In [2]:
```
import numpy as np
import pandas as pd
from collections import Counter
import warnings
warnings.filterwarnings('ignore')
```

In [5]:
```
train_input=pd.read_csv('Credit_Risk_Train_Data.csv')
test_input=pd.read_csv('Credit_Risk_Validate_Data.csv')
```

In [6]:
```
print(train_input.columns)
print(test_input.columns)
```

```
Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
       'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
       'Loan_Amount_Term', 'Credit_History', 'Property_Area', 'Loan_Status'],
      dtype='object')
Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
       'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
       'Loan_Amount_Term', 'Credit_History', 'Property_Area', 'outcome'],
      dtype='object')
```

In [7]:
```
#the lasst columns has a differnt name in both ,
#Lest make the names same. and the merge them together
#so that we can fill the missing value simultaneusly
test_input.rename(columns={'outcome':"Loan_Status"},inplace=True)
```

In [8]:
```
print(test_input.columns)
```

```
Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
       'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
       'Loan_Amount_Term', 'Credit_History', 'Property_Area', 'Loan_Status'],
      dtype='object')
```

In [9]:
```
data_all=pd.concat([train_input,test_input],axis=0)
```

In [10]:
```
print(data_all)
```

```
      Loan_ID Gender Married Dependents     Education Self_Employed  \
0    LP001002   Male      No          0      Graduate            No
1    LP001003   Male     Yes          1      Graduate            No
2    LP001005   Male     Yes          0      Graduate           Yes
3    LP001006   Male     Yes          0  Not Graduate            No
4    LP001008   Male      No          0      Graduate            No
..        ...    ...     ...        ...           ...           ...
362  LP002971   Male     Yes         3+  Not Graduate           Yes
363  LP002975   Male     Yes          0      Graduate            No
364  LP002980   Male      No          0      Graduate            No
365  LP002986   Male     Yes          0      Graduate            No
366  LP002989   Male      No          0      Graduate           Yes

     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
0               5849                0.0         NaN             360.0
1               4583             1508.0       128.0             360.0
2               3000                0.0        66.0             360.0
3               2583             2358.0       120.0             360.0
4               6000                0.0       141.0             360.0
..               ...                ...         ...               ...
362             4009             1777.0       113.0             360.0
363             4158              709.0       115.0             360.0
364             3250             1993.0       126.0             360.0
365             5000             2393.0       158.0             360.0
366             9200                0.0        98.0             180.0

     Credit_History Property_Area Loan_Status
0               1.0         Urban           Y
1               1.0         Rural           N
2               1.0         Urban           Y
3               1.0         Urban           Y
4               1.0         Urban           Y
..              ...           ...         ...
362             1.0         Urban           Y
363             1.0         Urban           Y
364             NaN     Semiurban           Y
365             1.0         Rural           N
366             1.0         Rural           Y

[981 rows x 13 columns]
```

In [11]:
```
data_all.shape
```

Out[11]:
```
(981, 13)
```

In [12]:
```
data_all.reset_index(inplace=True,drop=True)
```

In [13]:
```
print(data_all)
```

```
      Loan_ID Gender Married Dependents     Education Self_Employed  \
0    LP001002   Male      No          0      Graduate            No
1    LP001003   Male     Yes          1      Graduate            No
2    LP001005   Male     Yes          0      Graduate           Yes
3    LP001006   Male     Yes          0  Not Graduate            No
4    LP001008   Male      No          0      Graduate            No
..        ...    ...     ...        ...           ...           ...
976  LP002971   Male     Yes         3+  Not Graduate           Yes
977  LP002975   Male     Yes          0      Graduate            No
978  LP002980   Male      No          0      Graduate            No
979  LP002986   Male     Yes          0      Graduate            No
980  LP002989   Male      No          0      Graduate           Yes

     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
0               5849                0.0         NaN             360.0
1               4583             1508.0       128.0             360.0
2               3000                0.0        66.0             360.0
3               2583             2358.0       120.0             360.0
4               6000                0.0       141.0             360.0
..               ...                ...         ...               ...
976             4009             1777.0       113.0             360.0
977             4158              709.0       115.0             360.0
978             3250             1993.0       126.0             360.0
979             5000             2393.0       158.0             360.0
980             9200                0.0        98.0             180.0

     Credit_History Property_Area Loan_Status
0               1.0         Urban           Y
1               1.0         Rural           N
2               1.0         Urban           Y
3               1.0         Urban           Y
4               1.0         Urban           Y
..              ...           ...         ...
976             1.0         Urban           Y
977             1.0         Urban           Y
978             NaN     Semiurban           Y
979             1.0         Rural           N
980             1.0         Rural           Y

[981 rows x 13 columns]
```

In [14]:
```
print(data_all.tail())
```

```
      Loan_ID Gender Married Dependents     Education Self_Employed  \
976  LP002971   Male     Yes         3+  Not Graduate           Yes
977  LP002975   Male     Yes          0      Graduate            No
978  LP002980   Male      No          0      Graduate            No
979  LP002986   Male     Yes          0      Graduate            No
980  LP002989   Male      No          0      Graduate           Yes

     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
976             4009             1777.0       113.0             360.0
977             4158              709.0       115.0             360.0
978             3250             1993.0       126.0             360.0
979             5000             2393.0       158.0             360.0
980             9200                0.0        98.0             180.0

     Credit_History Property_Area Loan_Status
976             1.0         Urban           Y
977             1.0         Urban           Y
978             NaN     Semiurban           Y
979             1.0         Rural           N
980             1.0         Rural           Y
```

In [15]:
```
data_all.isnull().sum()
```

Out[15]:
```
Loan_ID               0
Gender               24
Married               3
Dependents           25
Education             0
Self_Employed        55
ApplicantIncome       0
CoapplicantIncome     0
LoanAmount           27
Loan_Amount_Term     20
Credit_History       79
Property_Area         0
Loan_Status           0
dtype: int64
```

In [16]:
```
data_all.shape
```

Out[16]:
```
(981, 13)
```

In [17]:
```
Counter(data_all['Gender'])
```

Out[17]:
```
Counter({'Male': 775, 'Female': 182, nan: 24})
```

In [18]:
```
#lets fill them with the Model of gender i.e Male
gender_null=data_all[data_all['Gender'].isnull()].index.tolist()
print(gender_null)
```

```
[23, 126, 171, 188, 314, 334, 460, 467, 477, 507, 576, 588, 592, 636, 665, 720,
752, 823, 845, 859, 893, 910, 917, 932]
```

In [19]:
```
gender_null_M=gender_null[:12]
print(gender_null_M)
```

```
[23, 126, 171, 188, 314, 334, 460, 467, 477, 507, 576, 588]
```

In [20]:
```
gender_null_F=gender_null[12:]
print(gender_null_F)
```

```
[592, 636, 665, 720, 752, 823, 845, 859, 893, 910, 917, 932]
```

In [21]:
```
data_all['Gender'].iloc[gender_null]='Male'
```

In [22]:
```
#check if  filed
print(sum(data_all['Gender'].isnull()))
```

```
0
```

In [23]:
```
Counter(data_all['Gender'])
```

Out[23]:
```
Counter({'Male': 799, 'Female': 182})
```

In [24]:
```
Counter(data_all['Married'])
```

Out[24]:
```
Counter({'No': 347, 'Yes': 631, nan: 3})
```

In [25]:
```
married_null=data_all[data_all['Married'].isnull()].index.tolist()
```

In [26]:
```
print(married_null)
```

```
[104, 228, 435]
```

In [68]:
```
data_all['Married'].iloc[married_null]='Yes'
```

In [69]:
```
data_all.isnull().sum()
```

Out[69]:
```
Loan_ID               0
Gender                0
Married               0
Dependents            0
Education             0
Self_Employed         0
ApplicantIncome       0
CoapplicantIncome     0
LoanAmount            0
Loan_Amount_Term      0
Credit_History       79
Property_Area         0
Loan_Status           0
dtype: int64
```

In [70]:
```
Counter(data_all['Dependents'])
```

Out[70]:
```
Counter({'0': 554, '1': 176, '2': 160, '3+': 91})
```

In [71]:
```
dep_null=data_all[data_all['Dependents'].isnull()].index.tolist()
```

In [72]:
```
print(dep_null)
```

```
[]
```

In [73]:
```
#let see the Dependants wrt married
pd.crosstab(data_all['Married'],data_all['Dependents'].isnull())
```

Out[73]:

Dependents	False
Married	
No	347
Yes	634

In [74]:
```
pd.crosstab(data_all['Married'],data_all['Dependents'])
```

Out[74]:

Dependents	0	1	2	3+
Married				
No	285	36	14	12
Yes	269	140	146	79

In [75]:
```
#for the bachelors ,lets fill the missing dependents as 0
#lets find the index of all rows with Dependents missing and Married No
bachelor_nulldependent=data_all[(data_all['Married']=='No')&(data_all['Dependents
print(bachelor_nulldependent)
```

```
[]
```

In [76]:
```
data_all['Dependents'].iloc[[bachelor_nulldependent]]='0'
```

In [77]:
```
Counter(data_all['Dependents'])
```

Out[77]:
```
Counter({'0': 554, '1': 176, '2': 160, '3+': 91})
```

In [78]:
```
#For the remaining 16 missing dependents
#lets see how many dependents Male & Female have
pd.crosstab(data_all["Gender"],data_all['Dependents'].isnull())
```

Out[78]:

Dependents	False
Gender	
Female	182
Male	799

In [79]:
```
pd.crosstab((data_all['Gender']=='Male')& (data_all['Married']=='Yes'),data_all['
```

Out[79]:

Dependents	False
row_0	
False	405
True	576

In [80]:
```
data_all['Dependents'].iloc[data_all[data_all['Dependents'].isnull()].index.tolis
```

In [81]:
```
data_all.isnull().sum()
```

Out[81]:
```
Loan_ID               0
Gender                0
Married               0
Dependents            0
Education             0
Self_Employed         0
ApplicantIncome       0
CoapplicantIncome     0
LoanAmount            0
Loan_Amount_Term      0
Credit_History       79
Property_Area         0
Loan_Status           0
dtype: int64
```

In [82]:
```
Counter(data_all['Self_Employed'])
```

Out[82]:
```
Counter({'No': 862, 'Yes': 119})
```

In [83]:
```
self_emp_null=data_all[data_all['Self_Employed'].isnull()].index.tolist()
```

In [84]:
```
print(self_emp_null)
```

```
[]
```

In [85]:
```
data_all['Self_Employed'].iloc[self_emp_null]='No'
```

In [86]:
```
Counter(data_all['Self_Employed'])
```

Out[86]:
```
Counter({'No': 862, 'Yes': 119})
```

In [87]:
```
data_all.isnull().sum()
```

Out[87]:
```
Loan_ID               0
Gender                0
Married               0
Dependents            0
Education             0
Self_Employed         0
ApplicantIncome       0
CoapplicantIncome     0
LoanAmount            0
Loan_Amount_Term      0
Credit_History       79
Property_Area         0
Loan_Status           0
dtype: int64
```

In [88]:
```
pd.crosstab(data_all['LoanAmount'].isnull(),data_all['Loan_Amount_Term'])
```

Out[88]:

Loan_Amount_Term	6.0	12.0	36.0	60.0	84.0	120.0	180.0	240.0	300.0	350.0	360.0	480.0
LoanAmount												
False	1	2	3	3	7	4	66	8	20	1	843	23

In [89]:
```
data_all.groupby(data_all['Loan_Amount_Term'])['LoanAmount'].mean()
```

Out[89]:
```
Loan_Amount_Term
6.0       95.000000
12.0     185.500000
36.0     117.666667
60.0     139.666667
84.0     121.142857
120.0     36.750000
180.0    131.090909
240.0    129.000000
300.0    166.250000
350.0    133.000000
360.0    143.849348
480.0    137.173913
Name: LoanAmount, dtype: float64
```

In [90]:
```
#lets fill the missing values in loan amount
#With the mean of respedtive Loan_Term
data_all['LoanAmount'][(data_all['LoanAmount'].isnull()) & (data_all["Loan_Amount
```

In [91]:
```
data_all['LoanAmount'][(data_all['LoanAmount'].isnull()) & (data_all["Loan_Amount
```

In [92]:
```
data_all['LoanAmount'][(data_all['LoanAmount'].isnull())]=130
```

In [93]:
```
(data_all['Loan_Amount_Term']).value_counts()
```

Out[93]:
```
360.0    843
180.0     66
480.0     23
300.0     20
240.0      8
84.0       7
120.0      4
60.0       3
36.0       3
12.0       2
350.0      1
6.0        1
Name: Loan_Amount_Term, dtype: int64
```

In [94]:
```
data_all["Loan_Amount_Term"][data_all['Loan_Amount_Term'].isnull()]=360
```

In [95]:
```
data_all.isnull().sum()
```

Out[95]:
```
Loan_ID               0
Gender                0
Married               0
Dependents            0
Education             0
Self_Employed         0
ApplicantIncome       0
CoapplicantIncome     0
LoanAmount            0
Loan_Amount_Term      0
Credit_History       79
Property_Area         0
Loan_Status           0
dtype: int64
```

In [96]:
```
Counter(data_all['Credit_History'])
```

Out[96]:
```
Counter({1.0: 754,
         0.0: 148,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
         nan: 1,
```

```
             nan: 1,
             nan: 1,
             nan: 1,
             nan: 1,
             nan: 1,
             nan: 1,
             nan: 1,
             nan: 1,
             nan: 1,
             nan: 1,
             nan: 1,
             nan: 1,
             nan: 1,
             nan: 1,
             nan: 1,
             nan: 1,
             nan: 1,
             nan: 1,
             nan: 1,
             nan: 1,
             nan: 1,
             nan: 1,
             nan: 1,
             nan: 1,
             nan: 1,
             nan: 1,
             nan: 1})
```

In [97]:
```
data_all['Credit_History'].value_counts()
```

Out[97]:
```
1.0    754
0.0    148
Name: Credit_History, dtype: int64
```

In [98]:
```
pd.crosstab(data_all['Education'],data_all['Credit_History'])
```

Out[98]:

Credit_History	0.0	1.0
Education		
Graduate	106	596
Not Graduate	42	158

In [99]:
```
pd.crosstab(data_all['Married'],data_all['Credit_History'])
```

Out[99]:

Credit_History	0.0	1.0
Married		
No	56	263
Yes	92	491

In [100]:
```
data_all['Credit_History'][data_all['Credit_History'].isnull()]=1
```

In [101]:
```
data_all.isnull().sum()
```

Out[101]:
```
Loan_ID              0
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 [103]:
```
data_all.head()
```

Out[103]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplica
0	LP001002	Male	No	0	Graduate	No	5849	
1	LP001003	Male	Yes	1	Graduate	No	4583	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	
3	LP001006	Male	Yes	0	Not Graduate	No	2583	
4	LP001008	Male	No	0	Graduate	No	6000	

In [105]:
```
data_all_new=pd.get_dummies(data_all.drop(['Loan_ID'],axis=1),drop_first=True)
```

In [107]:
```
data_all_new.head()
```

Out[107]:

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Gender_M
0	5849	0.0	144.0	360.0	1.0	
1	4583	1508.0	128.0	360.0	1.0	
2	3000	0.0	66.0	360.0	1.0	
3	2583	2358.0	120.0	360.0	1.0	
4	6000	0.0	141.0	360.0	1.0	

In [112]:
```
X=data_all_new.drop(['Loan_Status_Y'],axis=1)
y=data_all_new['Loan_Status_Y']
```

In [113]:
```
X.head()
```

Out[113]:

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Gender_M
0	5849	0.0	144.0	360.0	1.0	
1	4583	1508.0	128.0	360.0	1.0	
2	3000	0.0	66.0	360.0	1.0	
3	2583	2358.0	120.0	360.0	1.0	
4	6000	0.0	141.0	360.0	1.0	

In [114]:
```
y.head()
```

Out[114]:
```
0    1
1    0
2    1
3    1
4    1
Name: Loan_Status_Y, dtype: uint8
```

Train Test Split

In [115]:
```
from sklearn.model_selection import train_test_split
```

In [118]:
```
X_train ,X_test,y_train,y_test=train_test_split(X,y)
```

In [119]:
```
X_train.shape
```

Out[119]:
```
(735, 14)
```

In [120]:
```
X_test.shape
```

Out[120]:
```
(246, 14)
```

Data Preprocessing

In [124]:
```
from sklearn.preprocessing import StandardScaler
```

In [125]:
```
scaler=StandardScaler()
```

In [126]:
```
#fit only the training data
scaler.fit(X)
```

Out[126]:
```
StandardScaler()
```

In [130]:
```
#now apply the transformation to the dat
X_train=scaler.transform(X_train)
X_test=scaler.transform(X_test)
```

In [132]:
```
X_train[:5]
```

Out[132]:
```
array([[-0.70953135,  0.15275105, -0.29487359,  0.2705276 , -2.37242036,
         0.47726799, -1.35169869, -0.46758266,  2.26522626, -0.31976115,
         1.87082869, -0.37155221, -0.74311183,  1.36690199],
       [-0.39032267,  0.47769559, -0.15072949,  0.2705276 ,  0.42151046,
         0.47726799,  0.73980985, -0.46758266, -0.44145701, -0.31976115,
        -0.53452248, -0.37155221,  1.34569248, -0.73158135],
       [-0.11573644, -0.15379347,  0.09824667,  0.2705276 ,  0.42151046,
         0.47726799,  0.73980985, -0.46758266,  2.26522626, -0.31976115,
         1.87082869, -0.37155221,  1.34569248, -0.73158135],
       [-0.25171196,  0.19764712, -0.22935354,  0.2705276 ,  0.42151046,
         0.47726799,  0.73980985,  2.13865931, -0.44145701, -0.31976115,
        -0.53452248, -0.37155221, -0.74311183,  1.36690199],
       [ 1.48505029, -0.5895062 ,  1.80176779, -1.59140579,  0.42151046,
         0.47726799,  0.73980985, -0.46758266,  2.26522626, -0.31976115,
        -0.53452248, -0.37155221, -0.74311183,  1.36690199]])
```

Training the model using SVM

In [133]:
```
from sklearn import svm
```

In [134]:
```
clf=svm.SVC(kernel='linear',C=1.0)
```

In [135]:
```
clf.fit(X_train,y_train)
```

Out[135]:
```
SVC(kernel='linear')
```

In [136]:
```
prediction=clf.predict(X_test)
```

In [137]:
```
from sklearn.metrics import classification_report,confusion_matrix
```

In [139]:
```
print(confusion_matrix(y_test,prediction))
```

```
[[ 39  22]
 [  1 184]]
```

In [140]:
```
print(classification_report(y_test,prediction))
```

```
              precision    recall  f1-score   support

           0       0.97      0.64      0.77        61
           1       0.89      0.99      0.94       185

    accuracy                           0.91       246
   macro avg       0.93      0.82      0.86       246
weighted avg       0.91      0.91      0.90       246
```

In [ ]: