DAV Mini Project 2

Give your Team Details

Group Number: 4

Team members:

1. Dinesh Subramanian
2. S.Aishwarya
3. Vinod A

Data Preprocessing:

1. Import the necessary libraries and read the data. (1 marks)

In [238]:
```python
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
import warnings
warnings.filterwarnings("ignore")
pd.set_option('display.max_columns',None)
pd.set_option('display.max_rows',None)
```

In [239]:
```
df_fifa = pd.read_csv('fifa.csv')
df_fifa_variable_info = pd.read_csv('fifa_ variable_information.csv')
df_fifa_variable_info
```

Out[239]:

	ID	unique id for every player
0	Name	name
1	Age	age
2	Photo	url to the player's photo
3	Nationality	nationality
4	Flag	url to players's country flag
5	Overall	overall rating
6	Potential	potential rating
7	Club	current club
8	Club Logo	url to club logo
9	Value	current market value
10	Wage	current wage
11	Preferred Foot	left/right
12	International Reputation	rating on scale of 5
13	Weak Foot	rating on scale of 5
14	Skill Moves	rating on scale of 5
15	Work Rate	attack work rate/defence work rate
16	Body Type	body type of player
17	Position	position on the pitch
18	Jersey Number	jersey number
19	Joined	joined date
20	Loaned From	club name if applicable
21	Contract Valid Until	contract end date
22	Height	height of the player
23	Weight	weight of the player
24	Crossing	rating on scale of 100
25	Finishing	rating on scale of 100
26	HeadingAccuracy	rating on scale of 100
27	ShortPassing	rating on scale of 100
28	Volleys	rating on scale of 100
29	Dribbling	rating on scale of 100
30	Curve	rating on scale of 100
31	FKAccuracy	rating on scale of 100
32	LongPassing	rating on scale of 100

	ID	unique id for every player
33	BallControl	rating on scale of 100
34	Acceleration	rating on scale of 100
35	SprintSpeed	rating on scale of 100
36	Agility	rating on scale of 100
37	Reactions	rating on scale of 100\
38	Balance	rating on scale of 100
39	ShotPower	rating on scale of 100
40	Jumping	rating on scale of 100
41	Stamina	rating on scale of 100
42	Strength	rating on scale of 100
43	LongShots	rating on scale of 100
44	Aggression	rating on scale of 100
45	Interceptions	rating on scale of 100
46	Positioning	rating on scale of 100
47	Vision	rating on scale of 100
48	Penalties	rating on scale of 100
49	Composure	rating on scale of 100
50	Marking	rating on scale of 100
51	StandingTackle	rating on scale of 100
52	SlidingTackle	rating on scale of 100
53	GKDiving	rating on scale of 100
54	GKHandling	rating on scale of 100
55	GKKicking	rating on scale of 100
56	GKPositioning	rating on scale of 100
57	GKReflexes	rating on scale of 100
58	Release Clause	release clause value

2. Drop any columns that you deem unnecessary for analysis

In [240]:
```
df_fifa.columns
```

Out[240]:
```
Index(['ID', 'Name', 'Age', 'Photo', 'Nationality', 'Flag', 'Overall',
       'Potential', 'Club', 'Club Logo', 'Value', 'Wage', 'Preferred Foot',
       'International Reputation', 'Weak Foot', 'Skill Moves', 'Work Rate',
       'Body Type', 'Position', 'Jersey Number', 'Joined', 'Loaned From',
       'Contract Valid Until', 'Height', 'Weight', 'Crossing', 'Finishing',
       'HeadingAccuracy', 'ShortPassing', 'Volleys', 'Dribbling', 'Curve',
       'FKAccuracy', 'LongPassing', 'BallControl', 'Acceleration',
       'SprintSpeed', 'Agility', 'Reactions', 'Balance', 'ShotPower',
       'Jumping', 'Stamina', 'Strength', 'LongShots', 'Aggression',
       'Interceptions', 'Positioning', 'Vision', 'Penalties', 'Composure',
       'Marking', 'StandingTackle', 'SlidingTackle', 'GKDiving', 'GKHandling',
       'GKKicking', 'GKPositioning', 'GKReflexes', 'Release Clause'],
      dtype='object')
```

INFERENCE:

The following columns were dropped from the dataset 'fifa.csv' as they doesn't have any weightage.
Dropping Cloumns

1. Photo
2. Flag
3. Club
4. Club logo
5. Nationality
6. Jersey Number
7. Preferred Foot
8. Loaned From
9. Work Rate
10. Body Type

In [241]:
```
df_fifa.drop(columns=['Photo','Flag','Club Logo','Jersey Number','Club','Nat
```

In [242]:
```
df_fifa.head()
```
Out[242]:

	ID	Name	Age	Overall	Potential	Value	Wage	International Reputation	Weak Foot	Skill Moves	Position
0	158023	L. Messi	31	94	94	€110.5M	€565K	5.0	4.0	4.0	RF
1	20801	Cristiano Ronaldo	33	94	94	€77M	€405K	5.0	4.0	5.0	ST
2	190871	Neymar Jr	26	92	93	€118.5M	€290K	5.0	5.0	5.0	LW
3	193080	De Gea	27	91	93	€72M	€260K	4.0	3.0	1.0	GK
4	192985	K. De Bruyne	27	91	92	€102M	€355K	4.0	5.0	4.0	RCM

In [243]:
```
df_fifa.shape
```
Out[243]:
```
(18207, 50)
```

3. The following columns need to be converted for further analysis (5 marks)

(You might encounter Nan values in the above columns. Pandas treats Nan values as float. Please keep that in mind when making the conversions.)

a) 'Value' Amount with Euro symbol as prefix and suffix ‘K’ or ‘M’ indicating thousands and millions respectively. Convert to Float after getting rid of currency symbol and suffix

In [244]:
```python
df_Values_to_float = []
print ("Before conversion\n", df_fifa['Value'].head())
for val in df_fifa['Value']:
    vals = 0
    if isinstance(val,str) and val[-1:] == 'M':
        df_Values_to_float.append(float(val[1:-1]) * 1000000)
    elif isinstance(val,str) and val[-1:] == 'K':
        df_Values_to_float.append(float(val[1:-1]) * 1000)
    elif isinstance(val,str):
        df_Values_to_float.append(float(val[1:]))
    else:
        df_Values_to_float.append(val)
df_fifa['Value'] = df_Values_to_float
print ("After conversion\n", df_fifa['Value'].head())
```

```
Before conversion
 0    €110.5M
1       €77M
2     €118.5M
3       €72M
4      €102M
Name: Value, dtype: object
After conversion
 0    110500000.0
1     77000000.0
2    118500000.0
3     72000000.0
4    102000000.0
Name: Value, dtype: float64
```

b) 'Wage' Amount with Euro symbol as prefix and suffix ‘K’ or ‘M’ indicating thousands and millions respectively. Convert to Float after getting rid of currency symbol and suffix.

In [245]:

```python
df_Wages_to_float = []
print ("Before conversion\n", df_fifa['Wage'].head())

for wage in df_fifa['Wage']:
    w = 0
    if isinstance(wage,str) and wage[-1:] == 'M':
        df_Wages_to_float.append(float(wage[1:-1]) * 1000000)
    elif isinstance(wage,str) and wage[-1:] == 'K':
        df_Wages_to_float.append(float(wage[1:-1]) * 1000)
    elif isinstance(wage,str):
        df_Wages_to_float.append(float(wage[1:]))
    else:
        df_Wages_to_float.append(wage)
df_fifa['Wage'] = df_Wages_to_float
print ("After conversion\n", df_fifa['Wage'].head())
```

```
Before conversion
 0    €565K
1    €405K
2    €290K
3    €260K
4    €355K
Name: Wage, dtype: object
After conversion
 0    565000.0
1    405000.0
2    290000.0
3    260000.0
4    355000.0
Name: Wage, dtype: float64
```

c) 'Joined' Year as a string, in some cases complete date as string Convert to int with only year

In [246]:
```python
df_joined_to_year = []
print ("Before conversion\n", df_fifa['Joined'].head())

for doj in df_fifa['Joined']:
    if isinstance(doj,str):
        df_joined_to_year.append(int(doj[-4:]))
    else:
        df_joined_to_year.append(doj)
df_fifa['Joined'] = df_joined_to_year
print ("After conversion\n", df_fifa['Joined'].head())
```

```
Before conversion
 0     Jul 1, 2004
1    Jul 10, 2018
2     Aug 3, 2017
3     Jul 1, 2011
4    Aug 30, 2015
Name: Joined, dtype: object
After conversion
 0    2004.0
1    2018.0
2    2017.0
3    2011.0
4    2015.0
Name: Joined, dtype: float64
```

d) 'Contract Valid Until' Date as a string Convert to datetime type

In [247]:
```python
import datetime
# Function to convert string to datetime
def convert(date_time):
    format = '%b %d %Y'  # The format
    datetime_str = datetime.datetime.strptime(date_time, format)

    return datetime_str
```

In [248]:

```python
df_cv_datetime = []
print ("Before conversion\n", df_fifa['Contract Valid Until'].head())

for i in df_fifa['Contract Valid Until']:
    if isinstance(i, str) and len(i) == 4:
        date = 'Dec 31 ' + i
        df_cv_datetime.append(convert(date))
    elif isinstance(i, str) and len(i) > 4:
        i = i.replace(',','')
        df_cv_datetime.append(convert(i))
    else:
        df_cv_datetime.append(i)
df_fifa['Contract Valid Until'] = df_cv_datetime
print ("After conversion\n", df_fifa['Contract Valid Until'].head())

```

```
Before conversion
 0    2021
1    2022
2    2022
3    2020
4    2023
Name: Contract Valid Until, dtype: object
After conversion
 0   2021-12-31
1   2022-12-31
2   2022-12-31
3   2020-12-31
4   2023-12-31
Name: Contract Valid Until, dtype: datetime64[ns]
```

e) 'Height' In inches with a quotation mark Convert to Float with decimal points

In [249]:

```python
df_height_float = []
print ('Here we have converted height value from 5 foot 7 inches to 67 inche
print ("Before conversion\n", df_fifa['Height'].head())

for i in df_fifa['Height']:
    if isinstance(i,str):
        ft, inch = i.split("'")
        df_height_float.append(float(ft)*12 + float(inch))
    else:
        df_height_float.append(i)
df_fifa['Height'] = df_height_float
print ("After conversion\n", df_fifa['Height'].head())
```

```
Here we have converted height value from 5 foot 7 inches to 67 inches


Before conversion
 0     5'7
1     6'2
2     5'9
3     6'4
4    5'11
Name: Height, dtype: object
After conversion
 0    67.0
1    74.0
2    69.0
3    76.0
4    71.0
Name: Height, dtype: float64
```

f) 'Weight' Contains the suffix lbs Remove the suffix and convert to float

In [250]:
```python
df_weight_float = []
print ("Before conversion\n", df_fifa['Weight'].head())

for i in df_fifa['Weight']:
    if isinstance(i, str) and 'lbs' in i:
        df_weight_float.append(float(i[:-3]))
    else:
        df_weight_float.append(i)
df_fifa['Weight'] = df_weight_float
print ("After conversion\n", df_fifa['Weight'].head())
```

```
Before conversion
 0    159lbs
1    183lbs
2    150lbs
3    168lbs
4    154lbs
Name: Weight, dtype: object
After conversion
 0    159.0
1    183.0
2    150.0
3    168.0
4    154.0
Name: Weight, dtype: float64
```

g) 'Release Clause' Amount with Euro symbol as prefix and suffix ‘K’ or ‘M’ indicating thousands and millions respectively. Convert to Float after getting rid of currency symbol and suffix.

In [251]:
```python
df_rc_float = []
print ("Before conversion\n", df_fifa['Release Clause'].head())

for i in df_fifa['Release Clause']:
    if isinstance(i,str) and i[-1:] == 'M':
        df_rc_float.append(float(i[1:-1]) * 1000000)
    elif isinstance(i,str) and i[-1:] == 'K':
        df_rc_float.append(float(i[1:-1]) * 1000)
    elif isinstance(i,str):
        df_rc_float.append(float(i[1:]))
    else:
        df_rc_float.append(i)
df_fifa['Release Clause'] = df_rc_float
print ("After conversion\n", df_fifa['Release Clause'].head())

```

```
Before conversion
 0    €226.5M
1    €127.1M
2    €228.1M
3    €138.6M
4    €196.4M
Name: Release Clause, dtype: object
After conversion
 0    226500000.0
1    127100000.0
2    228100000.0
3    138600000.0
4    196400000.0
Name: Release Clause, dtype: float64
```

4. Check for missing values and do a mean imputation where necessary

In [252]:
```python
df_fifa.columns
```

Out[252]:
```
Index(['ID', 'Name', 'Age', 'Overall', 'Potential', 'Value', 'Wage',
       'International Reputation', 'Weak Foot', 'Skill Moves', 'Position',
       'Joined', 'Contract Valid Until', 'Height', 'Weight', 'Crossing',
       'Finishing', 'HeadingAccuracy', 'ShortPassing', 'Volleys', 'Dribbling',
       'Curve', 'FKAccuracy', 'LongPassing', 'BallControl', 'Acceleration',
       'SprintSpeed', 'Agility', 'Reactions', 'Balance', 'ShotPower',
       'Jumping', 'Stamina', 'Strength', 'LongShots', 'Aggression',
       'Interceptions', 'Positioning', 'Vision', 'Penalties', 'Composure',
       'Marking', 'StandingTackle', 'SlidingTackle', 'GKDiving', 'GKHandling',
       'GKKicking', 'GKPositioning', 'GKReflexes', 'Release Clause'],
      dtype='object')
```

```
In [253]:  1  df_fifa.dtypes
```

```
Out[253]: ID                                   int64
          Name                                object
          Age                                  int64
          Overall                              int64
          Potential                            int64
          Value                              float64
          Wage                               float64
          International Reputation           float64
          Weak Foot                          float64
          Skill Moves                        float64
          Position                            object
          Joined                             float64
          Contract Valid Until        datetime64[ns]
          Height                             float64
          Weight                             float64
          Crossing                           float64
          Finishing                          float64
          HeadingAccuracy                    float64
          ShortPassing                       float64
          Volleys                            float64
          Dribbling                          float64
          Curve                              float64
          FKAccuracy                         float64
          LongPassing                        float64
          BallControl                        float64
          Acceleration                       float64
          SprintSpeed                        float64
          Agility                            float64
          Reactions                          float64
          Balance                            float64
          ShotPower                          float64
          Jumping                            float64
          Stamina                            float64
          Strength                           float64
          LongShots                          float64
          Aggression                         float64
          Interceptions                      float64
          Positioning                        float64
          Vision                             float64
          Penalties                          float64
          Composure                          float64
          Marking                            float64
          StandingTackle                     float64
          SlidingTackle                      float64
          GKDiving                           float64
          GKHandling                         float64
          GKKicking                          float64
          GKPositioning                      float64
          GKReflexes                         float64
          Release Clause                     float64
          dtype: object
```

In [254]:

```
df_fifa.isnull().sum()
```

Out[254]:

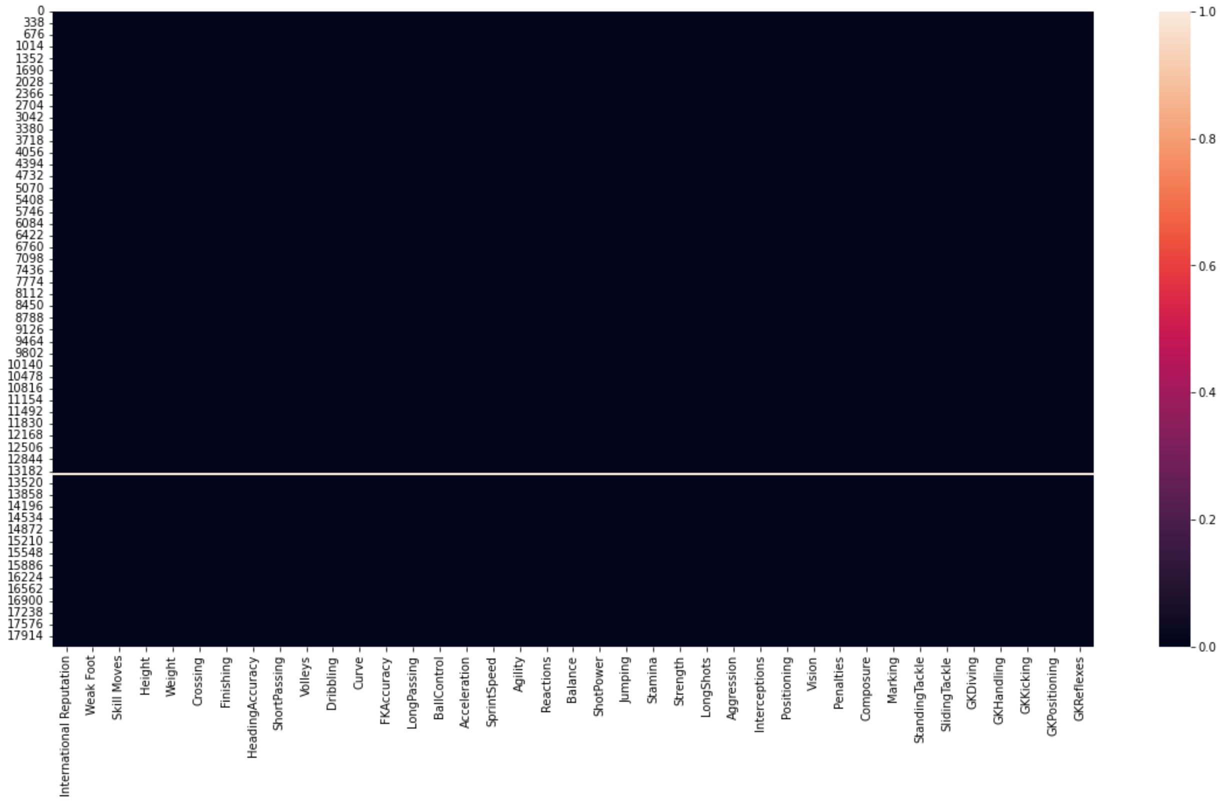
```
ID                             0
Name                           0
Age                            0
Overall                        0
Potential                      0
Value                          0
Wage                           0
International Reputation      48
Weak Foot                     48
Skill Moves                   48
Position                      60
Joined                      1553
Contract Valid Until         289
Height                        48
Weight                        48
Crossing                      48
Finishing                     48
HeadingAccuracy               48
ShortPassing                  48
Volleys                       48
Dribbling                     48
Curve                         48
FKAccuracy                    48
LongPassing                   48
BallControl                   48
Acceleration                  48
SprintSpeed                   48
Agility                       48
Reactions                     48
Balance                       48
ShotPower                     48
Jumping                       48
Stamina                       48
Strength                      48
LongShots                     48
Aggression                    48
Interceptions                 48
Positioning                   48
Vision                        48
Penalties                     48
Composure                     48
Marking                       48
StandingTackle                48
SlidingTackle                 48
GKDiving                      48
GKHandling                    48
GKKicking                     48
GKPositioning                 48
GKReflexes                    48
Release Clause              1564
dtype: int64
```

```
In [255]:
col_with_48_missing_val = []
for i in df_fifa:
    if df_fifa[i].isnull().sum() == 48:
        col_with_48_missing_val.append(i)
len(col_with_48_missing_val)
```

```
Out[255]: 39
```

```
In [256]:
plt.figure(figsize=(20,10))
sns.heatmap(df_fifa[col_with_48_missing_val].select_dtypes(exclude='object')
```

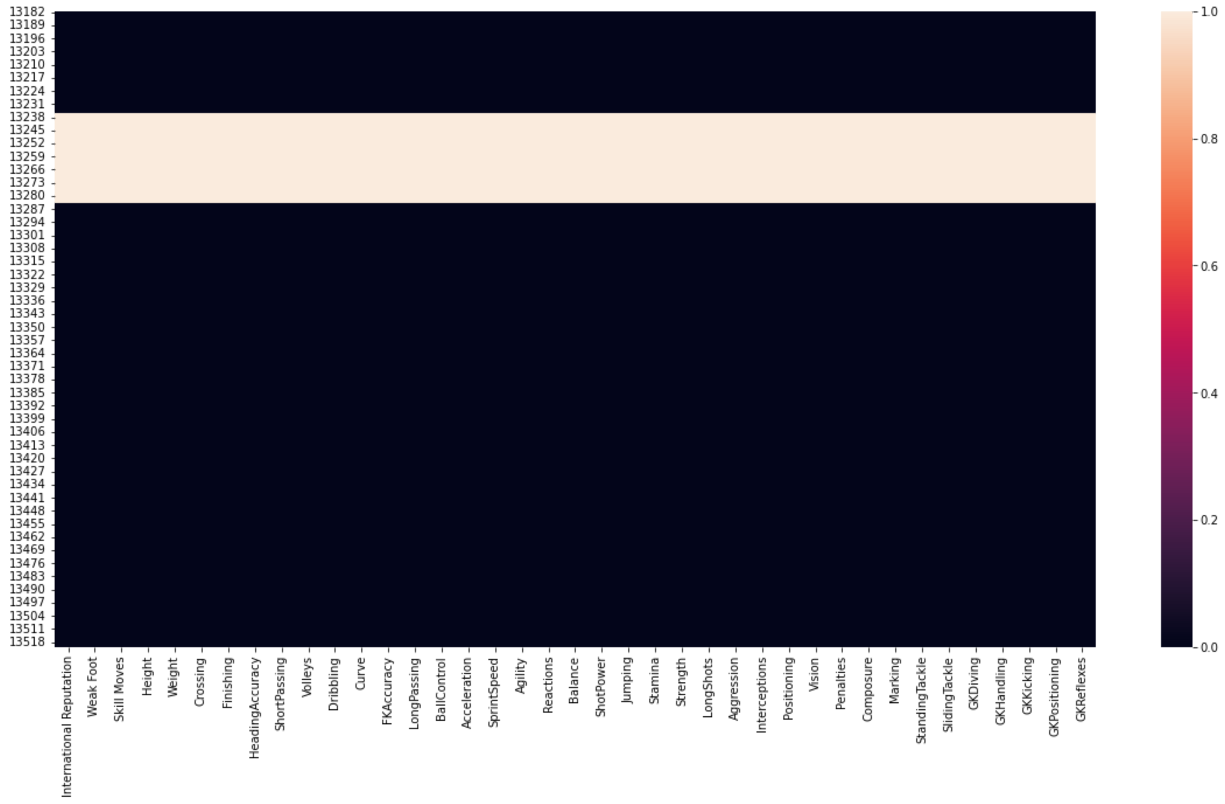
```
Out[256]: <AxesSubplot:>
```



Lets see the heatmap for the index between 13182 and 13520 for the missing values

In [257]:
```
plt.figure(figsize=(20,10))
sns.heatmap(df_fifa.loc[13182:13520,col_with_48_missing_val].select_dtypes(e
```

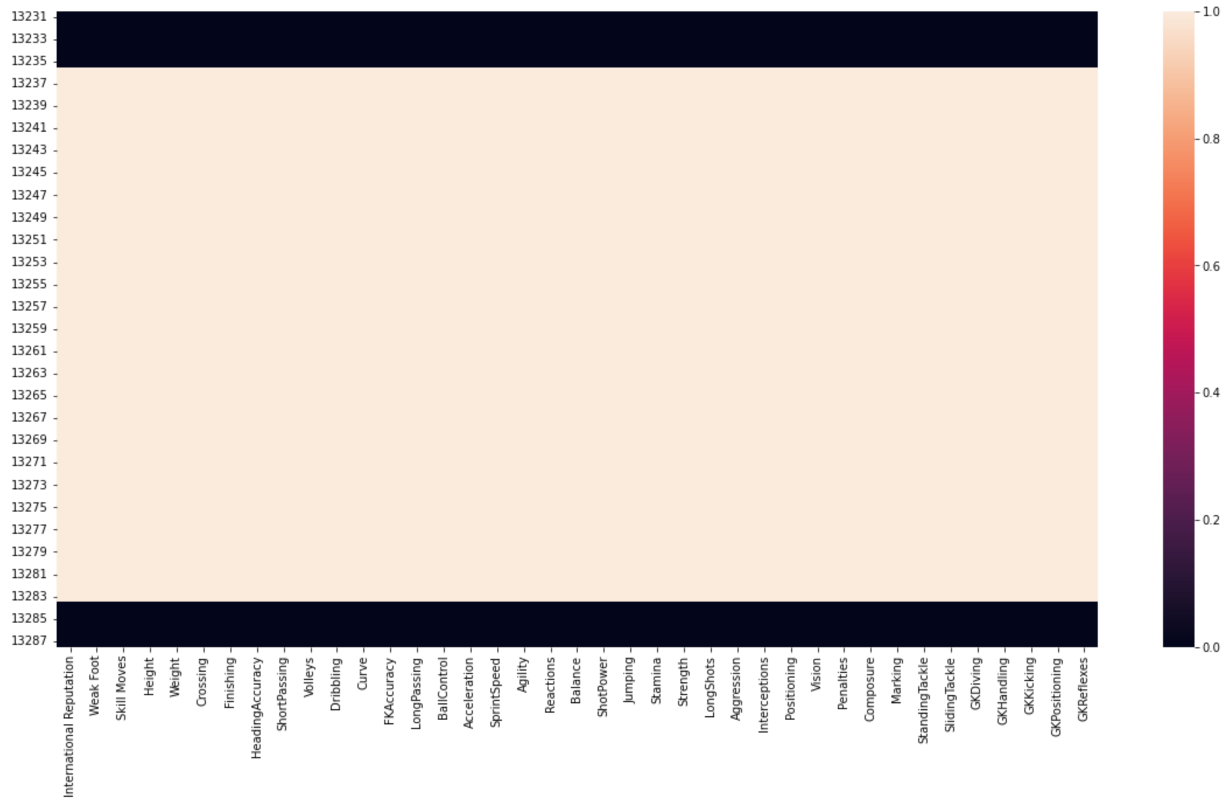
Out[257]:
```
<AxesSubplot:>
```



Lets see the heatmap for the index between 13231 and 13287 for the missing values

In [258]:
```python
plt.figure(figsize=(20,10))
sns.heatmap(df_fifa.loc[13231:13287,col_with_48_missing_val].select_dtypes(e
```

Out[258]:
```
<AxesSubplot:>
```



This seems like the 48 players missing their 39 rating parameters

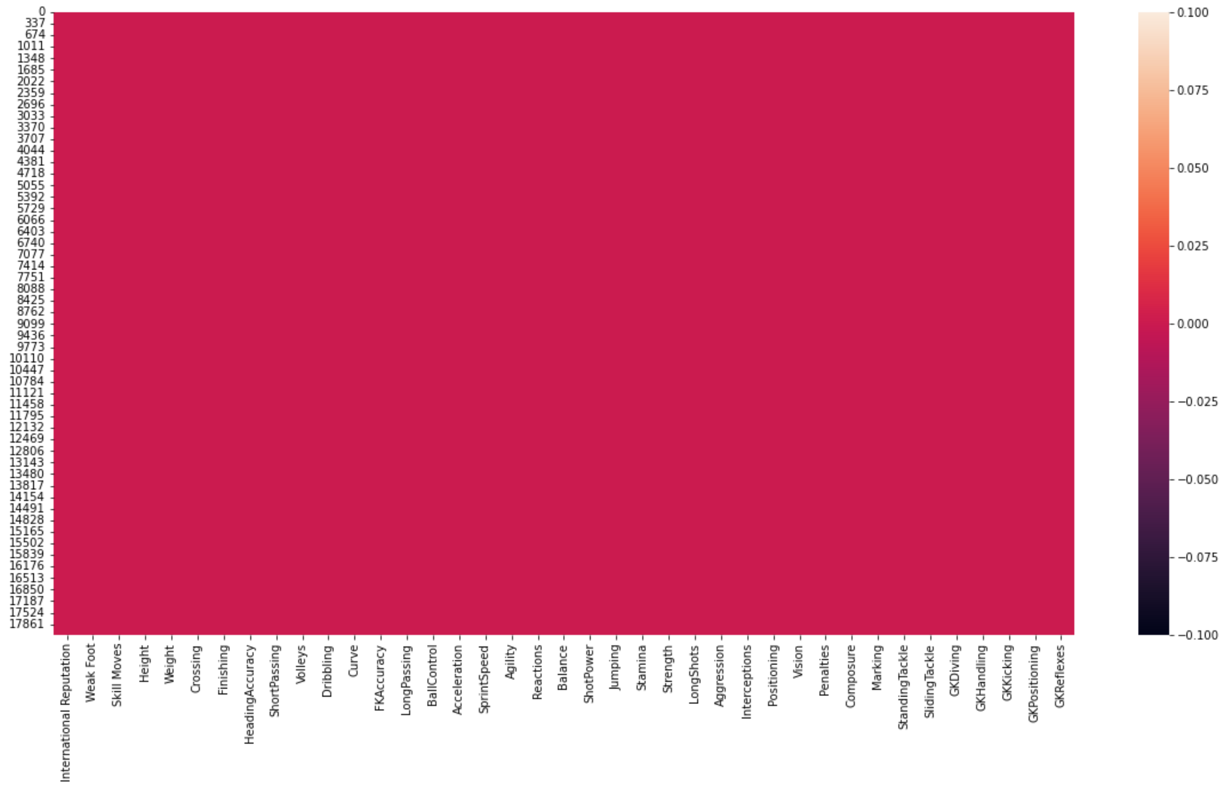
Since its creating a noise in a dataset

So, we can remove those 48 players from the list for further analysis

In [259]:
```python
df_fifa.dropna(axis=0,inplace=True,subset=col_with_48_missing_val)
df_fifa.reset_index(drop=True, inplace=True)
```

In [260]:
```
plt.figure(figsize=(20,10))
sns.heatmap(df_fifa[col_with_48_missing_val].select_dtypes(exclude='object')
```

Out[260]:
```
<AxesSubplot:>
```



In [261]:
```
df_fifa.shape
```

Out[261]:
```
(18159, 50)
```

```
In [262]: df_fifa.isnull().sum()
```

```
Out[262]: ID                          0
Name                        0
Age                         0
Overall                     0
Potential                   0
Value                       0
Wage                        0
International Reputation    0
Weak Foot                   0
Skill Moves                 0
Position                   12
Joined                   1505
Contract Valid Until      241
Height                      0
Weight                      0
Crossing                    0
Finishing                   0
HeadingAccuracy             0
ShortPassing                0
Volleys                     0
Dribbling                   0
Curve                       0
FKAccuracy                  0
LongPassing                 0
BallControl                 0
Acceleration                0
SprintSpeed                 0
Agility                     0
Reactions                   0
Balance                     0
ShotPower                   0
Jumping                     0
Stamina                     0
Strength                    0
LongShots                   0
Aggression                  0
Interceptions               0
Positioning                 0
Vision                      0
Penalties                   0
Composure                   0
Marking                     0
StandingTackle              0
SlidingTackle               0
GKDiving                    0
GKHandling                  0
GKKicking                   0
GKPositioning               0
GKReflexes                  0
Release Clause           1516
dtype: int64
```

In [263]:
```
df_fifa['Position'].value_counts(dropna=False)
```
Out[263]:
```
ST     2152
GK     2025
CB     1778
CM     1394
LB     1322
RB     1291
RM     1124
LM     1095
CAM     958
CDM     948
RCB     662
LCB     648
LCM     395
RCM     391
LW      381
RW      370
RDM     248
LDM     243
LS      207
RS      203
RWB      87
LWB      78
CF       74
LAM      21
RAM      21
RF       16
LF       15
NaN      12
Name: Position, dtype: int64
```

INFERENCE: Position for 12 players have missing values and we can't implement imputation.So we can removing the 12 players.If we are imputing some random position to the 12 players, their charcteristics might change as per domain inference.

In [264]:
```
df_fifa.dropna(axis=0,inplace=True,subset=['Position'])
df_fifa.reset_index(drop=True, inplace=True)
```
In [265]:
```
df_fifa.shape
```
Out[265]:
```
(18147, 50)
```

```
In [266]: df_fifa.isnull().sum()
```

```
Out[266]: ID                             0
          Name                           0
          Age                            0
          Overall                        0
          Potential                      0
          Value                          0
          Wage                           0
          International Reputation       0
          Weak Foot                      0
          Skill Moves                    0
          Position                       0
          Joined                      1493
          Contract Valid Until         229
          Height                         0
          Weight                         0
          Crossing                       0
          Finishing                      0
          HeadingAccuracy                0
          ShortPassing                   0
          Volleys                        0
          Dribbling                      0
          Curve                          0
          FKAccuracy                     0
          LongPassing                    0
          BallControl                    0
          Acceleration                   0
          SprintSpeed                    0
          Agility                        0
          Reactions                      0
          Balance                        0
          ShotPower                      0
          Jumping                        0
          Stamina                        0
          Strength                       0
          LongShots                      0
          Aggression                     0
          Interceptions                  0
          Positioning                    0
          Vision                         0
          Penalties                      0
          Composure                      0
          Marking                        0
          StandingTackle                 0
          SlidingTackle                  0
          GKDiving                       0
          GKHandling                     0
          GKKicking                      0
          GKPositioning                  0
          GKReflexes                     0
          Release Clause              1504
          dtype: int64
```

We have 1504 missing value in Release Clause dataset and it has outliers.So we can

implement median imputation with respect to each position.

In [267]:
```python
for i in range(len(df_fifa)):
    if np.isnan(df_fifa['Release Clause'][i]):
        posi = df_fifa['Position'][i]
        med = df_fifa[df_fifa['Position'] == posi].median()
        df_fifa['Release Clause'][i] = med['Release Clause']
```

In [268]:
```python
df_fifa['Release Clause'].isnull().sum()
```

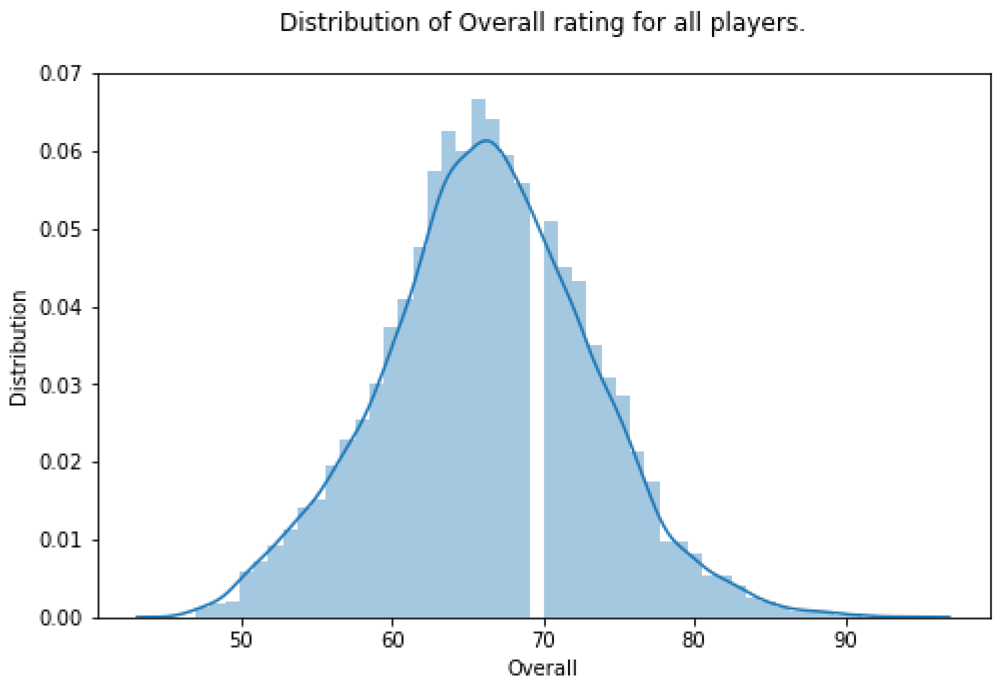
Out[268]:
```
0
```

INFERENCE: For 'Release Clause' column we have found median with respect to each 'Position' and imputed the same in place of null values.

Exploratory Analysis:

1. Plot the distribution of Overall rating for all players.

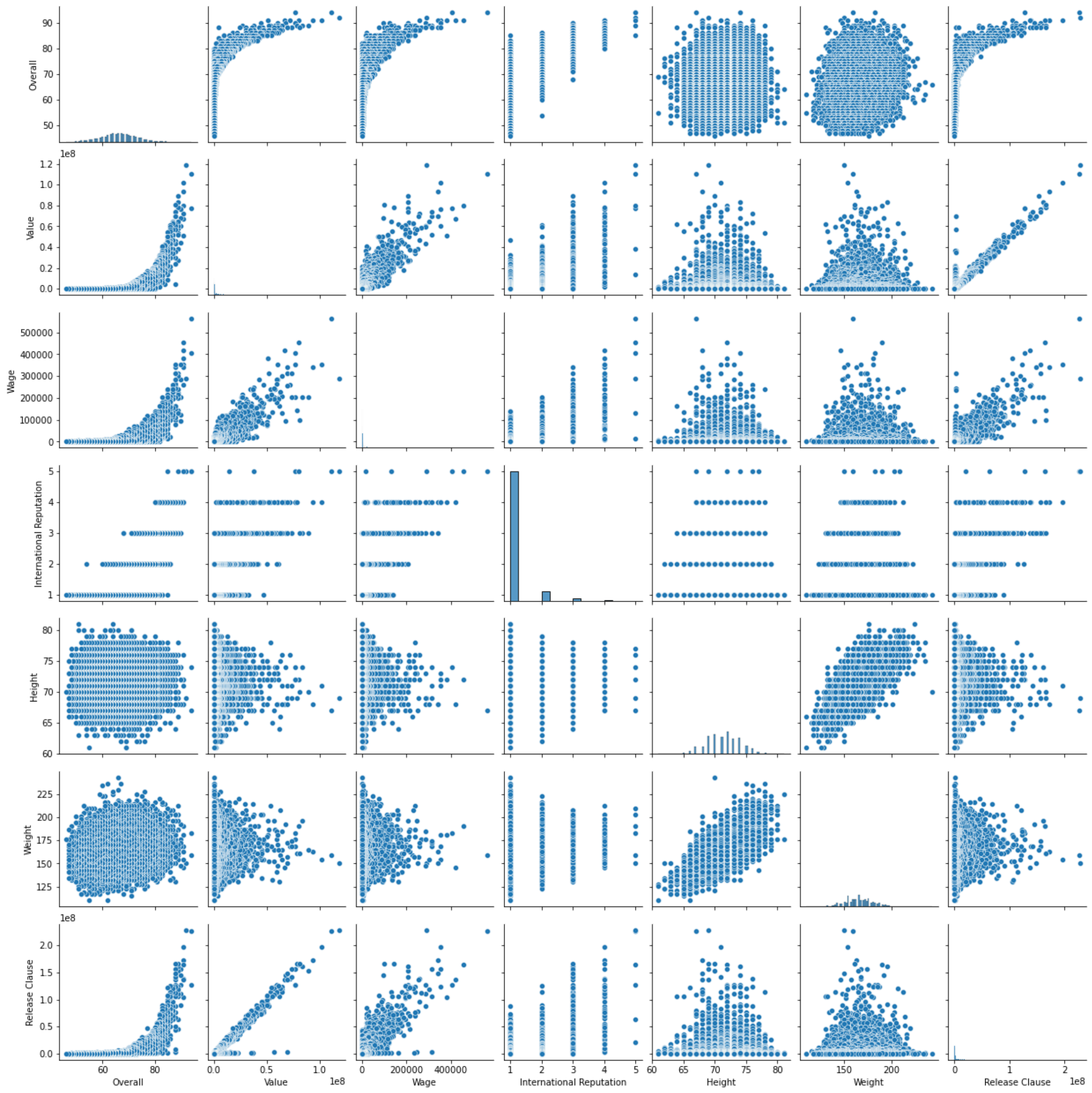
In [269]:
```python
plt.figure(figsize=(8,5))
plt.title('Distribution of Overall rating for all players.\n')
sns.distplot(df_fifa['Overall'])
plt.xlabel("Overall")
plt.ylabel("Distribution")
plt.show()
```



2. Generate pair plots for the following variables
Overall, Value, Wage, International Reputation, Height, Weight, Release Clause

In [270]:
```
pair_plot_col = ["Overall", "Value", "Wage", "International Reputation", "He
sns.pairplot(df_fifa[pair_plot_col])
```

Out[270]:
```
<seaborn.axisgrid.PairGrid at 0x2abdb2c8520>
```



3.Generate a table containing the top 20 players ranked by Overall score and whose contract expires in 2020

In [271]:
```
max_date = df_fifa['Contract Valid Until'].max()
df_fifa['Contract Valid Until'].fillna(max_date, inplace = True)
```

In [272]:
```
df_fifa['Contract Exp Year'] = pd.DatetimeIndex(df_fifa['Contract Valid Unti
type(df_fifa['Contract Exp Year'][0])
```

Out[272]:
```
numpy.int64
```

In [273]:
```python

df_top_20 = df_fifa.where(df_fifa['Contract Exp Year'] <= np.int64(2020)).nl
df_fifa.drop(columns=['Contract Exp Year'], inplace=True)
df_top_20
```

Out[273]:

	ID	Name	Age	Overall	Potential	Value	Wage	International Reputation	Weak Foot	Sk Mov
3	193080.0	De Gea	27.0	91.0	93.0	72000000.0	260000.0	4.0	3.0	1
5	183277.0	E. Hazard	27.0	91.0	91.0	93000000.0	340000.0	4.0	4.0	4
6	177003.0	L. Modrić	32.0	91.0	91.0	67000000.0	420000.0	4.0	4.0	4
8	155862.0	Sergio Ramos	32.0	91.0	91.0	51000000.0	380000.0	4.0	3.0	3
12	182493.0	D. Godín	32.0	90.0	90.0	44000000.0	125000.0	3.0	3.0	2
13	168542.0	David Silva	32.0	90.0	90.0	60000000.0	285000.0	4.0	2.0	4
21	179813.0	E. Cavani	31.0	89.0	89.0	60000000.0	200000.0	4.0	4.0	3
24	138956.0	G. Chiellini	33.0	89.0	89.0	27000000.0	215000.0	4.0	3.0	2
28	198710.0	J. Rodríguez	26.0	88.0	89.0	69500000.0	315000.0	4.0	3.0	4
31	190460.0	C. Eriksen	26.0	88.0	91.0	73500000.0	205000.0	3.0	5.0	4
38	167664.0	G. Higuaín	30.0	88.0	88.0	57000000.0	245000.0	4.0	4.0	3
39	164240.0	Thiago Silva	33.0	88.0	88.0	24000000.0	165000.0	4.0	3.0	2
41	1179.0	G. Buffon	40.0	88.0	88.0	4000000.0	77000.0	4.0	2.0	1
46	193041.0	K. Navas	31.0	87.0	87.0	30500000.0	195000.0	3.0	3.0	1
49	189332.0	Jordi Alba	29.0	87.0	87.0	38000000.0	250000.0	3.0	3.0	3
50	175943.0	D. Mertens	31.0	87.0	87.0	45000000.0	135000.0	3.0	4.0	4
51	172871.0	J. Vertonghen	31.0	87.0	87.0	34000000.0	155000.0	3.0	3.0	3
52	171877.0	M. Hamšík	30.0	87.0	87.0	46500000.0	125000.0	3.0	5.0	3
64	191043.0	Alex Sandro	27.0	86.0	86.0	36500000.0	160000.0	3.0	3.0	3
71	184087.0	T. Alderweireld	29.0	86.0	87.0	39000000.0	150000.0	3.0	3.0	2

INFERENCE: We have extracted top 20 players with highest Overall rating whose contract expires in 2020. We have also dropped the new column 'Contract_exp_year' that we created for this analysis.

a) What would the average wage for this set of players be?

In [274]:
```
print(df_top_20['Wage'].mean())
```

```
220100.0
```

b) What is the average age?

In [275]:
```
print(df_top_20['Age'].mean())
```

```
30.45
```

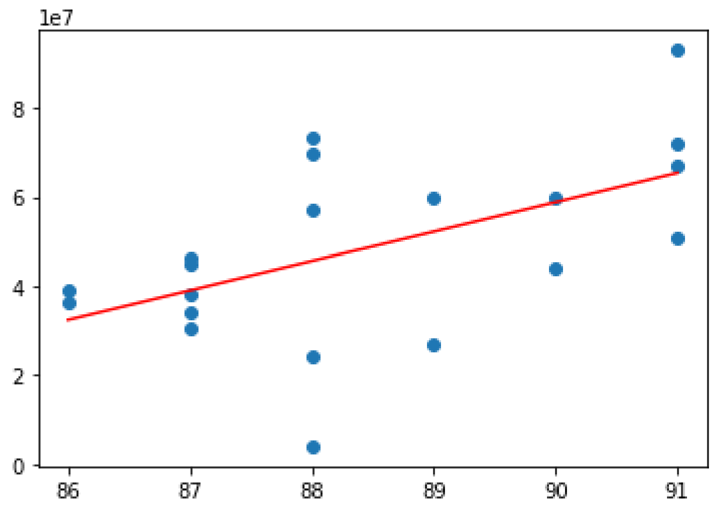
c) Is there a correlation between the Overall rating and Value for these players?

In [276]:
```
df_top_20['Overall'].corr(df_top_20['Value'])
```

Out[276]:
```
0.5376602976016892
```

In [277]:
```
plt.scatter(df_top_20['Overall'], df_top_20['Value'])
plt.plot(np.unique(df_top_20['Overall']),
         np.poly1d(np.polyfit(df_top_20['Overall'], df_top_20['Value'], 1))
         (np.unique(df_top_20['Overall'])), color='red')
```

Out[277]:
```
[<matplotlib.lines.Line2D at 0x2abf2acefa0>]
```



INFERENCE: From the above scatterplot, we can conclude that

columns 'Overall' and 'Value' follows moderate positive correlation.

4. Convert the categorical features to numerical features with suitable encoding techniques.

In [278]:
```python
df_fifa_obj = df_fifa.select_dtypes(include='object')
df_fifa_obj
```

Out[278]:

	Name	Position
0	L. Messi	RF
1	Cristiano Ronaldo	ST
2	Neymar Jr	LW
3	De Gea	GK
4	K. De Bruyne	RCM
5	E. Hazard	LF
6	L. Modrić	RCM
7	L. Suárez	RS
8	Sergio Ramos	RCB
9	J. Oblak	GK
10	R. Lewandowski	ST
11	T. Kroos	LCM

In [279]:
```python
from sklearn.preprocessing import LabelEncoder
encode=LabelEncoder()
df_fifa['Position encoded']=encode.fit_transform(df_fifa['Position'])
```

In [280]:
```python
encode.fit(df_fifa['Position'])
mapp =  dict(zip(range(len(encode.classes_)),encode.classes_))
print(mapp)
```

```
{0: 'CAM', 1: 'CB', 2: 'CDM', 3: 'CF', 4: 'CM', 5: 'GK', 6: 'LAM', 7: 'LB', 8:
'LCB', 9: 'LCM', 10: 'LDM', 11: 'LF', 12: 'LM', 13: 'LS', 14: 'LW', 15: 'LWB',
16: 'RAM', 17: 'RB', 18: 'RCB', 19: 'RCM', 20: 'RDM', 21: 'RF', 22: 'RM', 23:
'RS', 24: 'RW', 25: 'RWB', 26: 'ST'}
```

INFERENCE: The above code is simply used to map categorical value to its respective numerical value. For example, 'CAM' is encoded to value 0 .

In [281]:
```python
df_fifa['Position'] = df_fifa['Position encoded']
df_fifa.drop(columns='Position encoded', inplace=True)
```

5. Generate tables containing the top 5 players by Overall rating for each unique position.

In [282]:

```python
tables = []
for i in range(len(df_fifa['Position'].unique())):
    tables.append(df_fifa.where(df_fifa['Position'] == i).nlargest(5, ['Over
```

In [283]:

```python
for i in range(len(tables)):
    print('\n\n    Position : ', mapp[i])
    print('\n',tables[i]['Name'])
```

```


    Position :  CAM

 17       A. Griezmann
31          C. Eriksen
61    Roberto Firmino
66           T. Müller
74             M. Özil
Name: Name, dtype: object


    Position :  CB

 12         D. Godín
42        S. Umtiti
73       M. Benatia
89     N. Otamendi
102          Naldo
```

INFERENCE: Similarly, we can find top 5 players with highest overall rating for rest 26 Positions using 'for loop' . To avoid confusion we chose 1 features: 'Name' to be printed.

a) Are there any players appearing in more than one Table. Please point out such players.

In [284]:

```python
playerId = []
for i in range(len(tables)):
    for j in range(len(tables)-1):
        j = j+1
        if j != i:
            intr = (set(tables[i]['ID'])).intersection(set(tables[j]['ID']))
            if len(intr) != 0:
                playerId.append(intr)

if playerId==[]:
    print('NO players found in more than one table')
else:
    print('YES, players found in more than one table',playerId)

```

```
NO players found in more than one table
```

INFERENCE: From the above result, it is clear that there is no player appearing in more than 1 table.

b) What is the average wage one can expect to pay for the top 5 in every position?

In [285]:
```python
position = list(df_fifa['Position'].unique())
for i in position:
    print('Average wage for the top 5 in ',i,':',df_fifa[df_fifa['Position']
```

```
Average wage for the top 5 in  21 : 148000.0
Average wage for the top 5 in  26 : 294000.0
Average wage for the top 5 in  14 : 261000.0
Average wage for the top 5 in  5 : 192800.0
Average wage for the top 5 in  19 : 240800.0
Average wage for the top 5 in  11 : 121200.0
Average wage for the top 5 in  23 : 132200.0
Average wage for the top 5 in  18 : 231000.0
Average wage for the top 5 in  9 : 184400.0
Average wage for the top 5 in  1 : 139600.0
Average wage for the top 5 in  10 : 126600.0
Average wage for the top 5 in  0 : 174000.0
Average wage for the top 5 in  2 : 217000.0
Average wage for the top 5 in  13 : 130200.0
Average wage for the top 5 in  8 : 162000.0
Average wage for the top 5 in  22 : 131400.0
Average wage for the top 5 in  6 : 81600.0
Average wage for the top 5 in  12 : 164600.0
Average wage for the top 5 in  7 : 177200.0
Average wage for the top 5 in  20 : 105000.0
Average wage for the top 5 in  24 : 202000.0
Average wage for the top 5 in  4 : 130600.0
Average wage for the top 5 in  17 : 155400.0
Average wage for the top 5 in  16 : 45400.0
Average wage for the top 5 in  3 : 47400.0
Average wage for the top 5 in  25 : 44200.0
Average wage for the top 5 in  15 : 34200.0
```

INFERENCE: Above result clearly shows average wage for top 5 players in each position.