```
"""
Project-1
Apply
logistic Regression
SVM
Decision Tree
RandomForest
on the Loan dataset and check were you will get the best possible
accuracy
project note : Dependent Variable is Loan Status
"""
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import svm
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report
df=pd.read_csv("E:\loan.csv")
```

```
df['Gender'].value_counts()
```

```
Male      489
Female    112
Name: Gender, dtype: int64
```

```
df['Dependents'] = df['Dependents'].fillna('0')
df['Dependents'] = df['Dependents'].replace({'3+':3})


np.mean(df['Loan_Amount_Term'])
```

```
342.0
```

```
df['Loan_Amount_Term'] = df['Loan_Amount_Term'].fillna(342.0)
df['Loan_Amount_Term'].isnull().sum()
```

```
0
```

```
df['Credit_History'] = df['Credit_History'].fillna(1.0)
df['LoanAmount'] = df['LoanAmount'].fillna(146.4)
```

```
df=df.fillna(value=0)
```

```
df['Property_Area'].value_counts()
```

```
Semiurban    233
Urban        202
Rural        179
Name: Property_Area, dtype: int64
```

```
df['Gender'] = df['Gender'].replace({'Male':0, 'Female':1,'unknown' :
2})
```

```
df['Married'] = df['Married'].replace({'Yes' :1, 'No': 0,
'unknown':2})


df['Education'] = df['Education'].replace ({'Graduate' : 1, 'Not
Graduate' : 0})
df['Self_Employed'] = df['Self_Employed'].replace ({'Yes': 1,'No' : 0,
'unknown':2})

df['Property_Area'] = df['Property_Area'].replace ({'Semiurban':
1,'Urban' : 0, 'Rural':2})

df['Loan_Status'] =df['Loan_Status'].replace({'Y':1, 'N':0})

mylist_train = ['Gender', 'Married', 'Dependents', 'Education',
'Self_Employed', 'Credit_History', 'Property_Area' , 'Loan_Status']

for i in mylist_train:
    df[i] = df[i].astype({i:'category'})

df['Loan_Status'].value_counts(normalize=True)*100
```

```
1    68.729642
0    31.270358
Name: Loan_Status, dtype: float64
```

```
df= df.drop(columns=['Loan_ID'])

X_train = df.drop(columns=['Loan_Status'])

y_train = df['Loan_Status']

from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()

lr.fit(X_train,y_train)
lr.score(X_train,y_train)
```

```
C:\Users\Renuka\anaconda3\lib\site-packages\sklearn\linear_model\
_logistic.py:814: ConvergenceWarning: lbfgs failed to converge
(status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as
shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
C:\Users\Renuka\anaconda3\lib\site-packages\sklearn\base.py:566:
```

```
FutureWarning: Arrays of bytes/strings is being converted to decimal
numbers if dtype='numeric'. This behavior is deprecated in 0.24 and
will be removed in 1.1 (renaming of 0.26). Please convert your data to
numeric values explicitly instead.
  X = check_array(X, **check_params)
```

```
0.8094462540716613
```

```
from sklearn import svm
from sklearn.model_selection import train_test_split
x = df.iloc[:, :-2]
y = df.iloc[:, -1]
x_train, x_test, y_train, y_test = train_test_split(x, y, random_state
= 0, test_size = 0.2)

clf = svm.SVC(kernel='rbf')
clf.fit(x_train,y_train)
y_pred = clf.predict(x_test)
```

```
from sklearn.metrics import accuracy_score
print("Accuracy:", accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.7317073170731707
```

```
from sklearn.tree import DecisionTreeRegressor
tree=DecisionTreeRegressor()#(criterion="entropy",max_depth=4)
tree
```

```
DecisionTreeRegressor()
```

```
tree.fit(x_train,y_train)
```

```
DecisionTreeRegressor()
```

```
y_pred=tree.predict(x_test)
```

```
print(y_pred[0:5])
print(y_test[0:5])
```

```
[0. 1. 0. 0. 1.]
454    1
52     0
536    1
469    0
55     1
Name: Loan_Status, dtype: category
Categories (2, int64): [0, 1]
```

```
from sklearn import metrics
print("DecisionTrees's Accuracy: ",metrics.r2_score(y_pred,y_test))
```

```
DecisionTrees's Accuracy:  -0.21569767441860477
```

```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(random_state=42)
rfc.fit(x_train, y_train)
rfc_pred = rfc.predict(x_test)
```

```
from sklearn.metrics import classification_report,confusion_matrix
confusion_matrix(y_test, rfc_pred)
```

```
array([[14, 19],
       [ 7, 83]], dtype=int64)
```

```
classification_report(y_test, rfc_pred)
```

```
'               precision    recall  f1-score   support\n\n           0
0.67      0.42      0.52        33\n           1       0.81      0.92
0.86        90\n\n    accuracy                           0.79
123\n   macro avg       0.74      0.67      0.69       123\nweighted
avg       0.77      0.79      0.77       123\n'
```

```
######################################################################
##########################################################

"""
Project-2
Apply Exploratory Data Analysis on the FifaDataset
Convert the appropriate columns to machine understandable columns
using label encoding
Handle Null values
Apply the correlation and remove the column which are more than 60%
correlated
Apply the variance and remove the columns whose variance is less that
20%
Understand the Dataset whether the dataset is corr
"""
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from collections import Counter
import warnings
warnings.filterwarnings('ignore')
import plotly
sns.set_style('darkgrid')
```

```
fifa=pd.read_csv("E:/fifa.csv")
```

```
fifa.head()
```

```
       id          name                        full_name  birth_date
age  \
0  158023      L. Messi  Lionel Andrés Messi Cuccittini  1987-06-24
31
```

```
1  190460    C. Eriksen    Christian  Dannemann Eriksen  1992-02-14
27
2  195864      P. Pogba                        Paul Pogba  1993-03-15
25
3  198219    L. Insigne                   Lorenzo Insigne  1991-06-04
27
4  201024  K. Koulibaly                 Kalidou Koulibaly  1991-06-20
27

   height_cm  weight_kgs  positions nationality  overall_rating  ...
LWB  \
0     170.18        72.1   CF,RW,ST   Argentina              94  ...
64+2
1     154.94        76.2  CAM,RM,CM     Denmark              88  ...
71+3
2     190.50        83.9     CM,CAM      France              88  ...
76+3
3     162.56        59.0      LW,ST       Italy              88  ...
63+3
4     187.96        88.9         CB     Senegal              88  ...
73+3

    LDM   CDM   RDM   RWB    LB   LCB    CB   RCB    RB
0  61+2  61+2  61+2  64+2  59+2  48+2  48+2  48+2  59+2
1  71+3  71+3  71+3  71+3  66+3  57+3  57+3  57+3  66+3
2  77+3  77+3  77+3  76+3  74+3  72+3  72+3  72+3  74+3
3  58+3  58+3  58+3  63+3  58+3  44+3  44+3  44+3  58+3
4  77+3  77+3  77+3  73+3  76+3  85+3  85+3  85+3  76+3

[5 rows x 92 columns]
```

```
fifa.shape
```

```
(17954, 92)
```

```
for col in fifa.columns:
    print(col)
```

```
id
name
full_name
birth_date
age
height_cm
weight_kgs
positions
nationality
overall_rating
potential
value_euro
wage_euro
```

```
preferred_foot
international_reputation(1-5)
weak_foot(1-5)
skill_moves(1-5)
work_rate
body_type
release_clause_euro
club_team
club_rating
club_position
club_jersey_number
club_join_date
contract_end_year
national_team
national_rating
national_team_position
national_jersey_number
crossing
finishing
heading_accuracy
short_passing
volleys
dribbling
curve
freekick_accuracy
long_passing
ball_control
acceleration
sprint_speed
agility
reactions
balance
shot_power
jumping
stamina
strength
long_shots
aggression
interceptions
positioning
vision
penalties
composure
marking
standing_tackle
sliding_tackle
GK_diving
GK_handling
GK_kicking
GK_positioning
```

```
GK_reflexes
tags
traits
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
```

```
fifa['nationality'].value_counts()[0:10]
```

```
England        1658
Germany        1199
Spain          1070
France          925
Argentina       904
Brazil          832
Italy           655
Colombia        624
Japan           466
Netherlands     441
Name: nationality, dtype: int64
```

```
fifa['nationality'].value_counts()[:-1]
```

```
England             1658
Germany             1199
Spain               1070
France               925
Argentina            904
```

```
                    ...
Vietnam                1
Papua New Guinea       1
South Sudan            1
Malta                  1
Yemen                  1
Name: nationality, Length: 159, dtype: int64
```

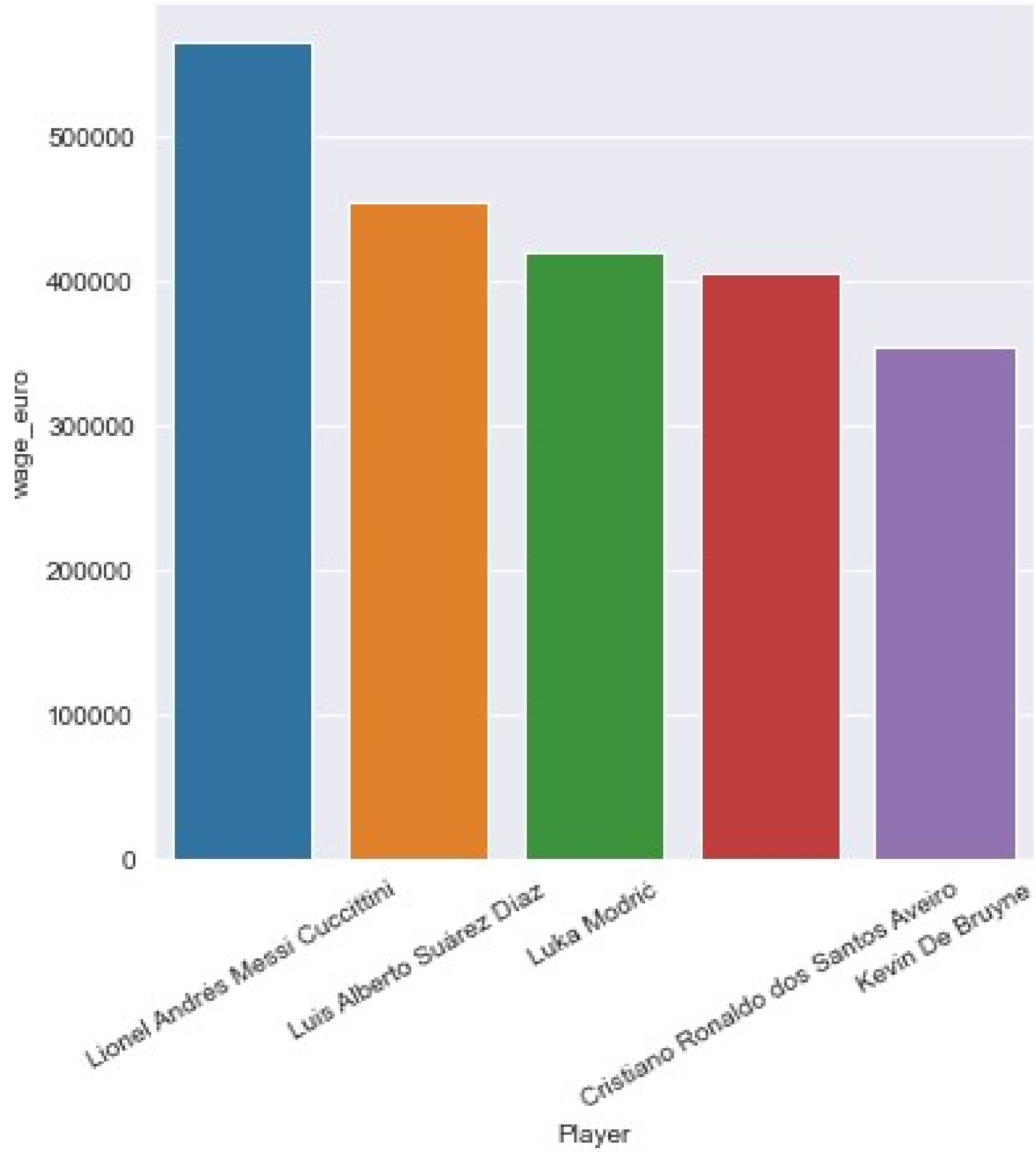
```
player_salary = fifa[['full_name','wage_euro']]
player_salary.head()
```

```
                        full_name  wage_euro
0  Lionel Andrés Messi Cuccittini   565000.0
1    Christian  Dannemann Eriksen   205000.0
2                      Paul Pogba   255000.0
3                 Lorenzo Insigne   165000.0
4               Kalidou Koulibaly   135000.0
```

```
player_salary = 
player_salary.sort_values(by=['wage_euro'],ascending=False)
player_salary.head()
```

```
                                 full_name  wage_euro
0           Lionel Andrés Messi Cuccittini   565000.0
17938             Luis Alberto Suárez Díaz   455000.0
17939                          Luka Modrić   420000.0
17944  Cristiano Ronaldo dos Santos Aveiro   405000.0
17941                      Kevin De Bruyne   355000.0
```

```
plt.figure(figsize=(6,6))
sns.barplot(player_salary['full_name'][0:5],player_salary['wage_euro']
[0:5])
plt.xticks(rotation=30)
plt.xlabel('Player')
plt.ylabel('wage_euro');
```



```
Germany = fifa[fifa['nationality']=='Germany']

Germany.sort_values(by=['height_cm'],ascending=False).head()
```

```
           id           name         full_name  birth_date  age
height_cm  \
14136  205451       B. Röcker   Benedikt Röcker  1989-11-19   29
198.12
16936  199833   L. Unnerstall   Lars Unnerstall  1990-07-20   28
198.12
14671  158657      T. Kessler    Thomas Kessler  1986-01-20   33
198.12
7839   203835  D. Orlishausen  Dirk Orlishausen  1982-08-15   36
198.12
17736  179783     R. Fährmann     Ralf Fährmann  1988-09-27   30
```

```
198.12

       weight_kgs positions nationality  overall_rating  ...   LWB
LDM  \
14136        92.1        CB     Germany              70  ...  56+2
62+2
16936       103.0        GK     Germany              76  ...   NaN
NaN
14671        92.1        GK     Germany              71  ...   NaN
NaN
7839         93.9        GK     Germany              61  ...   NaN
NaN
17736        98.0        GK     Germany              82  ...   NaN
NaN

        CDM   RDM   RWB    LB   LCB    CB   RCB    RB
14136  62+2  62+2  56+2  59+2  69+2  69+2  69+2  59+2
16936   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
14671   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
7839    NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
17736   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN

[5 rows x 92 columns]
```

```
Germany.sort_values(by=['weight_kgs'],ascending=False).head()
```

```
           id           name        full_name  birth_date  age
height_cm  \
8266   239746   L. Watkowiak  Lukas Watkowiak  1996-03-06   23
198.12
16936  199833  L. Unnerstall  Lars Unnerstall  1990-07-20   28
198.12
17736  179783    R. Fährmann    Ralf Fährmann  1988-09-27   30
198.12
37     212190        N. Süle      Niklas Süle  1995-09-03   23
195.58
63     213331         J. Tah     Jonathan Tah  1996-02-11   23
195.58

       weight_kgs positions nationality  overall_rating  ...   LWB
LDM  \
8266        107.0        GK     Germany              62  ...   NaN
NaN
16936       103.0        GK     Germany              76  ...   NaN
NaN
17736        98.0        GK     Germany              82  ...   NaN
NaN
37           97.1        CB     Germany              84  ...  69+2
75+2
63           97.1        CB     Germany              82  ...  70+2
```

```
76+2

        CDM   RDM   RWB    LB   LCB    CB   RCB    RB
8266    NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
16936   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
17736   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
37     75+2  75+2  69+2  73+2  81+2  81+2  81+2  73+2
63     76+2  76+2  70+2  73+2  81+2  81+2  81+2  73+2

[5 rows x 92 columns]
```

```
Germany[['full_name','wage_euro']].sort_values(by=['wage_euro'],ascend
ing=False).head()
```

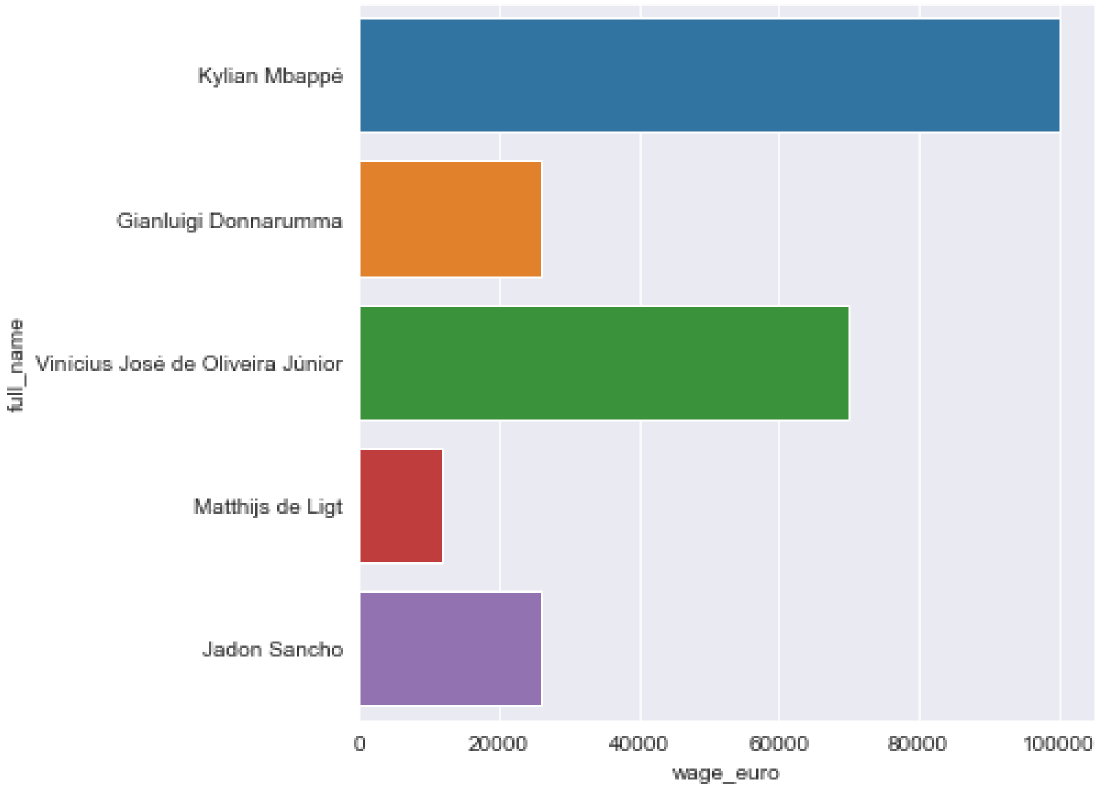
```
                   full_name  wage_euro
17934             Toni Kroos   355000.0
12     Marc-André ter Stegen   240000.0
17915             Leroy Sané   195000.0
17875             Mesut Özil   175000.0
17855         İlkay Gündoğan   175000.0
```

```
#Youngest palyers
player_age = fifa[['full_name','age']]
player_age.sort_values(by=['age']).head()
```

```
             full_name  age
9511        Adil Taoui   17
5637   Gilbert Fuentes   17
5636       Dylan Mbayo   17
5632  Helmer Andersson   17
5617    Adam Ben Lamin   17
```

```
#best youngest players
data_young =
fifa[fifa['age']<=20].sort_values(by=['potential','value_euro'],ascend
ing=[False,True])[0:5]
plt.figure(figsize=(6,6))
sns.barplot(y='full_name',x='wage_euro',data=data_young)
```

```
<AxesSubplot:xlabel='wage_euro', ylabel='full_name'>
```

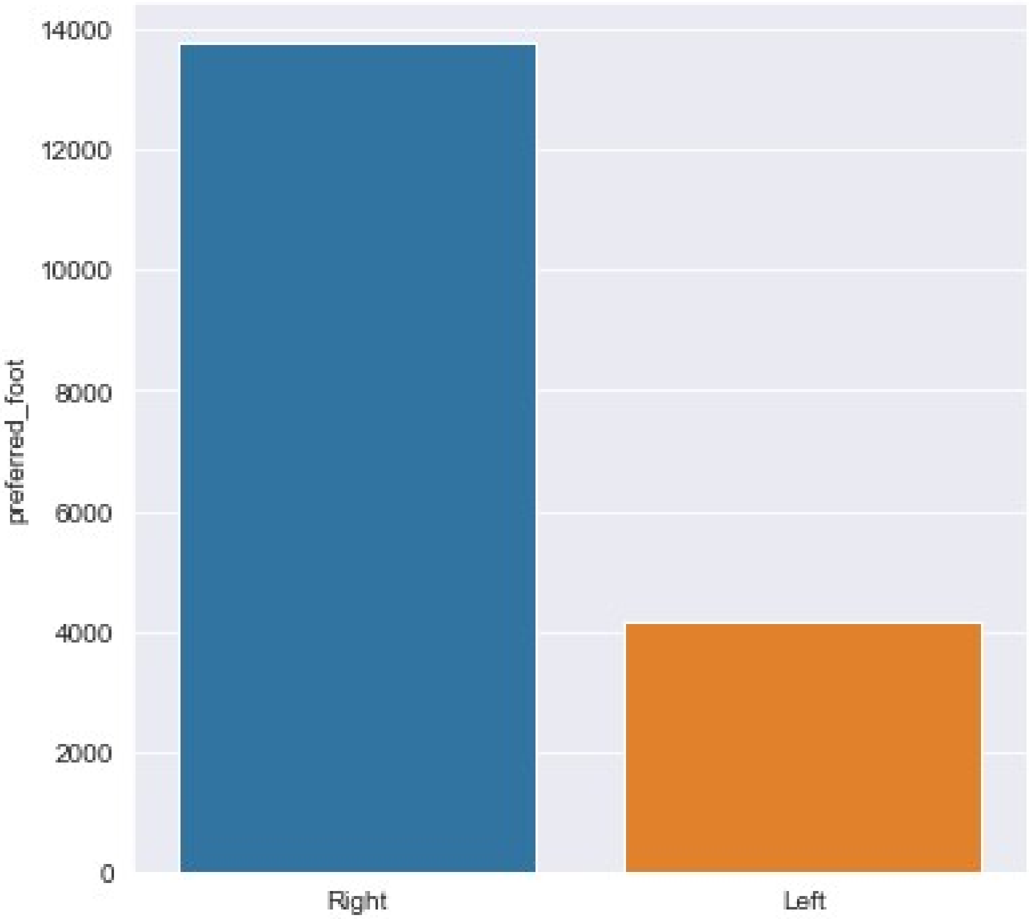


```
#Which foot do footballers prefer?

fifa['preferred_foot'].value_counts()
```

```
Right    13781
Left      4173
Name: preferred_foot, dtype: int64
```

```
#plotting preferred foot

plt.figure(figsize=(6,6))
sns.barplot(fifa['preferred_foot'].value_counts().keys(),fifa['preferr
ed_foot'].value_counts())
```

```
<AxesSubplot:ylabel='preferred_foot'>
```



```
fifa.isnull().sum()
```

```
id                0
name              0
full_name         0
birth_date        0
age               0
               ...
LB             2065
LCB            2065
CB             2065
RCB            2065
RB             2065
Length: 92, dtype: int64
```

```
fifa2=fifa.fillna(value=0)
```

```
fifa2.head()
```

```
       id          name                       full_name  birth_date
age  \
0  158023      L. Messi  Lionel Andrés Messi Cuccittini  1987-06-24
31
1  190460    C. Eriksen    Christian  Dannemann Eriksen  1992-02-14
27
2  195864      P. Pogba                      Paul Pogba  1993-03-15
25
3  198219    L. Insigne                 Lorenzo Insigne  1991-06-04
27
4  201024  K. Koulibaly               Kalidou Koulibaly  1991-06-20
27

   height_cm  weight_kgs  positions nationality  overall_rating  ...
LWB  \
0     170.18        72.1   CF,RW,ST   Argentina              94  ...
64+2
1     154.94        76.2  CAM,RM,CM     Denmark              88  ...
71+3
2     190.50        83.9     CM,CAM      France              88  ...
76+3
3     162.56        59.0      LW,ST       Italy              88  ...
63+3
4     187.96        88.9         CB     Senegal              88  ...
73+3

    LDM   CDM   RDM   RWB    LB   LCB    CB   RCB    RB
0  61+2  61+2  61+2  64+2  59+2  48+2  48+2  48+2  59+2
1  71+3  71+3  71+3  71+3  66+3  57+3  57+3  57+3  66+3
2  77+3  77+3  77+3  76+3  74+3  72+3  72+3  72+3  74+3
3  58+3  58+3  58+3  63+3  58+3  44+3  44+3  44+3  58+3
4  77+3  77+3  77+3  73+3  76+3  85+3  85+3  85+3  76+3

[5 rows x 92 columns]
```

```
from sklearn import preprocessing

# label_encoder object knows how to understand word labels.
label_encoder = preprocessing.LabelEncoder()

# Encode labels in column
fifa2['nationality']=
label_encoder.fit_transform(fifa2['nationality'])

fifa2['nationality'].unique()
```

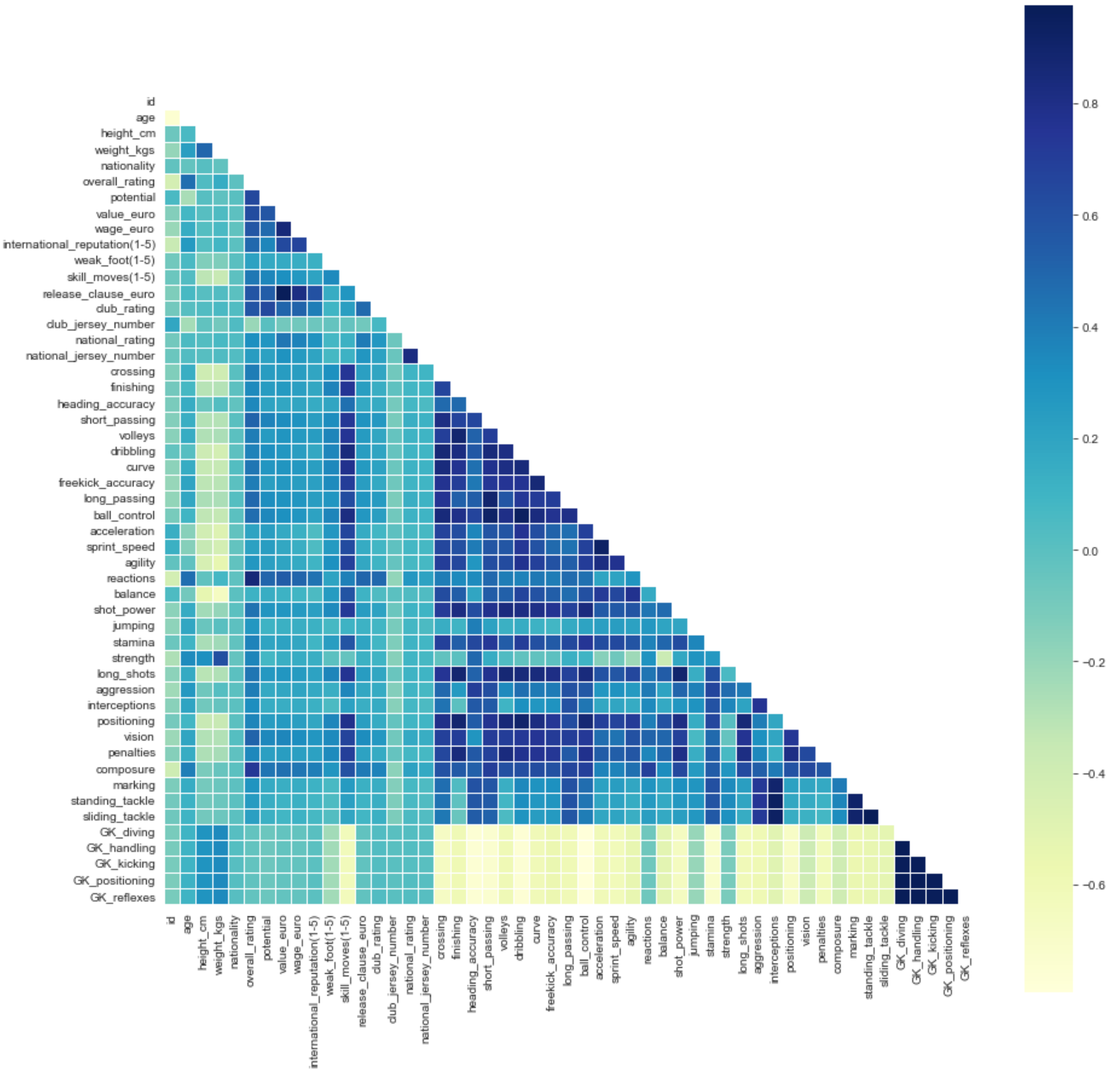
```
array([  6,  39,  53,  76, 126, 104,  57, 152, 133,  13,  42, 129,
18,
        33,  32,  29, 101, 119, 138,  17,  97,  44,   9,  70,  69,
156,
       149,  25, 127,  77,  22, 115,   8,   2, 121, 122,  75, 139,
27,
       146, 147, 108, 116, 110,  59, 151, 154,  73,  45,  24, 142,
125,
        28,  85,  99,  23,  93,  94,  58,  63, 118,  34, 109,  79,
106,
       131, 120,  41,  20,  37, 130,  12,  54,  49,  35,  38,  67,
128,
        64, 124,  51,  84, 158, 140,   7,  56,   4, 159,  31,  46,
74,
         1,  16,  55,  78,  21, 148,  14, 137,  52,  91, 144,  43,
83,
       113,  98,  66,  30,  10, 145,  88,  19,  82,  26,  50,  81,
47,
       100,  40, 143,  71,   3, 136,  68,  95,   0, 153, 112, 157,
92,
        96,  72,  15,  60, 117,  90,  65, 134,   5,  36, 123, 132,
114,
        87, 155, 107,  48,  11,  80, 135,  62, 103,  61, 102,  86,
89,
       105, 111, 141, 150])
```

```
fifa2['overall_rating']=
label_encoder.fit_transform(fifa2['overall_rating'])
```

```
fifa2['overall_rating'].unique()
```

```
array([46, 41, 42, 39, 40, 37, 38, 36, 35, 34, 33, 32, 31, 30, 29, 28,
27,
       26, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11,
10,
        9,  8,  7,  6,  5,  4,  3,  2,  0,  1, 43, 44, 45],
dtype=int64)
```

```
corr = fifa2.corr()
mask = np.zeros_like(corr)
mask[np.triu_indices_from(mask)] = True
with sns.axes_style("white"):
    f, ax = plt.subplots(figsize=(15, 15))
    ax =
sns.heatmap(corr,mask=mask,square=True,linewidths=.8,cmap="YlGnBu")
```



```
def correlation(dataset, threshold):#X_train,0.7
    col_corr = set()  # Set of all the names of correlated columns
    corr_matrix = dataset.corr()
    for i in range(len(corr_matrix)): #traverse through the rows
        for j in range(i): #traverse through column
            if abs(corr_matrix.iloc[i, j]) > threshold: # we are interested in absolute coeff value
                colname = corr_matrix.columns[i]  # getting the name of column
                col_corr.add(colname)
    return col_corr

corr_features = correlation(fifa2, 0.3)
len((corr_features))
```

```
46
```

```
fifa2.select_dtypes('object').columns
```

```
Index(['name', 'full_name', 'birth_date', 'positions',
'preferred_foot',
       'work_rate', 'body_type', 'club_team', 'club_position',
       'club_join_date', 'contract_end_year', 'national_team',
       'national_team_position', 'tags', 'traits', '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'],
      dtype='object')
```

```
fifa2.drop(fifa2.select_dtypes('object').columns,inplace=True,axis=1)
```

```
from sklearn.feature_selection import VarianceThreshold
var_thres=VarianceThreshold(threshold=0.5)
var_thres.fit(fifa2)
```

```
VarianceThreshold(threshold=0.5)
```

```
var_thres.get_support()
```

```
array([ True,  True,  True,  True,  True,  True,  True,  True,  True,
       False, False,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True])
```

```
fifa2.columns[var_thres.get_support() == True]
```

```
Index(['id', 'age', 'height_cm', 'weight_kgs', 'nationality',
'overall_rating',
       'potential', 'value_euro', 'wage_euro', 'skill_moves(1-5)',
       'release_clause_euro', 'club_rating', 'club_jersey_number',
       'national_rating', 'national_jersey_number', 'crossing',
'finishing',
       'heading_accuracy', 'short_passing', 'volleys', 'dribbling',
'curve',
       'freekick_accuracy', 'long_passing', 'ball_control',
'acceleration',
       'sprint_speed', 'agility', 'reactions', 'balance',
'shot_power',
       'jumping', 'stamina', 'strength', 'long_shots', 'aggression',
       'interceptions', 'positioning', 'vision', 'penalties',
'composure',
       'marking', 'standing_tackle', 'sliding_tackle', 'GK_diving',
       'GK_handling', 'GK_kicking', 'GK_positioning', 'GK_reflexes'],
      dtype='object')
```

```
columns_having_var_more_than_50 =
fifa2.columns[var_thres.get_support() == True]
```

```
len(columns_having_var_more_than_50)
```

```
49
```

```
len(fifa2.columns)
```

```
51
```

```
fifa2.columns[var_thres.get_support() == False]
```

```
Index(['international_reputation(1-5)', 'weak_foot(1-5)'],
dtype='object')
```

```
columns_having_var_less_than_50 =
fifa2.columns[var_thres.get_support() == False]
```

```
len(columns_having_var_less_than_50)
```

```
2
```

```
fifa2.drop(columns_having_var_less_than_50,inplace = True,axis= 1)
```

```
fifa2.head(2)
```

```
       id  age  height_cm  weight_kgs  nationality  overall_rating
potential  \
0  158023   31     170.18        72.1            6              46
94
1  190460   27     154.94        76.2           39              41
89

    value_euro  wage_euro  skill_moves(1-5)  ...  penalties  composure
\
0  110500000.0   565000.0                 4  ...         75         96

1   69500000.0   205000.0                 4  ...         67         88


   marking  standing_tackle  sliding_tackle  GK_diving  GK_handling  \
0       33               28              26          6           11
1       59               57              22          9           14

   GK_kicking  GK_positioning  GK_reflexes
0          15              14            8
1           7               7            6

[2 rows x 49 columns]
```

```
fifa2.columns
```

```
Index(['id', 'age', 'height_cm', 'weight_kgs', 'nationality',
'overall_rating',
       'potential', 'value_euro', 'wage_euro', 'skill_moves(1-5)',
       'release_clause_euro', 'club_rating', 'club_jersey_number',
       'national_rating', 'national_jersey_number', 'crossing',
'finishing',
       'heading_accuracy', 'short_passing', 'volleys', 'dribbling',
'curve',
       'freekick_accuracy', 'long_passing', 'ball_control',
'acceleration',
       'sprint_speed', 'agility', 'reactions', 'balance',
'shot_power',
       'jumping', 'stamina', 'strength', 'long_shots', 'aggression',
       'interceptions', 'positioning', 'vision', 'penalties',
'composure',
       'marking', 'standing_tackle', 'sliding_tackle', 'GK_diving',
       'GK_handling', 'GK_kicking', 'GK_positioning', 'GK_reflexes'],
      dtype='object')
```