

```
In [126]. import pandas as pd
from pandas.plotting import scatter_matrix
import numpy as np
from numpy import percentile
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn import svm
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestRegressor

In [127]. file=pd.read_csv("loan_prediction.csv")

In [128]. file

Out[128].
   Loan_ID  Gender  Married  Dependents  Education  Self_Employed  ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  Credit_History  Property_Area  Loan_Status
0  LP001002    Male    No         0      Graduate      No          5849              0.0         NaN          360.0          1.0      Urban      Y
1  LP001003    Male    Yes         1      Graduate      No          4583             1508.0        128.0          360.0          1.0      Rural      N
2  LP001005    Male    Yes         0      Graduate      Yes          3000              0.0         66.0          360.0          1.0      Urban      Y
3  LP001006    Male    Yes         0      Not Graduate  No          2583             2358.0        120.0          360.0          1.0      Urban      Y
4  LP001008    Male    No         0      Graduate      No          6000              0.0        141.0          360.0          1.0      Urban      Y
...
609 LP002978    Female   No         0      Graduate      No          2900              0.0         71.0          360.0          1.0      Rural      Y
610 LP002979    Male    Yes         3+    Graduate      No          4106              0.0         40.0          180.0          1.0      Rural      Y
611 LP002983    Male    Yes         1      Graduate      No          8072             240.0        253.0          360.0          1.0      Urban      Y
612 LP002984    Male    Yes         2      Graduate      No          7583              0.0        187.0          360.0          1.0      Urban      Y
613 LP002990    Female   No         0      Graduate      Yes          4583              0.0        133.0          360.0          0.0     Semiurban      N
614 rows x 13 columns

In [129]. file.shape

Out[129]. (614, 13)

In [130]. file.head()

Out[130].
   Loan_ID  Gender  Married  Dependents  Education  Self_Employed  ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  Credit_History  Property_Area  Loan_Status
0  LP001002    Male    No         0      Graduate      No          5849              0.0         NaN          360.0          1.0      Urban      Y
1  LP001003    Male    Yes         1      Graduate      No          4583             1508.0        128.0          360.0          1.0      Rural      N
2  LP001005    Male    Yes         0      Graduate      Yes          3000              0.0         66.0          360.0          1.0      Urban      Y
3  LP001006    Male    Yes         0      Not Graduate  No          2583             2358.0        120.0          360.0          1.0      Urban      Y
4  LP001008    Male    No         0      Graduate      No          6000              0.0        141.0          360.0          1.0      Urban      Y

In [131]. file.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
 #   Column              Non-Null Count  Dtype
---  --
 0   Loan_ID             614 non-null    object
 1   Gender              601 non-null    object
 2   Married             611 non-null    object
 3   Dependents          599 non-null    object
 4   Education           614 non-null    object
 5   Self_Employed       582 non-null    object
 6   ApplicantIncome     614 non-null    int64
 7   CoapplicantIncome   614 non-null    float64
 8   LoanAmount          582 non-null    float64
 9   Loan_Amount_Term    609 non-null    float64
10   Credit_History       564 non-null    float64
11   Property_Area       614 non-null    object
12   Loan_Status         614 non-null    object
dtypes: float64(4), int64(1), object(8)
memory usage: 62.5+ KB

In [132]. file.isnull().sum()

Out[132].
Loan_ID      0
Gender       13
Married       3
Dependents   15
Education     0
Self_Employed 32
ApplicantIncome 0
CoapplicantIncome 0
LoanAmount   22
Loan_Amount_Term 14
Credit_History 50
Property_Area 0
Loan_Status  0
dtype: int64

In [133]. file.drop("Loan_ID", axis =1, inplace =True)

In [134]. file

Out[134].
   Gender  Married  Dependents  Education  Self_Employed  ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  Credit_History  Property_Area  Loan_Status
0      Male     No         0      Graduate      No          5849              0.0         NaN          360.0          1.0      Urban      Y
1      Male     Yes         1      Graduate      No          4583             1508.0        128.0          360.0          1.0      Rural      N
2      Male     Yes         0      Graduate      Yes          3000              0.0         66.0          360.0          1.0      Urban      Y
3      Male     Yes         0      Not Graduate  No          2583             2358.0        120.0          360.0          1.0      Urban      Y
4      Male     No         0      Graduate      No          6000              0.0        141.0          360.0          1.0      Urban      Y
...
609  Female     No         0      Graduate      No          2900              0.0         71.0          360.0          1.0      Rural      Y
610  Male      Yes         3+    Graduate      No          4106              0.0         40.0          180.0          1.0      Rural      Y
611  Male      Yes         1      Graduate      No          8072             240.0        253.0          360.0          1.0      Urban      Y
612  Male      Yes         2      Graduate      No          7583              0.0        187.0          360.0          1.0      Urban      Y
613  Female     No         0      Graduate      Yes          4583              0.0        133.0          360.0          0.0     Semiurban      N
614 rows x 12 columns

In [135]. file.describe()

Out[135].
      ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  Credit_History
count      614.000000          614.000000      582.000000          600.000000      564.000000
mean      5403.459283          1621.245798      146.412162          342.000000      0.842199
std       6109.041673          2926.248369      85.587325          65.12041         0.364878
min       150.000000           0.000000       9.000000          12.000000      0.000000
25%      2877.500000           0.000000      100.000000          360.000000      1.000000
50%      3812.500000          1188.500000      128.000000          360.000000      1.000000
75%      5795.000000          2297.250000      168.000000          360.000000      1.000000
max      81000.000000        41667.000000      700.000000          480.000000      1.000000

In [136]. file.isnull().sum()

Out[136].
Gender       13
Married       3
Dependents   15
Education     0
Self_Employed 32
ApplicantIncome 0
CoapplicantIncome 0
LoanAmount   22
Loan_Amount_Term 14
Credit_History 50
Property_Area 0
Loan_Status  0
dtype: int64

In [137]. # dropping all the null values
file = file.dropna()

In [138]. file.isnull().sum()

Out[138].
Gender       0
Married       0
Dependents    0
Education     0
Self_Employed 0
ApplicantIncome 0
CoapplicantIncome 0
LoanAmount    0
Loan_Amount_Term 0
Credit_History 0
Property_Area 0
Loan_Status   0
dtype: int64

In [139]. scatter_matrix(file, figsize=(15,15),diagonal="hist",color="black")
plt.show()

In [140]. file.head()

Out[140].
   Gender  Married  Dependents  Education  Self_Employed  ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  Credit_History  Property_Area  Loan_Status
1      Male     Yes         1      Graduate      No          4583             1508.0        128.0          360.0          1.0      Rural      N
2      Male     Yes         0      Graduate      Yes          3000              0.0         66.0          360.0          1.0      Urban      Y
3      Male     Yes         0      Not Graduate  No          2583             2358.0        120.0          360.0          1.0      Urban      Y
4      Male     No         0      Graduate      No          6000              0.0        141.0          360.0          1.0      Urban      Y
5      Male     Yes         2      Graduate      Yes          5417             4196.0        267.0          360.0          1.0      Urban      Y

In [141]. file.replace({'Married':{'No':0,'Yes':1},'Gender':{'Male':0,'Female':1},'Self_Employed':{'No':0,'Yes':1},'Loan_Status':{'N':0,'Y':1},'Dependents':{'0':0,'1':1,'2':2,'3'+':3'},
'Property_Area':{'Rural':1,'Semiurban':2,'Urban':0},'Education':{'Graduate':1,'Not Graduate':0}}, inplace =True)

In [142]. file["Dependents"].value_counts()

Out[142].
0    274
2     85
1     80
3     41
Name: Dependents, dtype: int64

In [143]. file.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 480 entries, 1 to 613
Data columns (total 12 columns):
 #   Column              Non-Null Count  Dtype
---  --
 0   Gender              480 non-null    int64
 1   Married             480 non-null    int64
 2   Dependents          480 non-null    int64
 3   Education           480 non-null    int64
 4   Self_Employed       480 non-null    int64
 5   ApplicantIncome     480 non-null    int64
 6   CoapplicantIncome   480 non-null    float64
 7   LoanAmount          480 non-null    float64
 8   Loan_Amount_Term    480 non-null    float64
 9   Credit_History       480 non-null    float64
10   Property_Area       480 non-null    int64
11   Loan_Status         480 non-null    int64
dtypes: float64(4), int64(8)
memory usage: 48.8 KB

In [145]. scatter_matrix(file, figsize=(20,20),diagonal="hist",color="black")
plt.show()

In [95]. plt.figure(figsize=(20,20),dpi=90)
sns.boxplot(data=file[['Gender','Married','Dependents','Education','Self_Employed','ApplicantIncome','CoapplicantIncome','LoanAmount','Loan_Amount_Term','Credit_History','Property_Area','Loan_Status']])
plt.show()

In [109]. sns.boxplot(x="Loan_Status", y ="ApplicantIncome", data =file)
plt.show()

In [ ]: # building module

In [96]: x =file.drop("Loan_Status",axis =1)
y =file["Loan_Status"]

In [97]: print(x)
print(y)

In [98]:
   Gender  Married  Dependents  Education  Self_Employed  ApplicantIncome  \
1      0      0      1          1          1          1
2      0      1          0          1          1          1
3      0      0          0          1          0          0
4      0      0          0          1          0          0
5      0      1          2          1          1          1
...
609    1      0          0          1          1          0
610    0      1          3          1          0          0
611    0      1          1          1          0          0
612    0      1          2          1          0          0
613    1      0          0          1          1          1

   CoapplicantIncome  LoanAmount  Loan_Amount_Term  Credit_History  \
1             1508.0         128.0              0          1.0
2              66.0           360.0              1          1.0
3             2358.0         120.0              1          1.0
4             141.0           360.0              1          1.0
5             4196.0          267.0              1          1.0
...
609              0.0           71.0             1          1.0
610              0.0           40.0             0          1.0
611             240.0          253.0             1          1.0
612              0.0           187.0            0          1.0
613              0.0           133.0            0          0.0

   Property_Area
1              0
2              0
3              0
4              0
5              0
...
609            1
610            1
611            0
612            0
613            2
Name: Loan_Status, Length: 480, dtype: int64

In [98]: x_train,x_train_y,train_y, test=train_test_split(x,y, test_size=0.1,random_state=2)

In [99]: print(x.shape,x_train.shape,x_test.shape)

(480, 11) (432, 11) (48, 11)

In [100]. clas =svm.SVC(kernel='linear')

In [101]. clas.fit(x_train,y_train)

Out[101]. SVC(kernel='linear')

In [102]. # train data prediction
x_train_predic = clas.predict(x_train)
x_data_accuracy =accuracy_score(x_train_predic ,y_train)

In [103]. print("Accuracy of x_data :",x_data_accuracy)

Accuracy of x_data : 0.7731481481481481

In [105]. # test data prediction
x_test_predic = clas.predict(x_test)
x_data1_accuracy =accuracy_score(x_test_predic ,y_test)

In [107]. print("Accuracy of x_data1 :",x_data1_accuracy)

Accuracy of x_data1 : 0.7916666666666666

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:
```