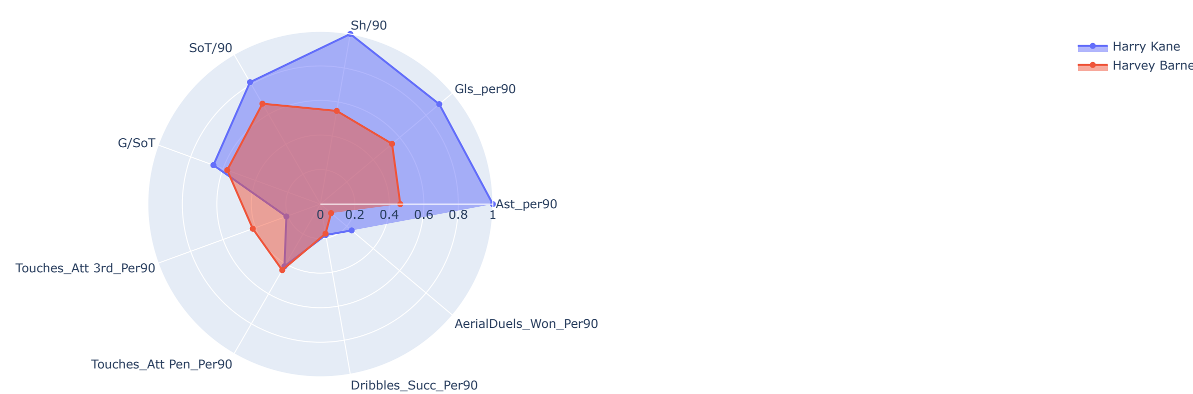
Patrick Bamford

Harry Kane vs Patrick Bamford



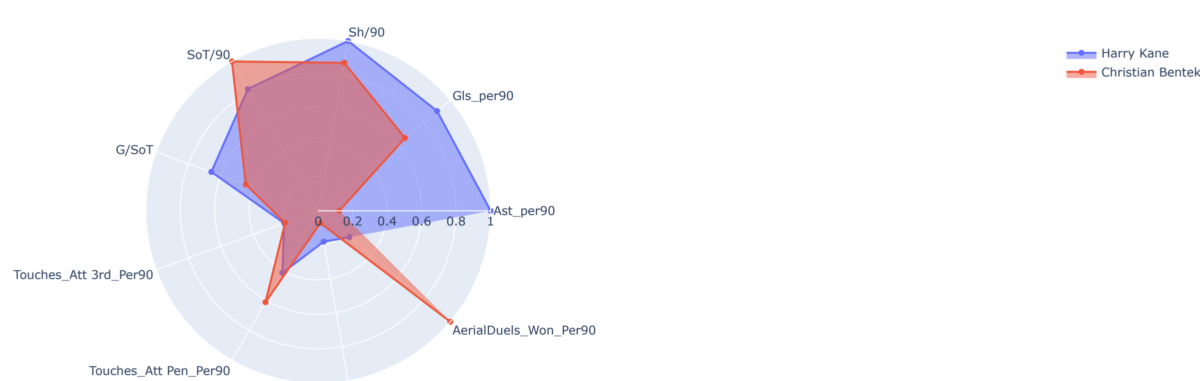
Harvey Barnes

Harry Kane vs Harvey Barnes



Christian Benteke

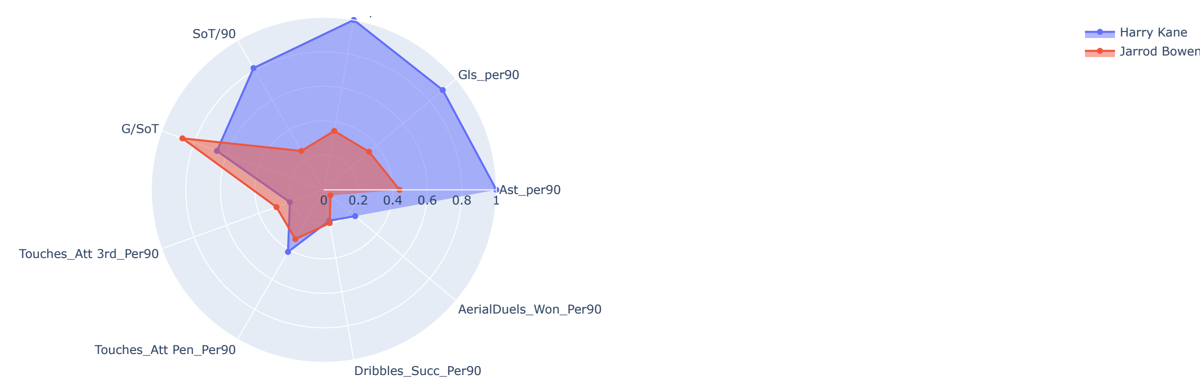
Harry Kane vs Christian Benteke



Jarrod Bowen

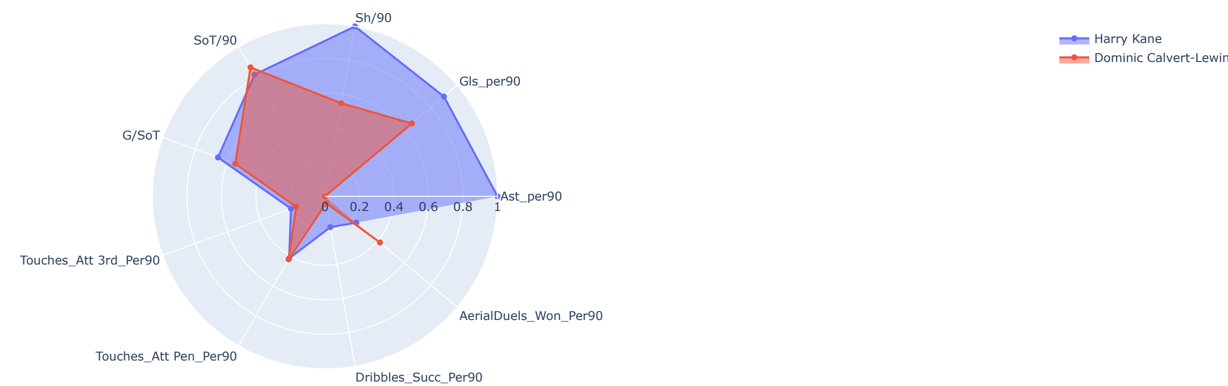
Harry Kane vs Jarrod Bowen

Sh/90



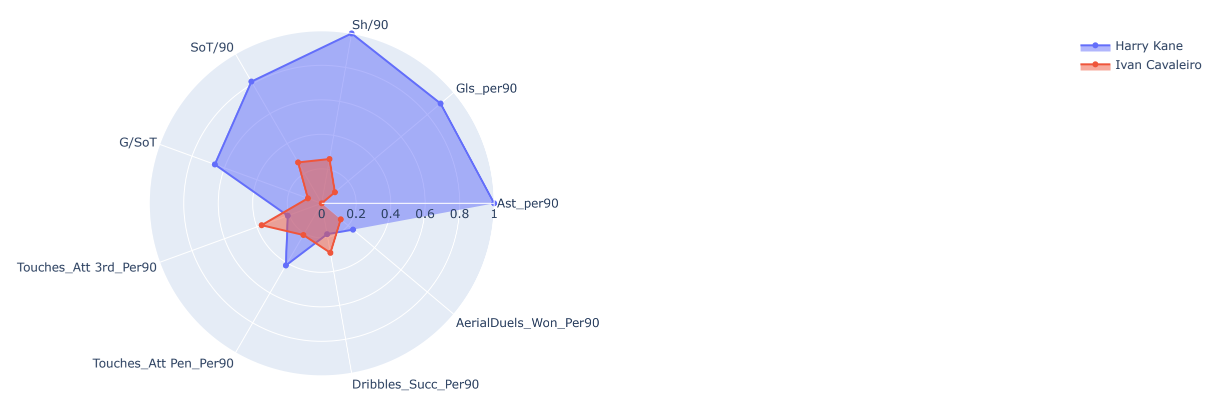
Dominic Calvert-Lewin

Harry Kane vs Dominic Calvert-Lewin



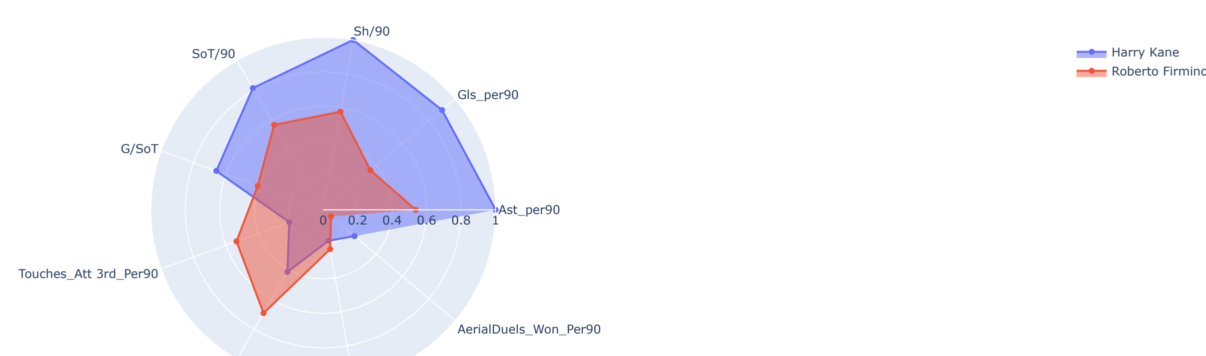
Ivan Cavaleiro

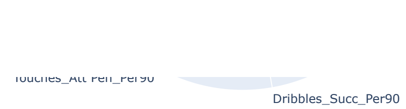
Harry Kane vs Ivan Cavaleiro



Roberto Firmino

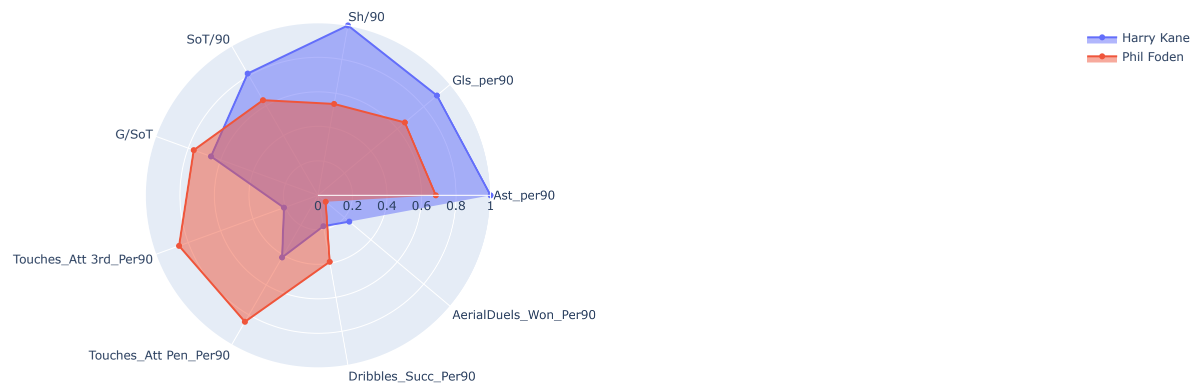
Harry Kane vs Roberto Firmino





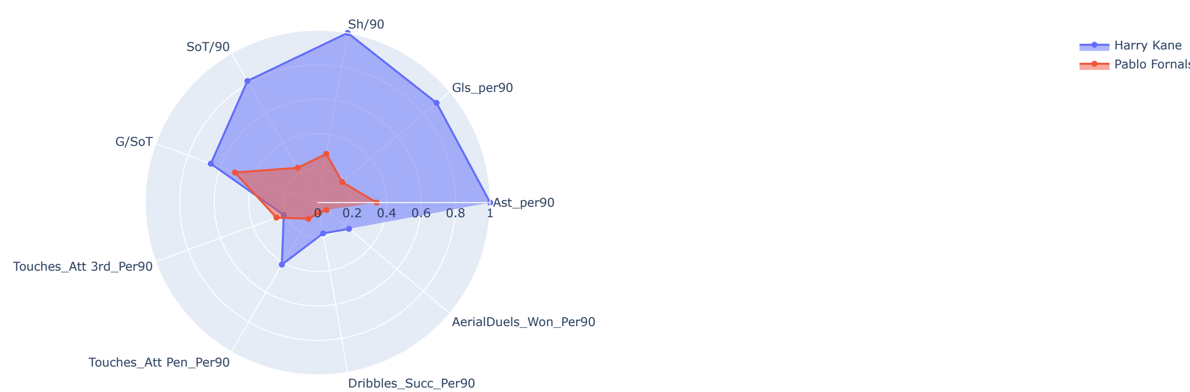
Phil Foden

Harry Kane vs Phil Foden



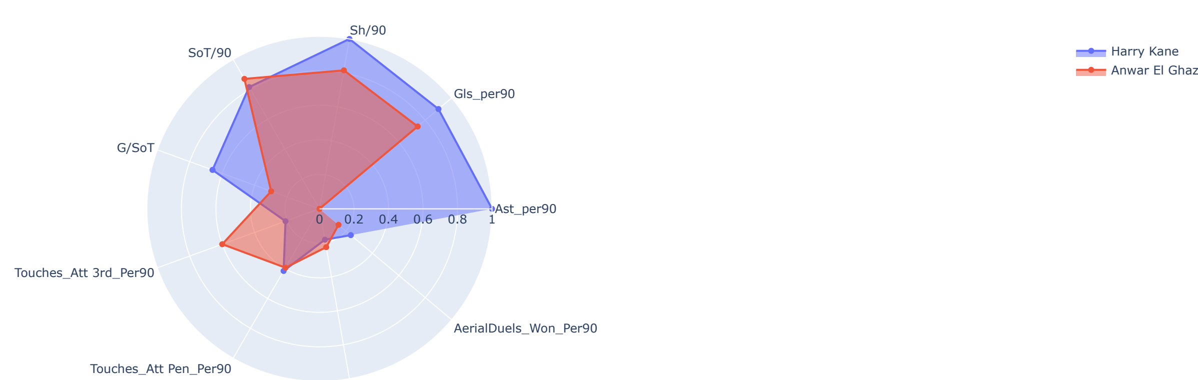
Pablo Fornals

Harry Kane vs Pablo Fornals



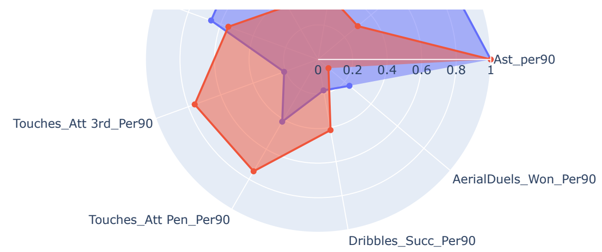
Anwar El Ghazi

Harry Kane vs Anwar El Ghazi



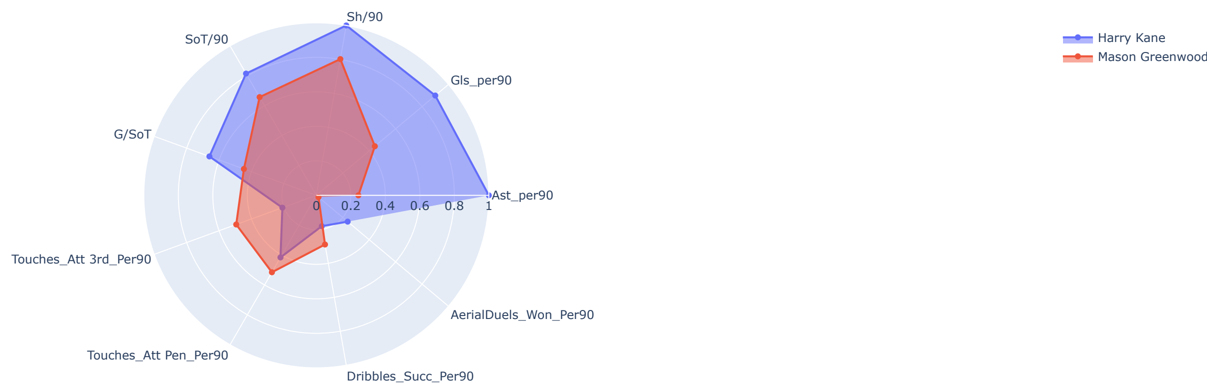
Jack Grealish

Harry Kane vs Jack Grealish



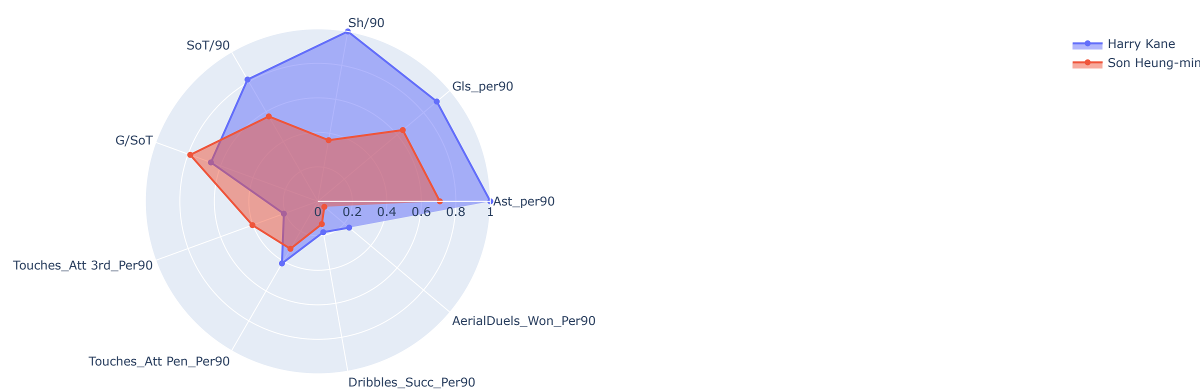
Mason Greenwood

Harry Kane vs Mason Greenwood



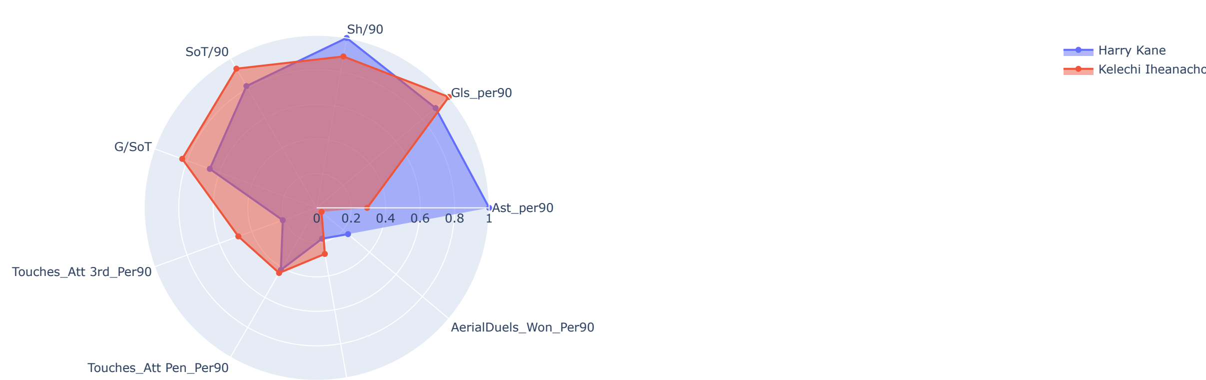
Son Heung-min

Harry Kane vs Son Heung-min



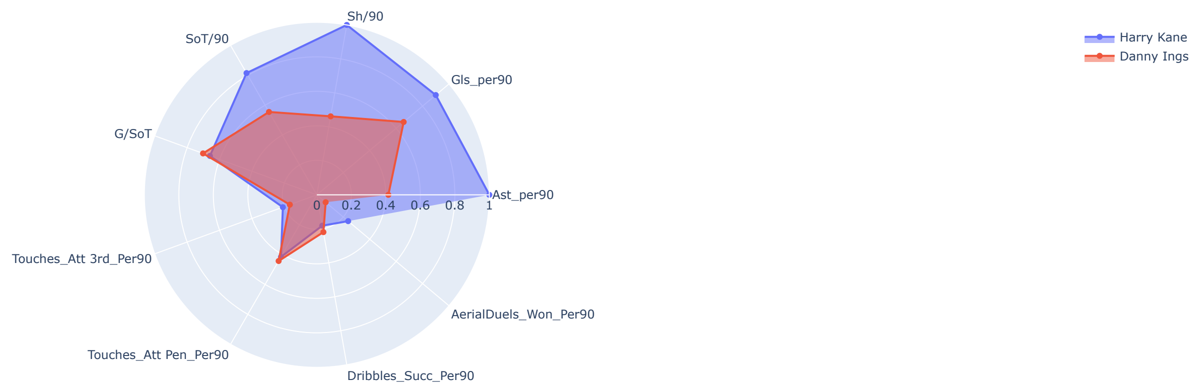
Kelechi Iheanacho

Harry Kane vs Kelechi Iheanacho



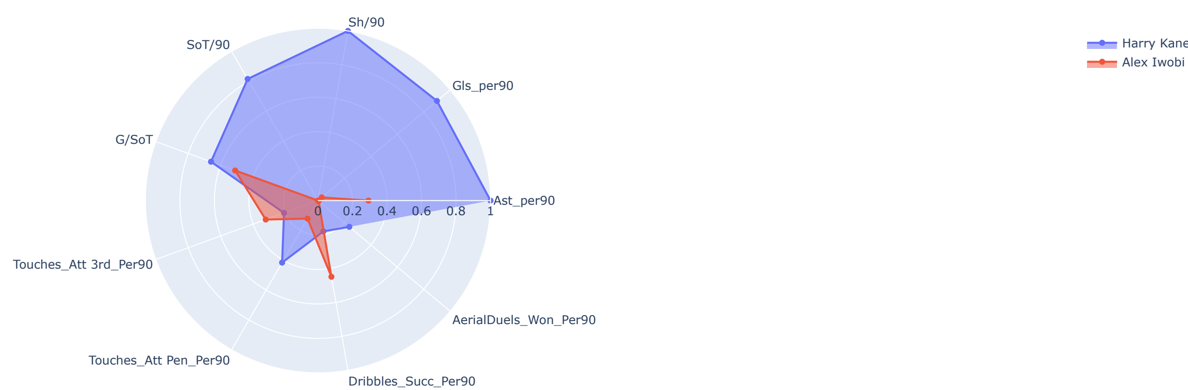
Danny Ings

Harry Kane vs Danny Ings



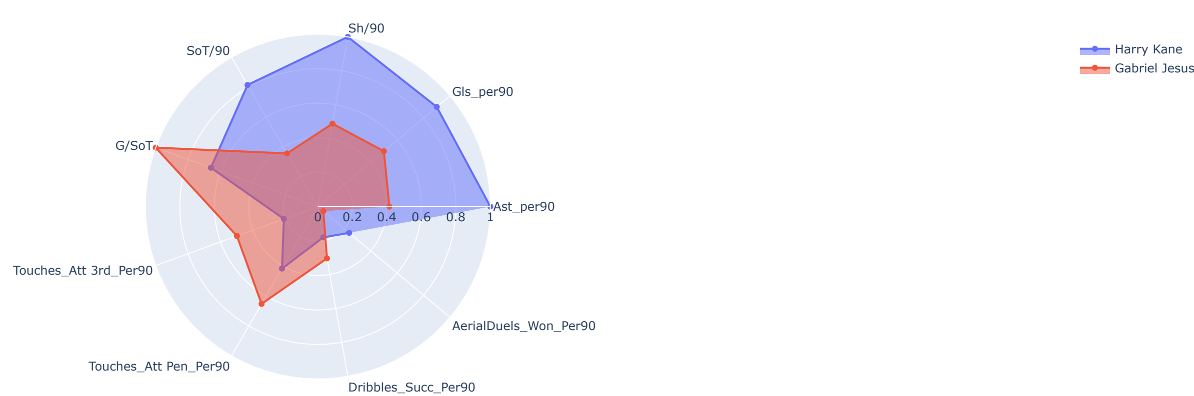
Alex Iwobi

Harry Kane vs Alex Iwobi



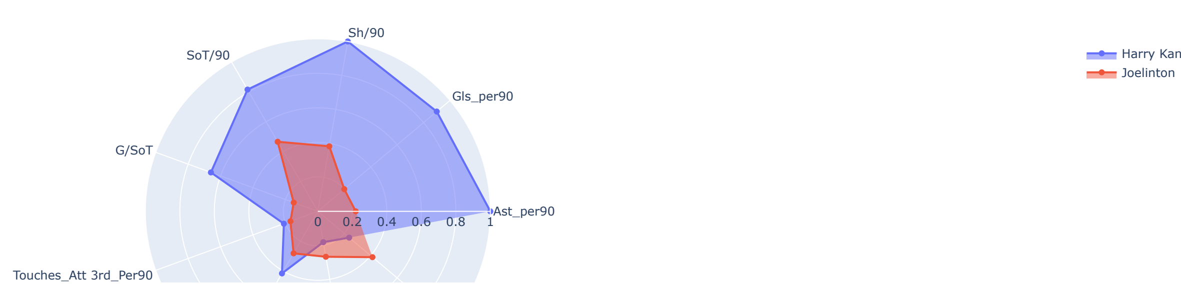
Gabriel Jesus

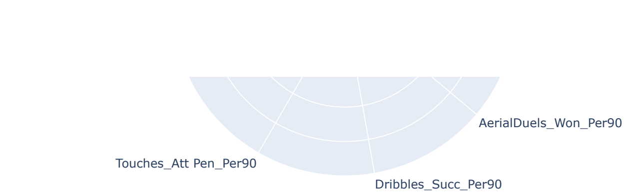
Harry Kane vs Gabriel Jesus



Joelinton

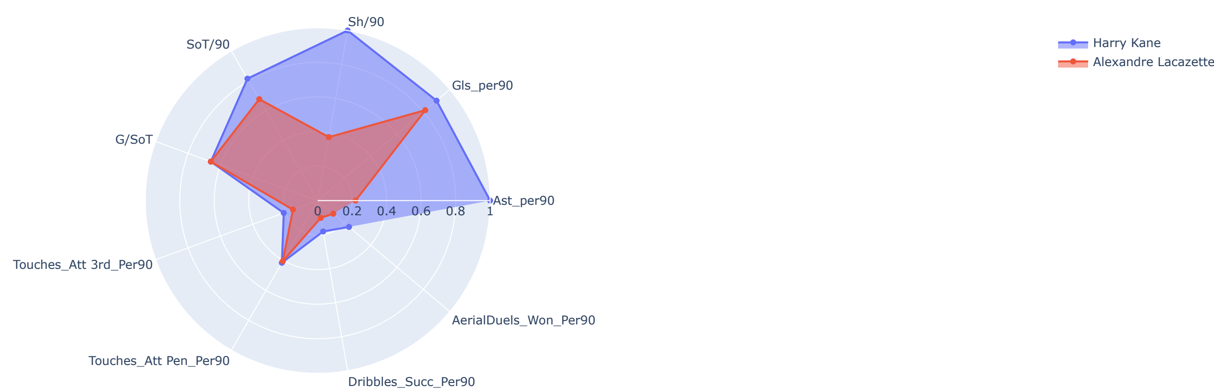
Harry Kane vs Joelinton





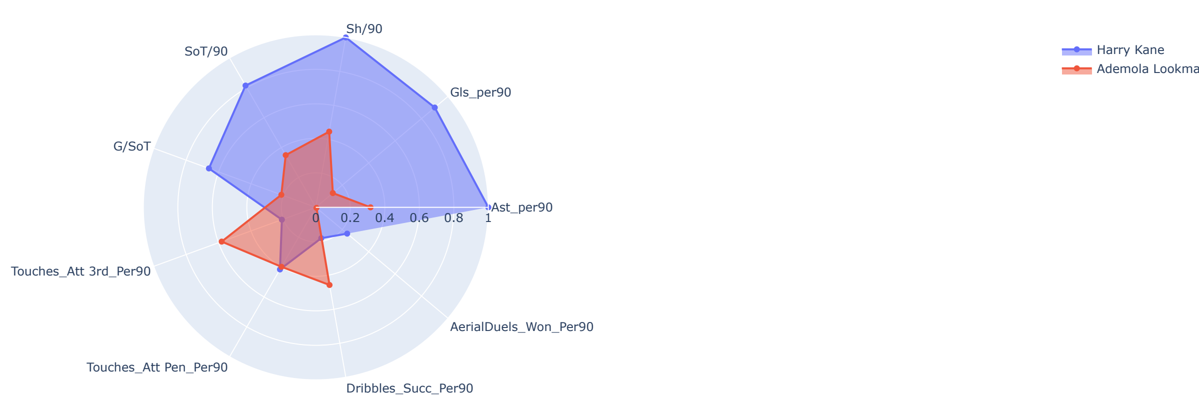
Alexandre Lacazette

Harry Kane vs Alexandre Lacazette



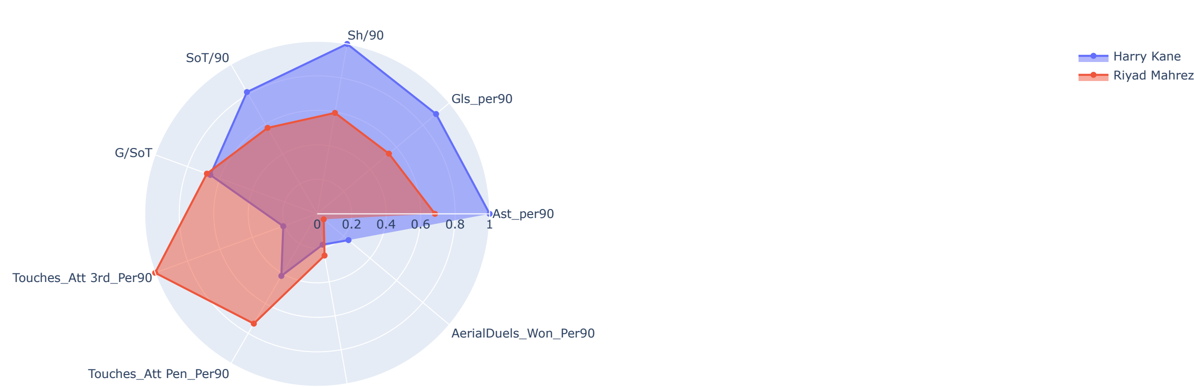
Ademola Lookman

Harry Kane vs Ademola Lookman



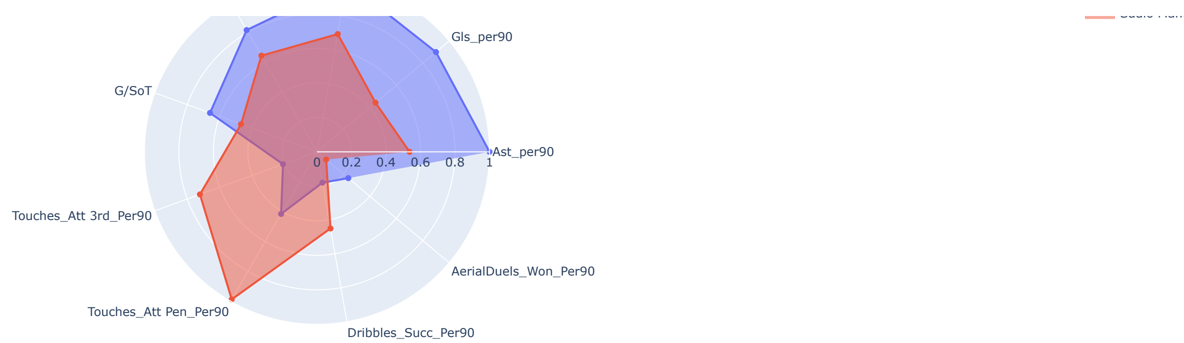
Riyad Mahrez

Harry Kane vs Riyad Mahrez



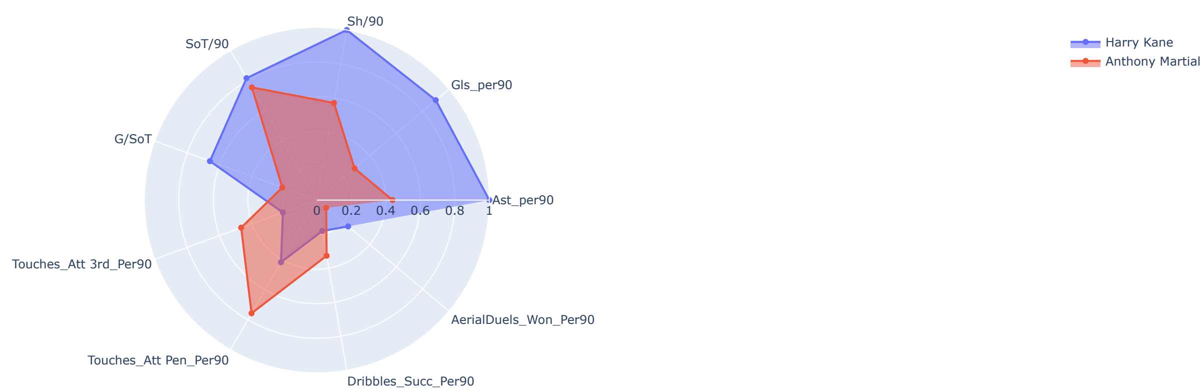
Sadio Mané

Harry Kane vs Sadio Mané



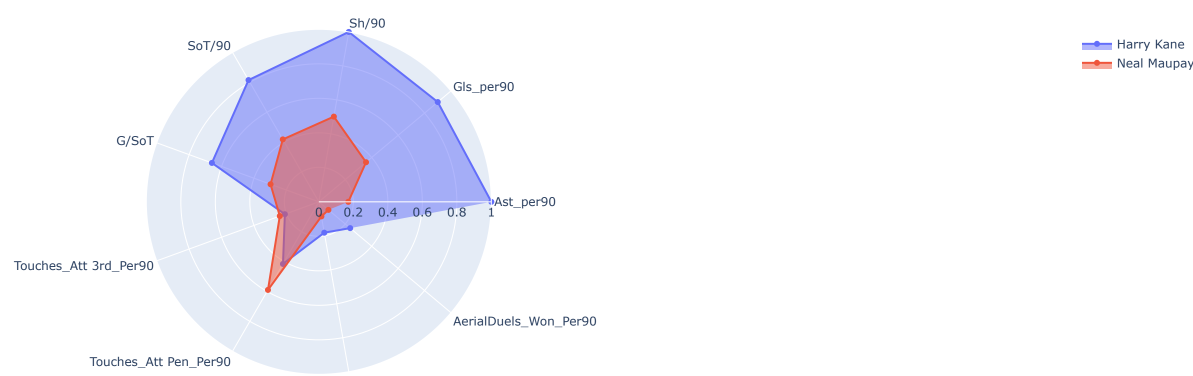
Anthony Martial

Harry Kane vs Anthony Martial



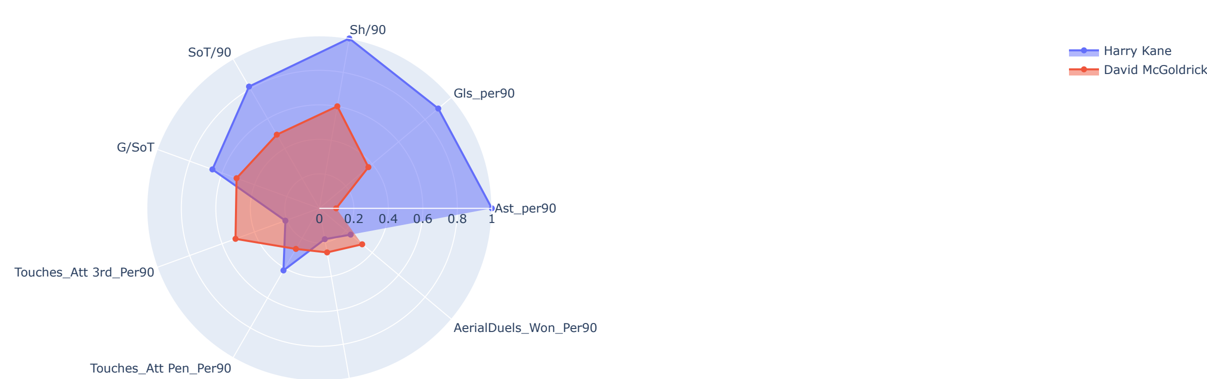
Neal Maupay

Harry Kane vs Neal Maupay



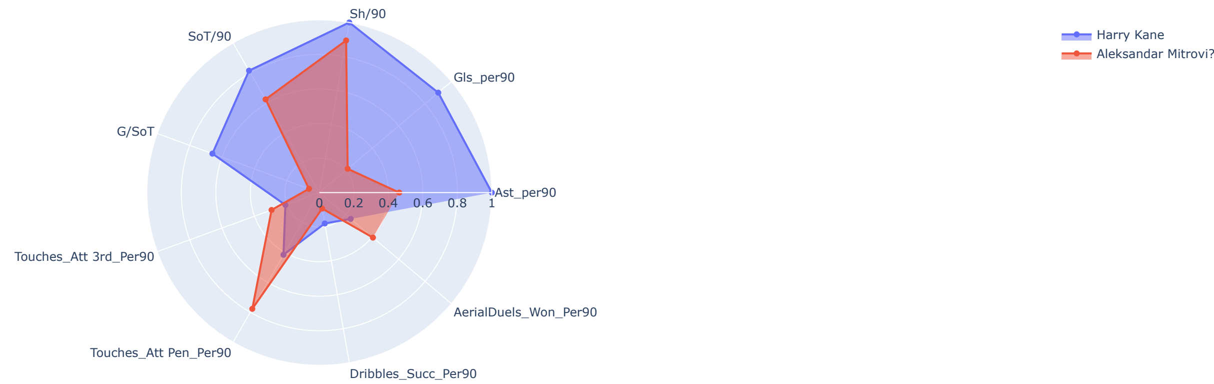
David McGoldrick

Harry Kane vs David McGoldrick



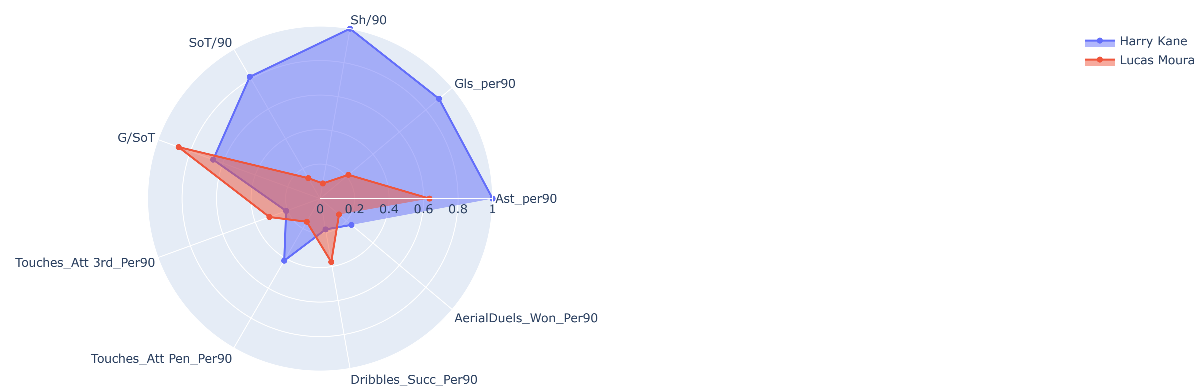
Aleksandar Mitrovi?

Harry Kane vs Aleksandar Mitrovi?



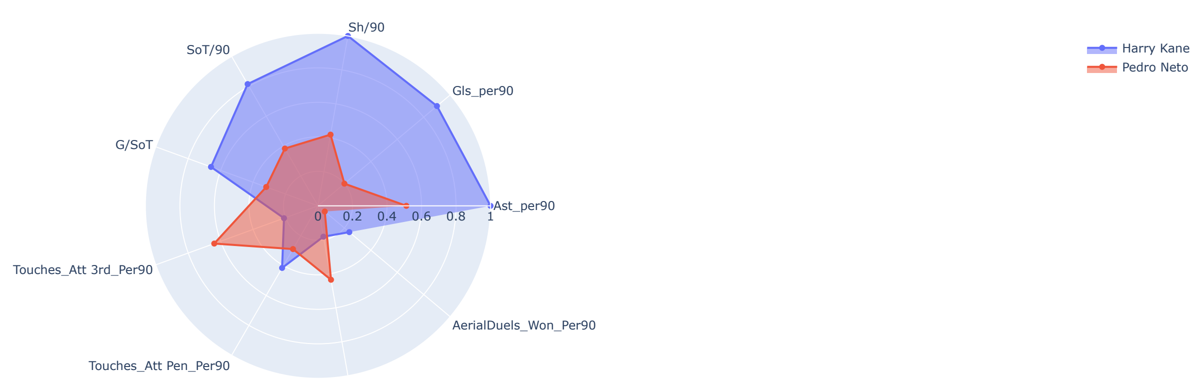
Lucas Moura

Harry Kane vs Lucas Moura



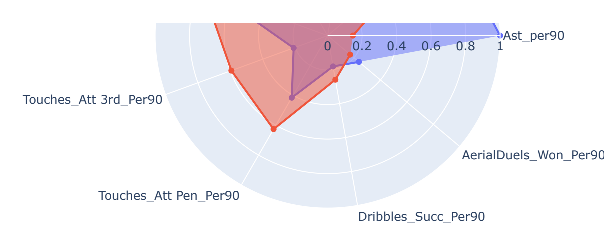
Pedro Neto

Harry Kane vs Pedro Neto



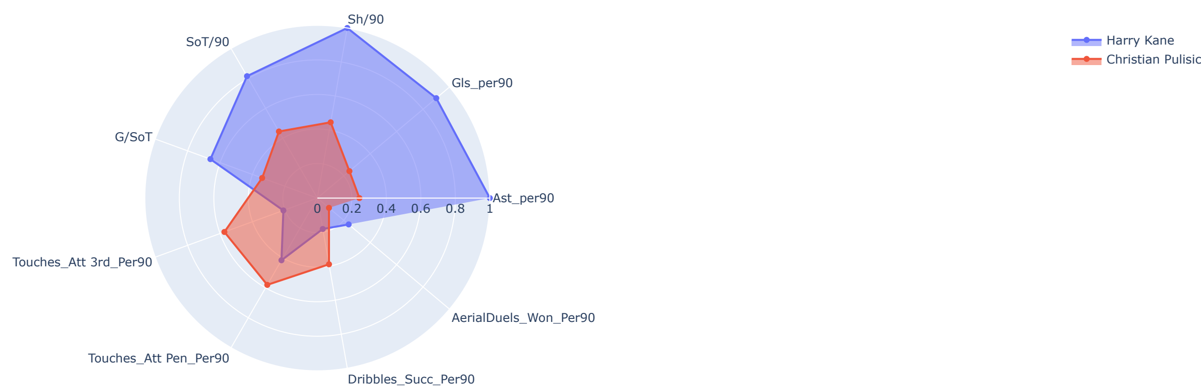
Nicolas Pépé

Harry Kane vs Nicolas Pépé



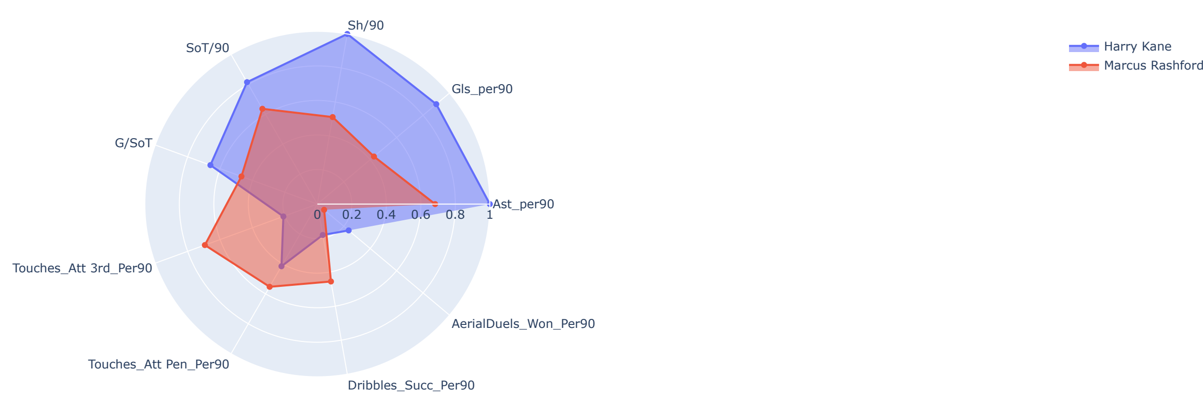
Christian Pulisic

Harry Kane vs Christian Pulisic



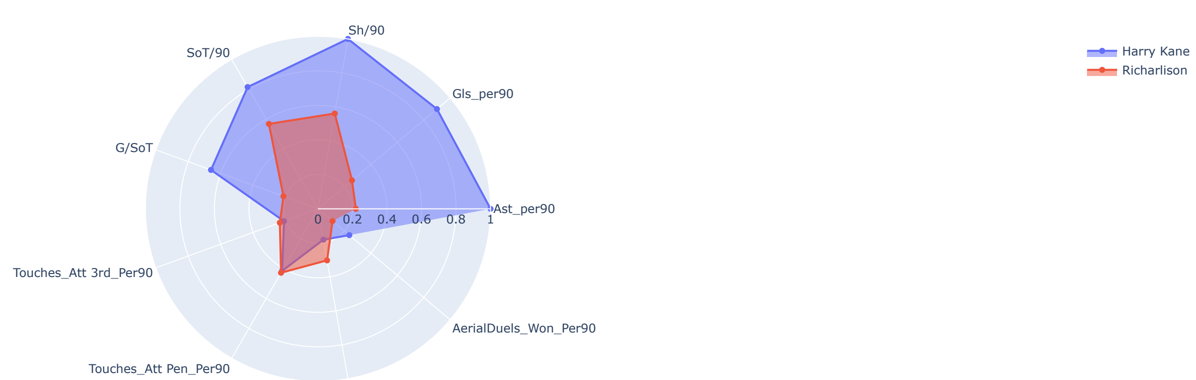
Marcus Rashford

Harry Kane vs Marcus Rashford



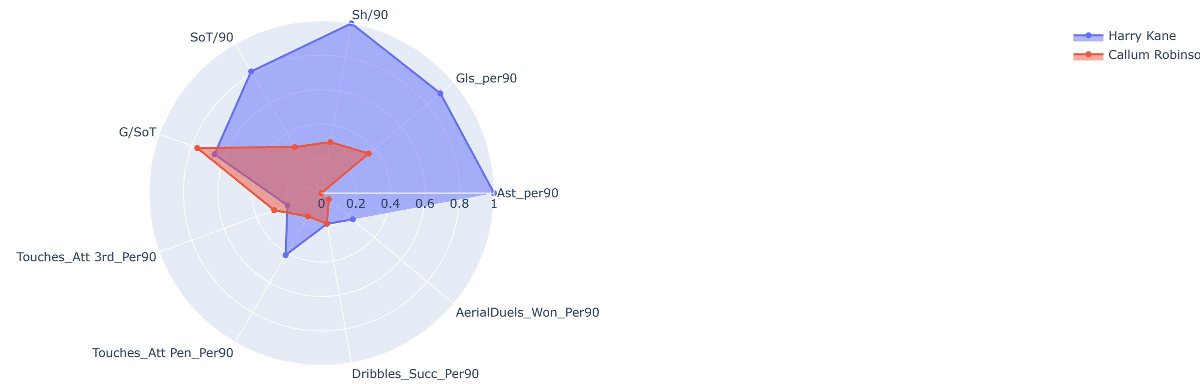
Richarlison

Harry Kane vs Richarlison



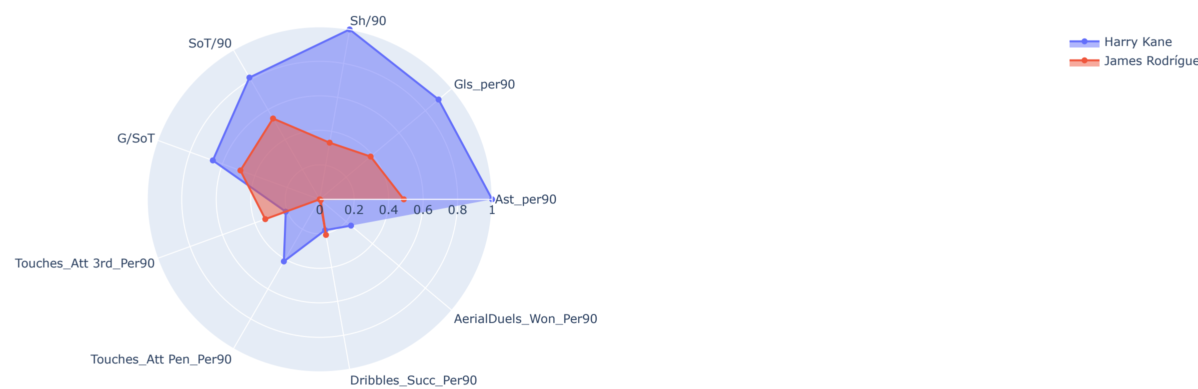
Callum Robinson

Harry Kane vs Callum Robinson



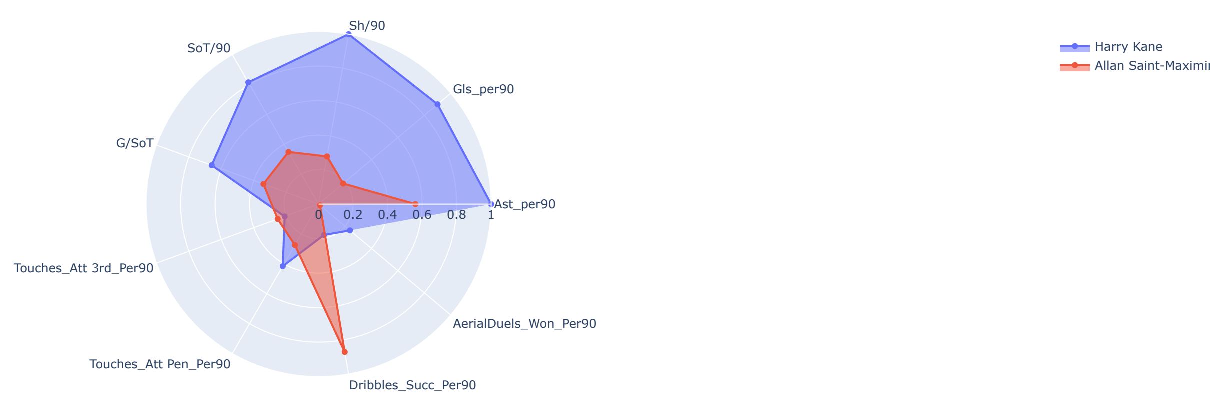
James Rodríguez

Harry Kane vs James Rodríguez



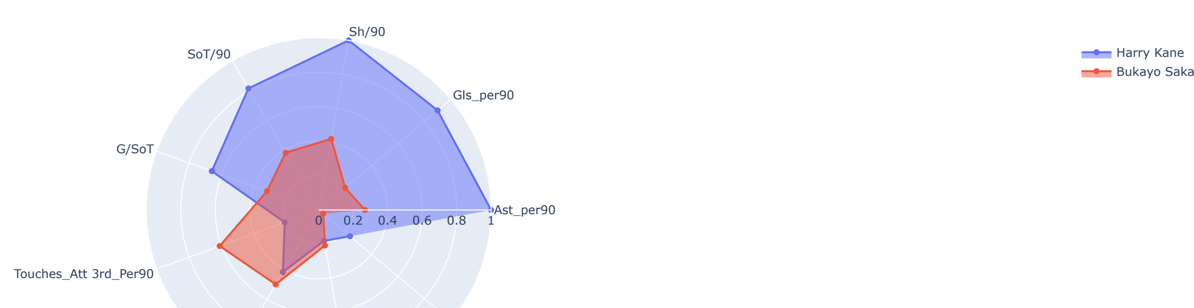
Allan Saint-Maximin

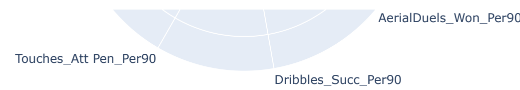
Harry Kane vs Allan Saint-Maximin



Bukayo Saka

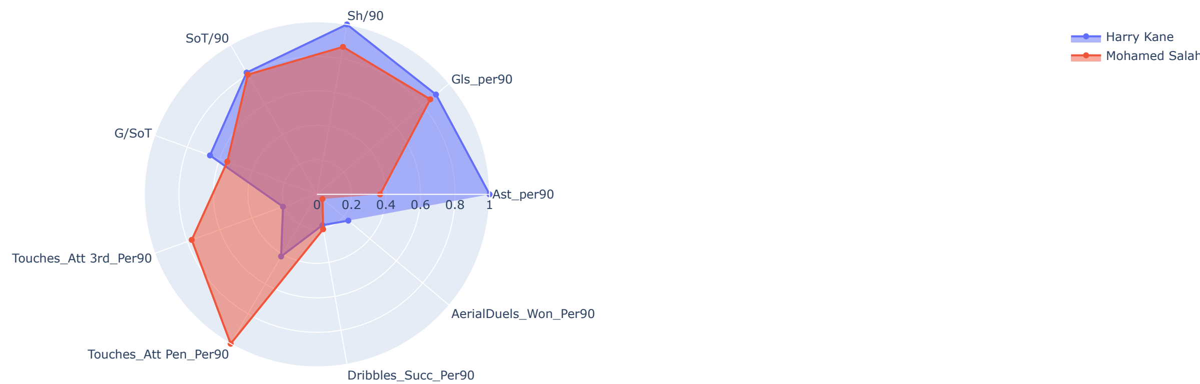
Harry Kane vs Bukayo Saka





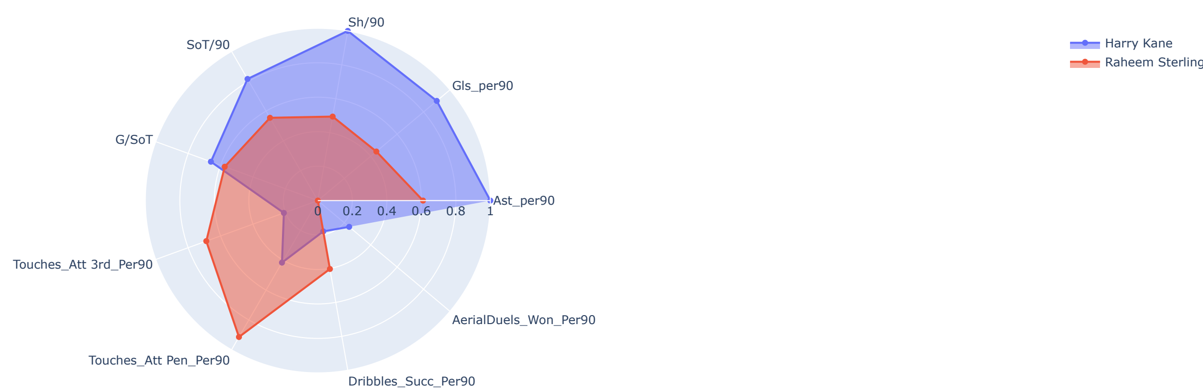
Mohamed Salah

Harry Kane vs Mohamed Salah



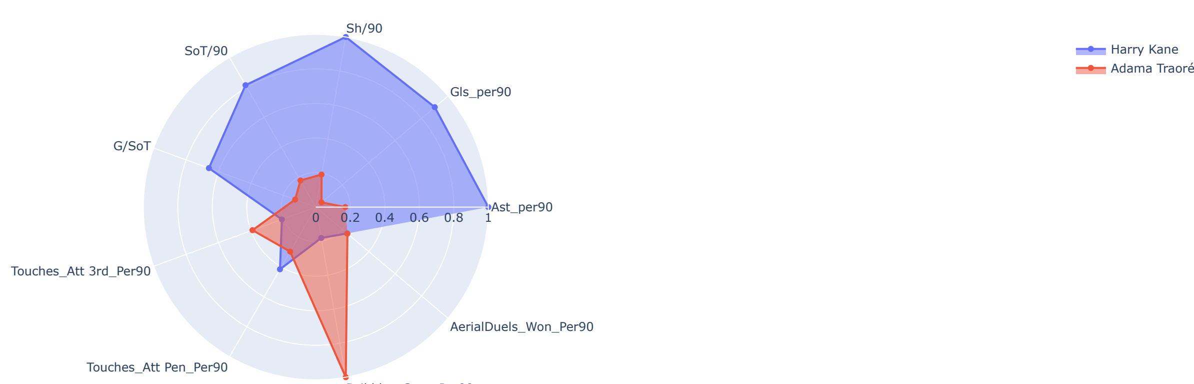
Raheem Sterling

Harry Kane vs Raheem Sterling



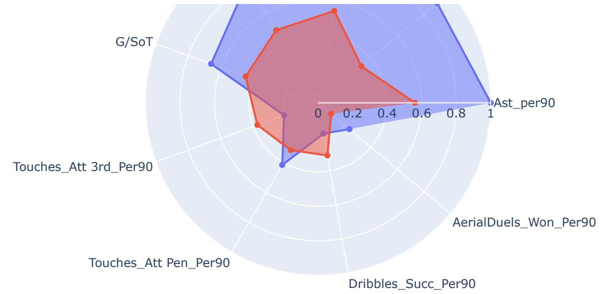
Adama Traoré

Harry Kane vs Adama Traoré



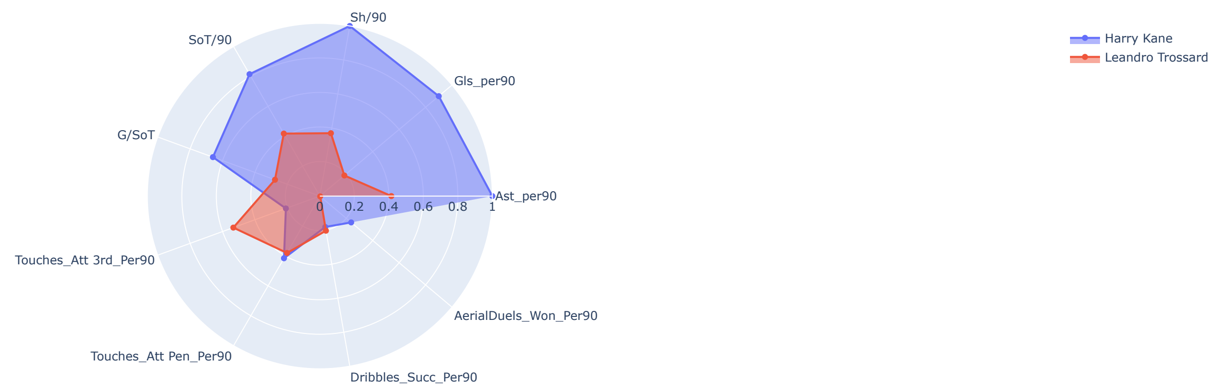
Bertrand Traoré

Harry Kane vs Bertrand Traoré



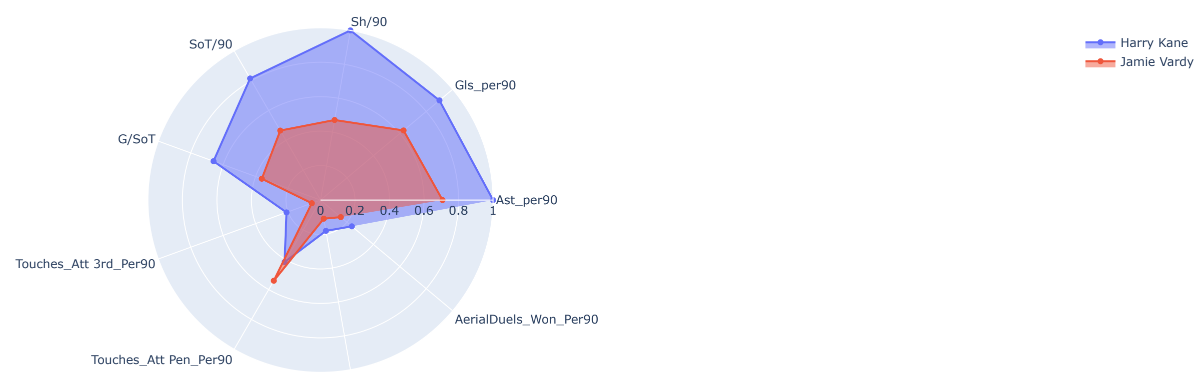
Leandro Trossard

Harry Kane vs Leandro Trossard



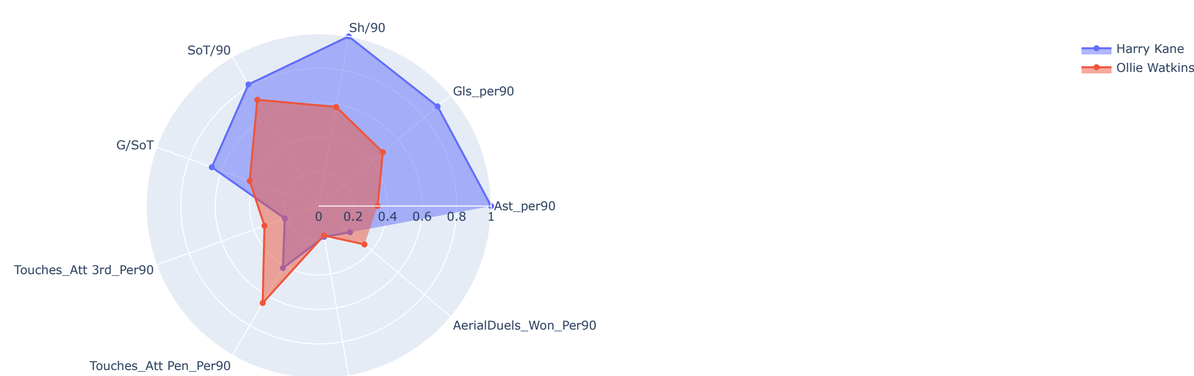
Jamie Vardy

Harry Kane vs Jamie Vardy



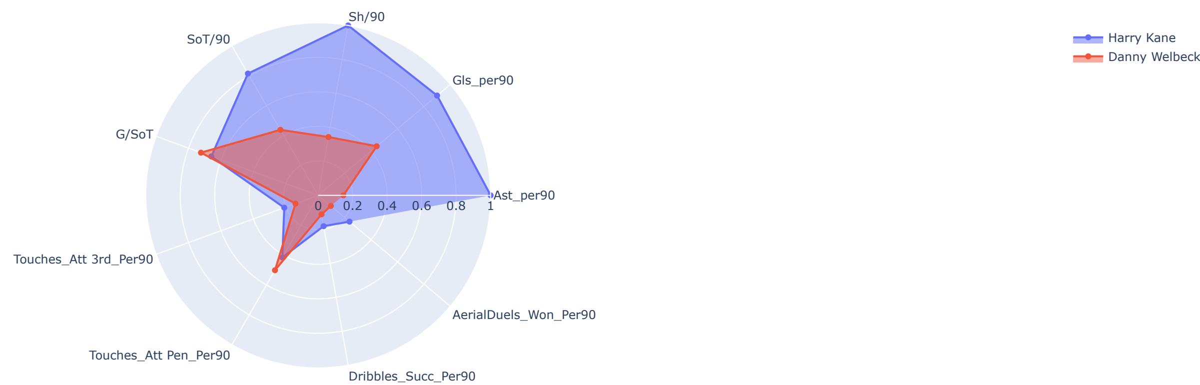
Ollie Watkins

Harry Kane vs Ollie Watkins



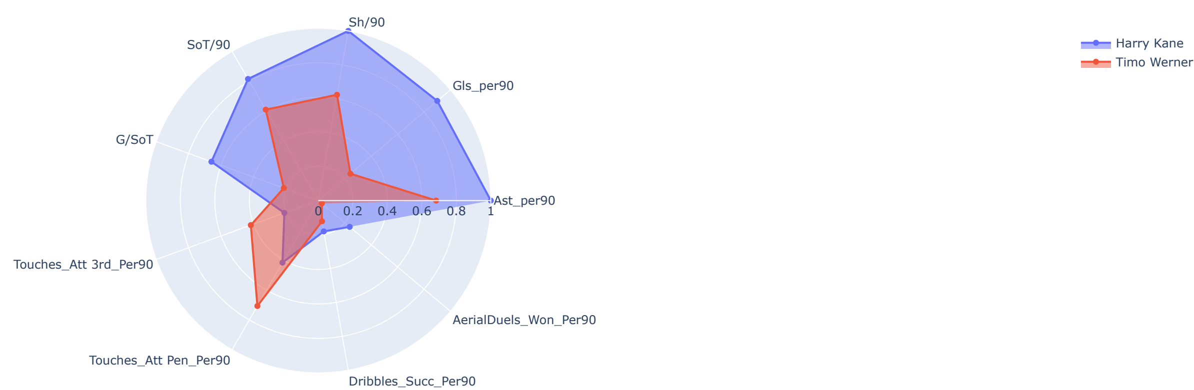
Danny Welbeck

Harry Kane vs Danny Welbeck



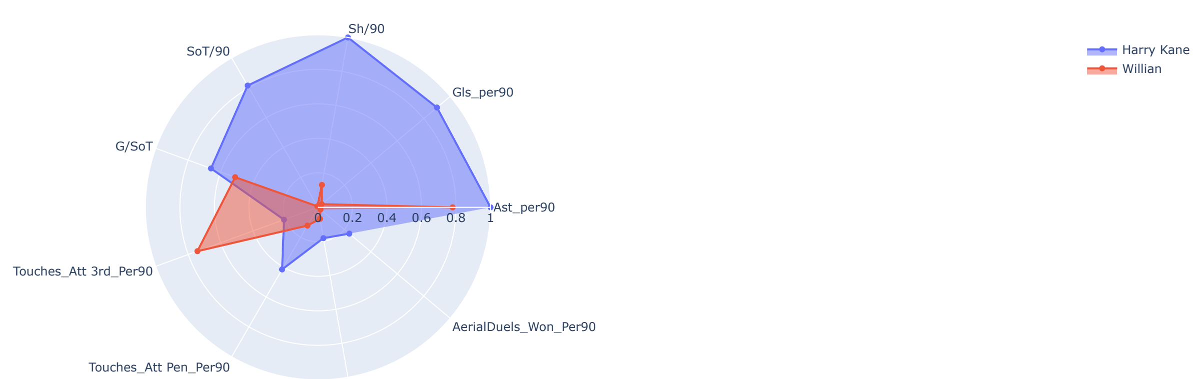
Timo Werner

Harry Kane vs Timo Werner



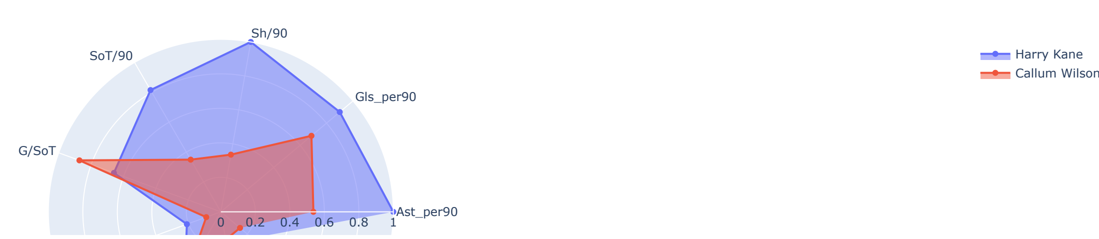
Willian

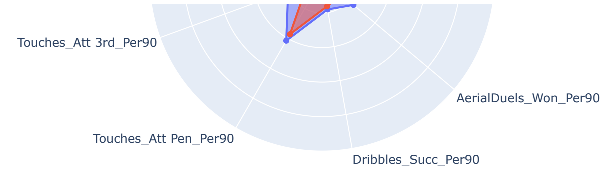
Harry Kane vs Willian



Callum Wilson

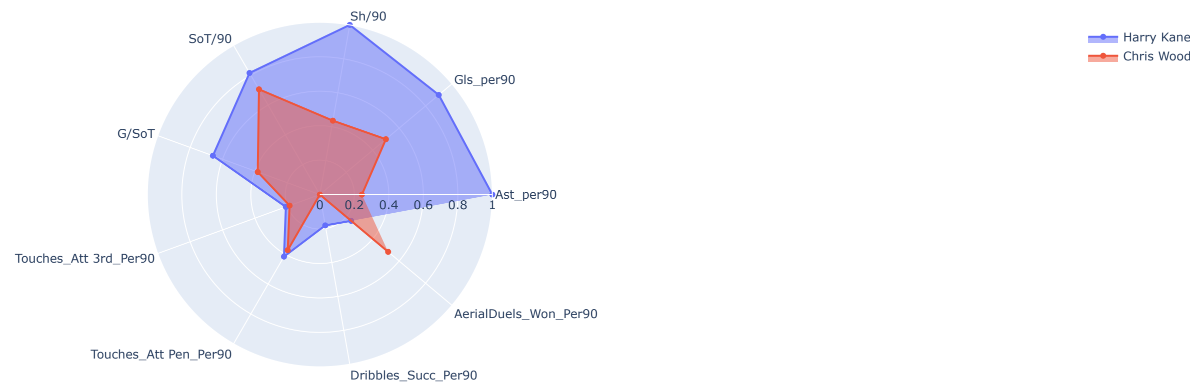
Harry Kane vs Callum Wilson





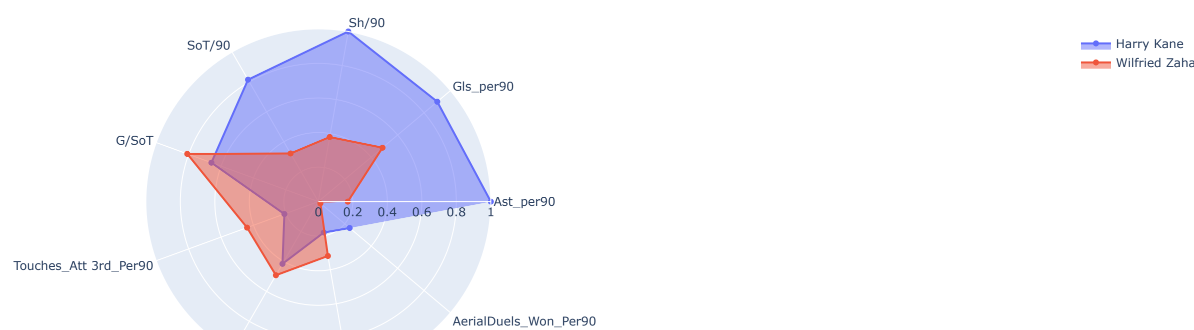

```
Chris Wood
```

Harry Kane vs Chris Wood




```
Wilfried Zaha
```

Harry Kane vs Wilfried Zaha



2) METHOD II: PLAYER RECOMMENDATION USING KNN TECHNIQUE

```
#slicing dataframe to obtain required attribute columns.
attributes = final_data.iloc[:, [4,7,8,9,10,12]]
att_upd = final_data.iloc[:, 17:21]
attributes = pd.concat([attributes, att_upd], axis=1)
data_epl=attributes
attributes=attributes.dropna()
data_epl['Player'] = final_data['Player']
data_epl=data_epl.dropna()
print(attributes.columns)
```

```
Index(['Min', 'Ast_per90', 'Gls_per90', 'Sh/90', 'SoT/90', 'G/SoT',
       'Touches_Att 3rd_Per90', 'Touches_Att Pen_Per90', 'Dribbles_Succ_Per90',
       'AerialDuels_Won_Per90'],
      dtype='object')
```

```
#fit the dataset
scaled=MinMaxScaler()
X = scaled.fit_transform(attributes)
```

```
#Obtain similar players to Harry Kane using KNN
from sklearn.neighbors import NearestNeighbors
recommendations = NearestNeighbors(n_neighbors=6, algorithm='ball_tree').fit(X)
```

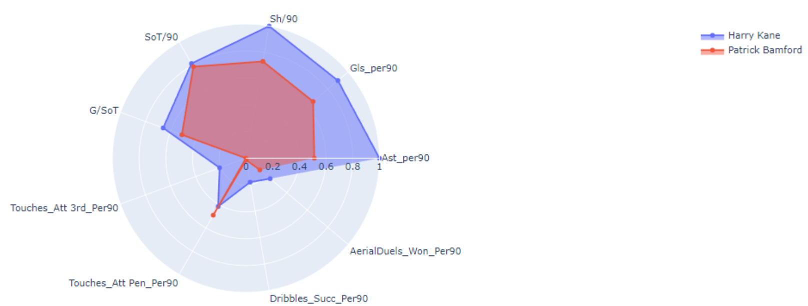
KNN model Output

```
player_indices = recommendations.kneighbors(X)[1]

def get_index(x):
    return data_epl[data_epl['Player']==x].index.tolist()[0]

def recommend(playername):
```

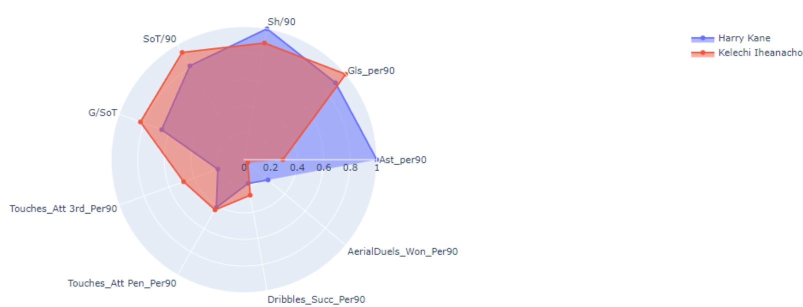
```
    print('Here are 5 players similar to', playername, ':' '\n')
    index = get_index(playername)
    for i in player_indices[index][1:]:
            print(data_epl.iloc[i]['Player'], '\n')

recommend("Harry Kane")
```

```
Here are 5 players similar to Harry Kane :

Patrick Bamford

Jamie Vardy

Son Heung-min

Michail Antonio

Ollie Watkins
```

Harry Kane vs Patrick Bamford



Harry Kane vs Kelechi Iheanacho



CONCLUSION

- From the graphical approach,Patrick Bamford shows similar radar plot shape but lesser statistics to Harry Kane,which implies that if the former increases his stats per game,will show similar attributes to Harry Kane.
- Using the KNN technique,Patrick Bamford is the most similar player to Harry Kane considering only the 2020/21 season of the Premier League.

Interesting Fact:

> From the Radar Plot comparisons of the 2020/21 season, Kelechi Iheanacho showed slightly better stats per game than Harry Kane in most attributes considered but with lesser assists per match and aerial duels won per match.

To conclude from the analysis using both the approaches, Patrick Bamford is the next Harry Kane.