FIFA 20 EXPLORATORY DATA ANALYSIS

O P JOY JEFFERSON, R SHAILESH, RAMIT BATHULA, RAZIK FATIN SHARIFF

ABSTRACT: We have carried out exploratory data analysis on the FIFA 20 player database making useful inferences and predictions.

The graphs we used for useful visualizations are namely pie-chart , count plots, bar graph , box plot , co-relation heatmap. We have also carried out hypothesis testing, t-test hypothesis namely.

For carrying out predictions we have used K Nearest Neighbours predefined model(algorithm='kdtree') from sklearn package. The trained model was able to predict similar players with maximum accuracy, given a player as an input.

Normalization and standardization were performed on the numeric data and plotted corresponding graphs.

INTRODUCTION: Football is a game played on a field between two teams of 11 players each with the object to propel a round ball into the opponent's goal by kicking or by hitting it with any part of the body except the hands and arms.

FIFA 20 is a football simulation video game published by Electronic Arts as part of the FIFA series. It is the 27th instalment in the FIFA series, and was released on 27 September 2019 for Microsoft Windows, PlayStation 4, Xbox One, and Nintendo Switch.

We have carried out interesting club analysis focused on Liverpool inferring useful and meaningful results.

We have also built a model that recommends five similar players to the given player.

```
In [2]: df=pd.read_csv("C:/Users/joyje/OneDrive/Desktop/players_20.csv")

In [3]: rows,col =df.shape
        print("Number of Rows in the dataset: ",rows)
        print("Number of Cols in the dataset: ",col)

        Number of Rows in the dataset:  18278
        Number of Cols in the dataset:  104

In [4]: df.head(10)
Out[4]:
```

_url	short_name	long_name	age	dob	height_cm	weight_kg	nationality	club	...	lwb	ldm	cdm	rdm	rwb	lb	lcb	cb	rcb	rb
om 023 si/...	L. Messi	Lionel Andrés Messi Cuccittini	32	24-06-1987	170	72	Argentina	FC Barcelona	...	68+2	66+2	66+2	66+2	68+2	63+2	52+2	52+2	52+2	63+2
om 1/c- s-...	Cristiano Ronaldo	Cristiano Ronaldo dos Santos Aveiro	34	05-02-1985	187	83	Portugal	Juventus	...	65+3	61+3	61+3	61+3	65+3	61+3	53+3	53+3	53+3	61+3
om 871 sil...	Neymar Jr	Neymar da Silva Santos Junior	27	05-02-1992	175	68	Brazil	Paris Saint-Germain	...	66+3	61+3	61+3	61+3	66+3	61+3	46+3	46+3	46+3	61+3
om 389 0/...	J. Oblak	Jan Oblak	26	07-01-1993	188	87	Slovenia	Atlético Madrid	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
om 277 len- /2...	E. Hazard	Eden Hazard	28	07-01-1991	175	74	Belgium	Real Madrid	...	66+3	63+3	63+3	63+3	66+3	61+3	49+3	49+3	49+3	61+3
om 985 uy...	K. De Bruyne	Kevin De Bruyne	28	28-06-1991	181	70	Belgium	Manchester City	...	77+3	77+3	77+3	77+3	77+3	73+3	66+3	66+3	66+3	73+3
om 448 te...	M. ter Stegen	Marc-André ter Stegen	27	30-04-1992	187	85	Germany	FC Barcelona	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
om 376 di...	V. van Dijk	Virgil van Dijk	27	08-07-1991	193	92	Netherlands	Liverpool	...	79+3	83+3	83+3	83+3	79+3	81+3	87+3	87+3	87+3	81+3
om 003 /2...	L. Modrić	Luka Modrić	33	09-09-1985	172	66	Croatia	Real Madrid	...	81+3	81+3	81+3	81+3	81+3	79+3	72+3	72+3	72+3	79+3
om 331 ed- ah...	M. Salah	Mohamed Salah Ghaly	27	15-06-1992	175	71	Egypt	Liverpool	...	70+3	67+3	67+3	67+3	70+3	66+3	57+3	57+3	57+3	66+3

Dataset : The datasets provided include the players data for the Career Mode of FIFA 20 ("players_20.csv"). The data allows multiple comparison of various attributes of a player of the videogame. The dataset was taken from Kaggle.

Data has been scraped from the publicly available website https://sofifa.com. The various features of the dataset are the player name, respective country, attributes of the particular player(i.e. lwb , lw, ldm,lb, rwb, rb , rdm etc.).

Pre-processing and Data cleaning: There were close to 7k NaN values in the dataset. Few columns of less importance were also dropped. The NaN values were imputed to zero or median depending on the attribute types.

After imputation and removing columns with less importance there was zero NaN values. The code snippets for the corresponding cleaning and imputations have been added in the subsequent slides. After cleaning the number of columns was reduced to 97 from 104.

Before cleaning	After cleaning

In [5]: `df.isnull().sum()`
Out[5]:
sofifa_id 0
player_url 0
short_name 0
long_name 0
age 0
...
lb 2036
lcb 2036
cb 2036
rcb 2036
rb 2036
Length: 104, dtype: int64
	In [8]: `cols = ["dribbling", "defending", "physic", "passing", "shooting", "pace"]`
	`for col in cols:`
	`    df[col]=df[col].fillna(df[col].median())`
	`df=df.fillna(0)`
	`df.isnull().sum()`
	Out[8]:
	sofifa_id 0
	player_url 0
	short_name 0
	long_name 0
	age 0
	..
	lb 0
	lcb 0
	cb 0
	rcb 0
	rb 0
	Length: 104, dtype: int64

In [7]:

```
stats = ['ls', 'st', 'rs', 'lw', 'lf', 'cf', 'rf', 'rw', 'lam', 'cam', 'ram',
         'lm', 'lcm', 'cm', 'rcm', 'rm', 'lwb', 'ldm', 'cdm', 'rdm', 'rwb', 'lb',
         'lcb', 'cb', 'rcb', 'rb']
for col in stats:
    new = df[col].str.split("+", n = 1, expand = True)
    df[col] = new[0]

df[stats] = df[stats].fillna(0)
df[stats]=df[stats].astype(int)
df[stats].head(10)
```

Out[7]:

	ls	st	rs	lw	lf	cf	rf	rw	lam	cam	...	lwb	ldm	cdm	rdm	rwb	lb	lcb	cb	rcb	rb
0	89	89	89	93	93	93	93	93	93	93	...	68	66	66	66	68	63	52	52	52	63
1	91	91	91	89	90	90	90	89	88	88	...	65	61	61	61	65	61	53	53	53	61
2	84	84	84	90	89	89	89	90	90	90	...	66	61	61	61	66	61	46	46	46	61
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	83	83	83	89	88	88	88	89	89	89	...	66	63	63	63	66	61	49	49	49	61
5	82	82	82	87	87	87	87	87	88	88	...	77	77	77	77	77	73	66	66	66	73
6	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
7	69	69	69	67	69	69	69	67	69	69	...	79	83	83	83	79	81	87	87	87	81
8	77	77	77	84	83	83	83	84	86	86	...	81	81	81	81	81	79	72	72	72	79
9	84	84	84	88	88	88	88	88	87	87	...	70	67	67	67	70	66	57	57	57	66

10 rows × 26 columns

In [8]:

```
cols = ["dribbling", "defending", "physic", "passing", "shooting", "pace"]
for col in cols:
    df[col]=df[col].fillna(df[col].median())
df=df.fillna(0)
df.isnull().sum()
```

Out[8]:

```
sofifa_id     0
player_url    0
short_name    0
long_name     0
age           0
             ..
lb            0
lcb           0
cb            0
rcb           0
rb            0
Length: 104, dtype: int64
```

Exploratory Data Analysis:

Few interesting insights we obtained from our EDA are:

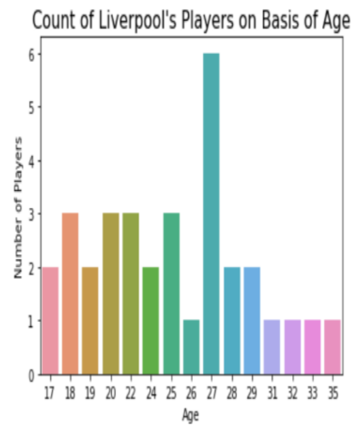
- The average age of the players from team Liverpool was 24.5 which was also inferred from the count plot.
- Team Liverpool had most of it's players from England.
- The overall rating of a player was directly proportional to their weekly wages, i.e. more the overall rating, higher the wage.
- Another interesting inference made was there is no relation between BMI and the player's nationality, which was something usually misunderstood.
- We also inferred that left foot and right foot attributes are two independent groups using boxplot and hypothesis testing.

In [23]:
```
plt.figure()

ax = sns.countplot(x='age', data=liverpool)
ax.set_title(label='Count of Liverpool\'s Players on Basis of Age', fontsize=16)

ax.set_xlabel(xlabel='Age')
ax.set_ylabel(ylabel='Number of Players')
```

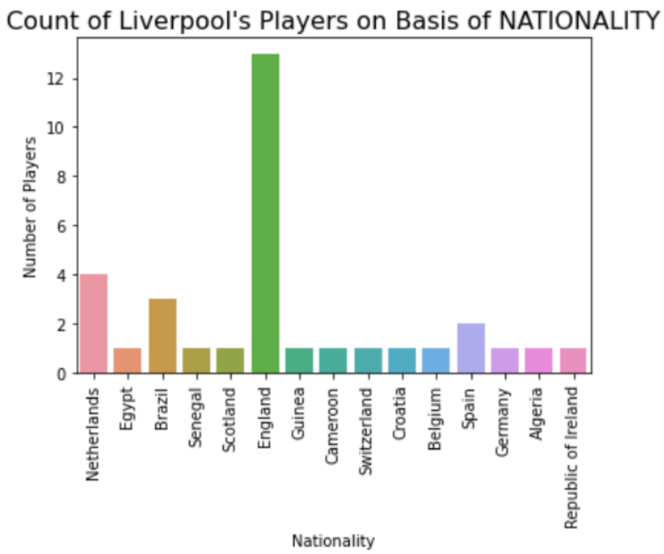
Out[23]: Text(0, 0.5, 'Number of Players')



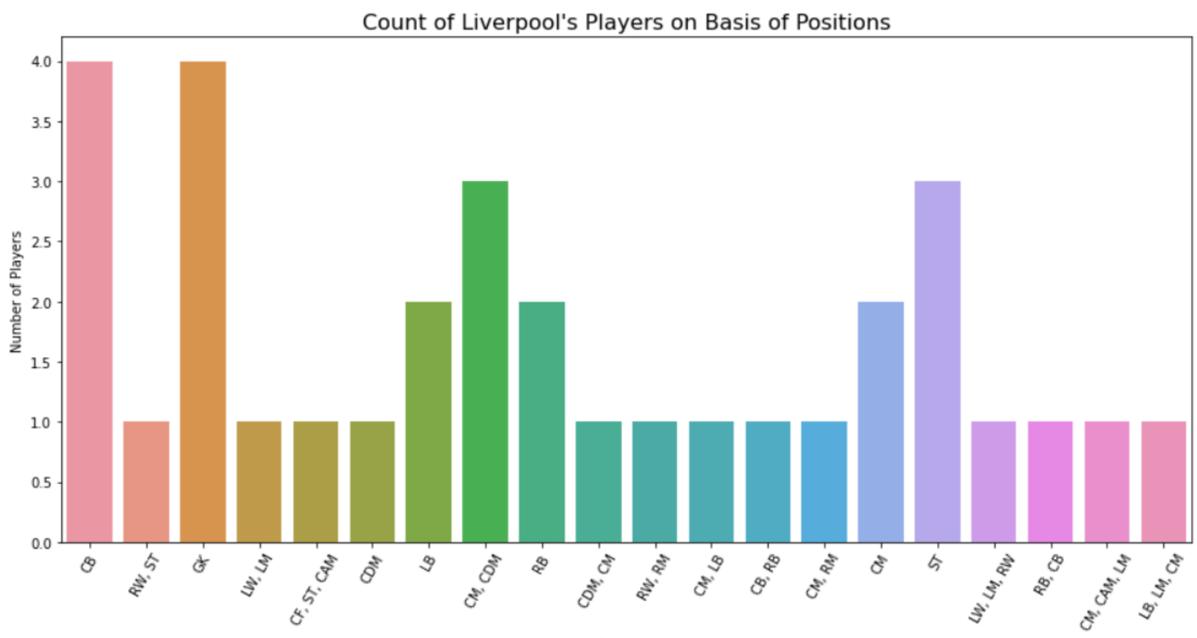
In [24]:
```
avg_age = liverpool["age"].mean()
avg_age = round(avg_age,1)
print('Liverpool Squad\'s have an average age of ',avg_age, ' Years')
```

```
Liverpool Squad's have an average age of  24.5  Years
```

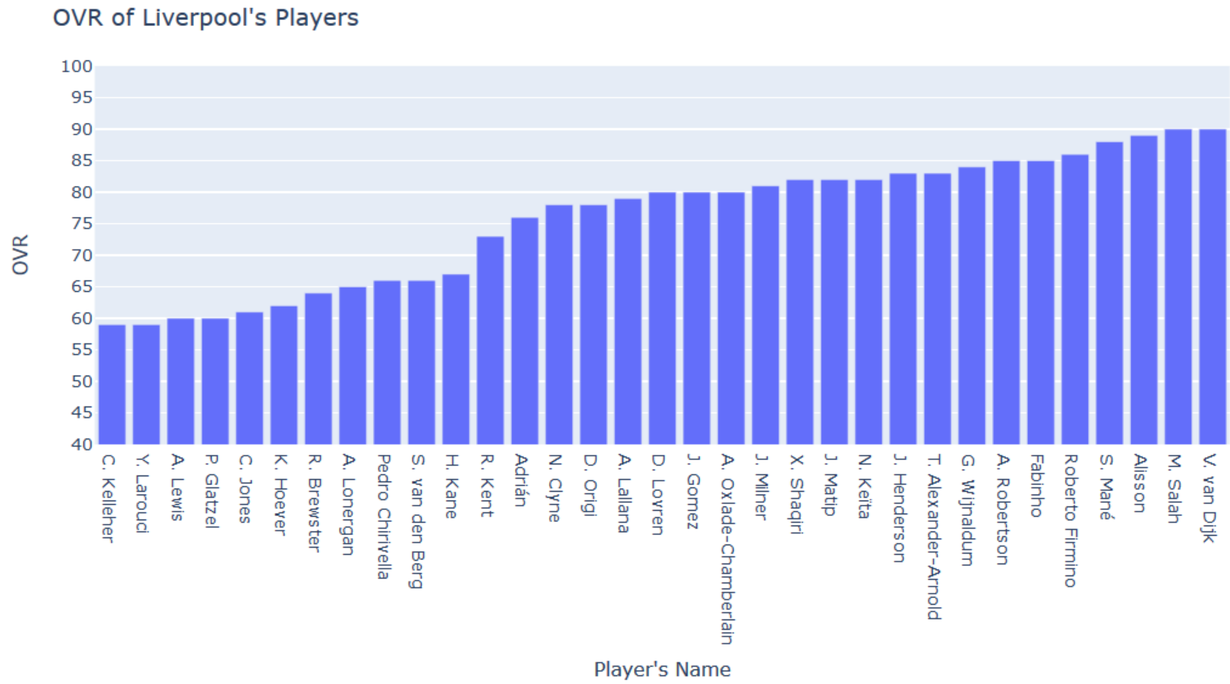
In [25]:
```
plt.figure()
ax = sns.countplot(x='nationality', data=liverpool)
ax.set_title(label='Count of Liverpool\'s Players on Basis of NATIONALITY', fontsize=16)
ax.set_xlabel(xlabel='Nationality')
ax.set_ylabel(ylabel='Number of Players')
plt.xticks(rotation=90)
plt.show()
```



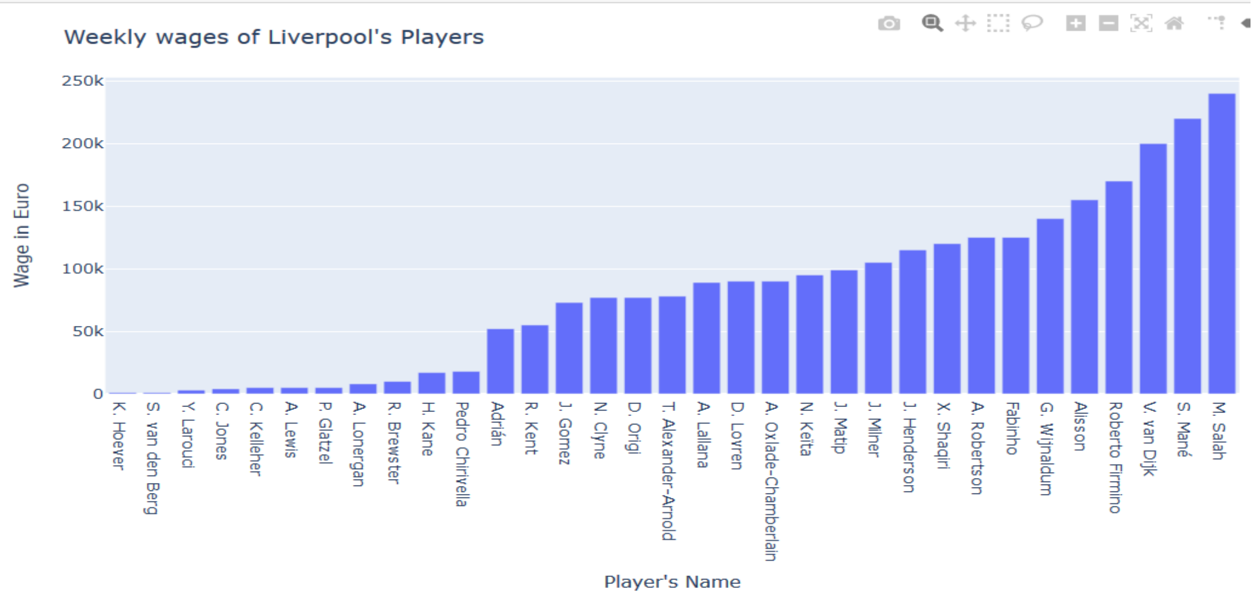
In [26]:
```
plt.figure(figsize= (15,7))
ax = sns.countplot(x='player_positions', data=liverpool)
ax.set_title(label='Count of Liverpool\'s Players on Basis of Positions', fontsize=16)
ax.set_xlabel(xlabel='Positions')
ax.set_ylabel(ylabel='Number of Players')
plt.xticks(rotation=60)
plt.show()
```



```
In [27]: tempdf = liverpool.sort_values(by='overall')
         fig = px.bar(tempdf, x='short_name', y='overall')
         fig['layout']['yaxis1'].update(title='', range=[40, 100], dtick=5, autorange=False)
         fig.update_layout(title='OVR of Liverpool\'s Players',
                           xaxis_title="Player's Name",
                           yaxis_title="OVR")
         py.iplot(fig)
```




```
In [28]: tempdf = liverpool.sort_values(by='wage_eur')
         fig = px.bar(tempdf, x='short_name', y='wage_eur')
         fig.update_layout(title='Weekly wages of Liverpool\'s Players',
                           xaxis_title="Player's Name",
                           yaxis_title="Wage in Euro")
         py.iplot(fig)
```



```
In [29]: df['bmi'] = df['weight_kg'] / (df['height_cm']/100)**2
```

```
In [30]: fig = go.Figure()
sample = df.sort_values(by='nationality')
fig.add_trace(go.Box(
    x = sample['nationality'],
    y = sample['bmi'],
    name="Suspected Outliers",
    boxpoints='suspectedoutliers',
    marker=dict(
        size=12,
        color='rgb(180, 222, 43)',
        outliercolor='rgba(31, 158, 137, 0.6)',
        line=dict(
            outliercolor='rgba(31, 158, 137, 0.6)',
            outlierwidth=2)),
    line_color='rgb(72, 40, 120)',
    text= sample['short_name']
))
fig.update_layout(title='Box Plot - Nationality vs BMI',
                  xaxis_title='Nationality',
                  yaxis_title='BMI',
                  paper_bgcolor='rgba(0,0,0,0)',
                  plot_bgcolor='rgba(0,0,0,0)',
                  font=dict(family='Cambria, monospace', size=12, color='#000000'),
                  xaxis_rangeslider_visible=True)
fig.show()
```

Lime : Confirmed Outliers

Green : Suspected Outliers

Hypothesis Testing : A t-test is a type of inferential statistic used to determine if there is a significant difference between the means of two groups, which may be related in certain features.

Two Sample independent t-test is used to compare the means of two independent groups. For example, we have two different playing foots (Left foot and Right foot) and would like to compare if the overall rating of left-foot is significantly different from right-foot.

- Null hypotheses: Two group means are equal
- Alternative hypotheses: Two group means are different (two-tailed)


```
In [32]: from scipy import stats
         a = df[df['preferred_foot']=='Left'][['skill_ball_control','power_shot_power', 'attacking_finishing']]
         b = df[df['preferred_foot']=='Right'][['skill_ball_control','power_shot_power', 'attacking_finishing']]
```

```
In [54]: new_a = a['skill_ball_control'].sample(n=30)
         new_b = b['skill_ball_control'].sample(n=30)
         t1, p1 = stats.ttest_ind(new_a,new_b, equal_var=False)
         print("t = " + str(t1))
         print("p = " + str(p1))
         if p1 <0.05:
           print("we reject null hypothesis")
         else:
           print("we accept null hypothesis")

         t = -0.32771421072049556
         p = 0.7444670991417686
         we accept null hypothesis
```

```
In [34]: data = [new_a, new_b]
         fig = plt.figure(figsize =(10, 7))
         ax = fig.add_subplot(111)


         bp = ax.boxplot(data, patch_artist = True,
                         notch ='True', vert = 0)


         colors = ['#0000FF', '#00FF00',
                   '#FFFF00', '#FF00FF']
```

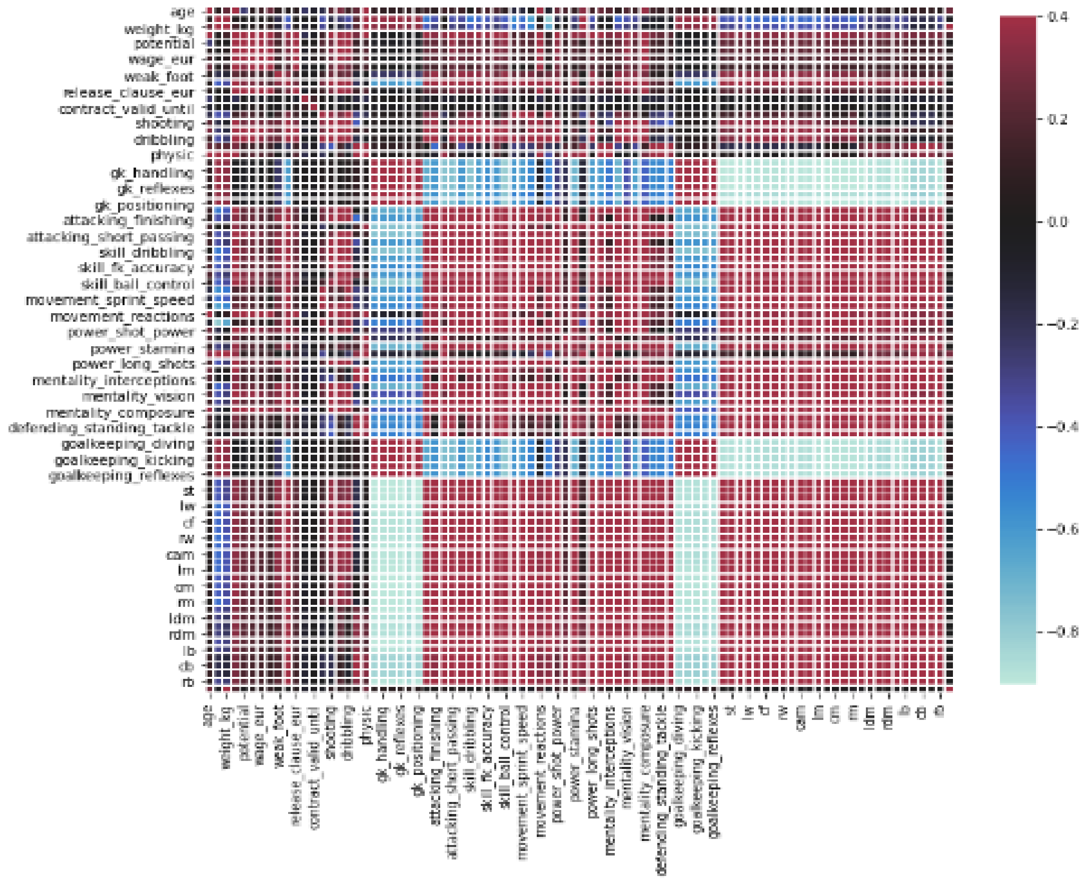
t-test is performed when sample size is less than equal to 30 and for non-Gaussian distributions.

Here in our case we reject null hypothesis.

Correlation matrix : Is a table showing correlation coefficients between variable(features of the dataset). Each cell in the matrix shows the correlation between two variables. A correlation matrix is used to summarize data, as an input into a more advanced analysis, and as a diagnostic for advanced analysis.

```
In [34]: plt.figure(figsize=(12,12))
         corr = sample.corr()
         sns.heatmap(corr, vmax=.4, center=0,
                     square=True, linewidths=.7, cbar_kws={"shrink": 0.8})

Out[34]: <AxesSubplot:>
```



Predictions : For further useful predictions we used K Nearest Neighbours predefined model from sklearn . We performed unsupervised learning techniques for our models training, after which our model was able predict or recommend five similar players given a player as an input.

```
In [37]: scaled = StandardScaler()
         X = scaled.fit_transform(sample)
         recommendations = NearestNeighbors(n_neighbors=7,algorithm='kd_tree')
         recommendations.fit(X)
         player_index = recommendations.kneighbors(X)[1]
```

```
In [60]: def player_name(x):
             return df[df['short_name']==x].index.tolist()[0]



         def recommend_similar(player):
             print("These are 5 players similar to {} : ".format(player))
             index= player_name(player)
             for i in player_index[index][1:]:
                 print("Name: {}\nOverall: {}\nMarket Value: €{}\nAge: {}\nBMI: {:.2f}\n".format(df.iloc[i]['short_name'],df.iloc
```

```
In [61]: recommend_similar('E. Hazard')
```

```
These are 5 players similar to E. Hazard :
Name: L. Messi
Overall: 94
Market Value: €95500000
Age: 32
BMI: 24.91

Name: K. De Bruyne
Overall: 91
Market Value: €90000000
Age: 28
BMI: 21.37

Name: A. Griezmann
Overall: 89
Market Value: €69000000
Age: 28
BMI: 23.57

Name: Neymar Jr
Overall: 92
Market Value: €105500000
Age: 27
BMI: 22.20

Name: M. Salah
Overall: 90
Market Value: €80500000
Age: 27
BMI: 23.18
```

Results and discussion: The following conclusions were made from performing EDA on the dataset:

1) We could infer the average of age of all players of a particular team.

2) We came to the conclusion that higher the rating of the player higher is his weekly salary.

3) We also saw that most of the players from Liverpool were from England.

4) We made an assumption that there is a relationship between bmi and nationality of a player, but that was proven wrong from the box plot that was plotted

5) We have normalized and standardized our data to obtain 0 mean and 1 variance.

6) We could also recommend 5 players of similar attributes given a player.

Possible analysis with our dataset:

- Historical comparison between Messi and Ronaldo (what skill attributes changed the most during time - compared to real-life stats);
- Ideal budget to create a competitive team (at the level of top n teams in Europe) and at which point the budget does not allow to buy significantly better players for the 11-men line-up. An extra is the same comparison with the Potential attribute for the line-up instead of the Overall attribute;
- Sample analysis of top n% players (e.g. top 5% of the player) to see if some important attributes as

Agility or BallControl or Strength have been popular or not across the FIFA versions. An example would be seeing that the top 5% players of FIFA 20 are more fast (higher Acceleration and Agility) compared to FIFA 15. The trend of attributes is also an important indication of how some attributes are necessary for players to win games (a version with more top 5% players with high BallControl stats would indicate that the game is more focused on the technique rather than the physical aspect).

O P JOY JEFFERSON : PES2UG19CS270 E

R SHAILESH : PES2UG19CS307 E

RAMIT BATHULA : PES2UG19CS319 E

RAZIK FATIN SHARIFF : PES2UG19CS323 E

THANK YOU