In [ ]:
```
#225229140 pml lab 7
```

In [57]:
```
import pandas as pd
```

In [58]:
```
#step 1
```

In [59]:
```
df=pd.read_csv('train_loan.csv')
df
```

20	LP001043	Male	Yes	0	Not Graduate	No	7660
21	LP001046	Male	Yes	1	Graduate	No	5955
22	LP001047	Male	Yes	0	Not Graduate	No	2600
23	LP001050	NaN	Yes	2	Not Graduate	No	3365
24	LP001052	Male	Yes	1	Graduate	NaN	3717
25	LP001066	Male	Yes	0	Graduate	Yes	9560
26	LP001068	Male	Yes	0	Graduate	No	2799
27	LP001073	Male	Yes	2	Not Graduate	No	4226

In [60]:
```
df.shape
```

Out[60]:
```
(614, 13)
```

In [61]:
```
df.head
```

9	LP001020	Male	Yes	1	Graduate	No	
10	LP001024	Male	Yes	2	Graduate	No	
11	LP001027	Male	Yes	2	Graduate	NaN	
12	LP001028	Male	Yes	2	Graduate	No	
13	LP001029	Male	No	0	Graduate	No	
14	LP001030	Male	Yes	2	Graduate	No	
15	LP001032	Male	No	0	Graduate	No	
16	LP001034	Male	No	1	Not Graduate	No	
17	LP001036	Female	No	0	Graduate	No	
18	LP001038	Male	Yes	0	Not Graduate	No	
19	LP001041	Male	Yes	0	Graduate	NaN	
20	LP001043	Male	Yes	0	Not Graduate	No	
21	LP001046	Male	Yes	1	Graduate	No	
22	LP001047	Male	Yes	0	Not Graduate	No	
23	LP001050	NaN	Yes	2	Not Graduate	No	
24	LP001052	Male	Yes	1	Graduate	NaN	
25	LP001066	Male	Yes	0	Graduate	Yes	
26	LP001068	Male	Yes	0	Graduate	No	
27	LP001073	Male	Yes	2	Not Graduate	No	
28	LP001086	Male	No	0	Not Graduate	No	

In [62]:
```
df.columns
```

Out[62]:
```
Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
       'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
       'Loan_Amount_Term', 'Credit_History', 'Property_Area', 'Loan_Status'],
      dtype='object')
```

In [63]:
```
df.dtypes
```

Out[63]:
```
Loan_ID               object
Gender                object
Married               object
Dependents            object
Education             object
Self_Employed         object
ApplicantIncome        int64
CoapplicantIncome    float64
LoanAmount           float64
Loan_Amount_Term     float64
Credit_History       float64
Property_Area         object
Loan_Status           object
dtype: object
```

In [64]:
```
df.info
```

3	2583	2358.0	120.0	360.0
4	6000	0.0	141.0	360.0
5	5417	4196.0	267.0	360.0
6	2333	1516.0	95.0	360.0
7	3036	2504.0	158.0	360.0
8	4006	1526.0	168.0	360.0
9	12841	10968.0	349.0	360.0
10	3200	700.0	70.0	360.0
11	2500	1840.0	109.0	360.0
12	3073	8106.0	200.0	360.0
13	1853	2840.0	114.0	360.0
14	1299	1086.0	17.0	120.0
15	4950	0.0	125.0	360.0
16	3596	0.0	100.0	240.0
17	3510	0.0	76.0	360.0
18	4887	0.0	133.0	360.0
19	2600	3500.0	115.0	NaN
20	7660	0.0	104.0	360.0
21	5955	5625.0	315.0	360.0

In [65]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
Loan_ID              614 non-null object
Gender               601 non-null object
Married              611 non-null object
Dependents           599 non-null object
Education            614 non-null object
Self_Employed        582 non-null object
ApplicantIncome      614 non-null int64
CoapplicantIncome    614 non-null float64
LoanAmount           592 non-null float64
Loan_Amount_Term     600 non-null float64
Credit_History       564 non-null float64
Property_Area        614 non-null object
Loan_Status          614 non-null object
dtypes: float64(4), int64(1), object(8)
memory usage: 62.4+ KB
```

In [66]:
```
df.Credit_History.value_counts
```

Out[66]:
```
<bound method IndexOpsMixin.value_counts of 0      1.0
1      1.0
2      1.0
3      1.0
4      1.0
5      1.0
6      1.0
7      0.0
8      1.0
9      1.0
10     1.0
11     1.0
12     1.0
13     1.0
14     1.0
15     1.0
16     NaN
17     0.0
18     1.0
19     1.0
20     0.0
21     1.0
22     0.0
23     0.0
24     NaN
25     1.0
26     1.0
27     1.0
28     1.0
29     1.0
      ...
584    0.0
585    1.0
586    1.0
587    1.0
588    1.0
589    0.0
590    1.0
591    1.0
592    1.0
593    1.0
594    1.0
595    1.0
596    1.0
597    0.0
598    1.0
599    1.0
600    NaN
601    1.0
602    1.0
603    1.0
604    1.0
605    1.0
606    1.0
607    1.0
```

```
608    1.0
609    1.0
610    1.0
611    1.0
612    1.0
613    0.0
Name: Credit_History, Length: 614, dtype: float64>
```

In [67]:
```
#step 2
```

In [68]:
```
df['Dependents'].dtypes
```

Out[68]:
```
dtype('O')
```

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

Out[69]:
```
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 [70]:
```
df["Dependents"].fillna("No_Dep",inplace=True)
df["Dependents"]
```

```
595         0
596         2
597    No_Dep
598         0
599         2
600        3+
601         0
602        3+
603         0
604         1
605         0
606         1
607         2
608         0
609         0
610        3+
611         1
612         2
613         0
Name: Dependents, Length: 614, dtype: object
```

In [71]:
```
dep={'0':0,'1':1,'2':2,'3+':3,'No_Dep':0}
df.Dependents=[dep[item] for item in df.Dependents]
```

In [72]:
```
df['Dependents'].astype(int)
```

Out[72]:

```
0      0
1      1
2      0
3      0
4      0
5      2
6      0
7      3
8      2
9      1
10     2
11     2
12     2
13     0
14     2
15     0
16     1
17     0
18     0
19     0
20     0
21     1
22     0
23     2
24     1
25     0
26     0
27     2
28     0
29     2
      ..
584    1
585    1
586    0
587    0
588    0
589    2
590    0
591    2
592    3
593    0
594    0
595    0
596    2
597    0
598    0
599    2
600    3
601    0
602    3
603    0
604    1
605    0
606    1
607    2
```

```
608    0
609    0
610    3
611    1
612    2
613    0
Name: Dependents, Length: 614, dtype: int32
```

In [78]:
```
df['Gender'].fillna(df['Gender'].mode()[0],inplace=True)
```

In [80]:
```
df['Married'].fillna(df['Married'].mode()[0],inplace=True)
df['Education'].fillna(df['Education'].mode()[0],inplace=True)
df['Self_Employed'].fillna(df['Self_Employed'].mode()[0],inplace=True)
df['Credit_History'].fillna(df['Credit_History'].mode()[0],inplace=True)
```

In [83]:
```
df.isnull().sum()
```

Out[83]:
```
Loan_ID               0
Gender                0
Married               0
Dependents            0
Education             0
Self_Employed         0
ApplicantIncome       0
CoapplicantIncome     0
LoanAmount           22
Loan_Amount_Term     14
Credit_History        0
Property_Area         0
Loan_Status           0
dtype: int64
```

In [85]:
```
df['LoanAmount'].fillna(df['LoanAmount'].mean(),inplace=True)
df['Loan_Amount_Term'].fillna(df['Loan_Amount_Term'].mean(),inplace=True)
```

In [86]:
```
df.drop(["Loan_ID"],axis=1)
```

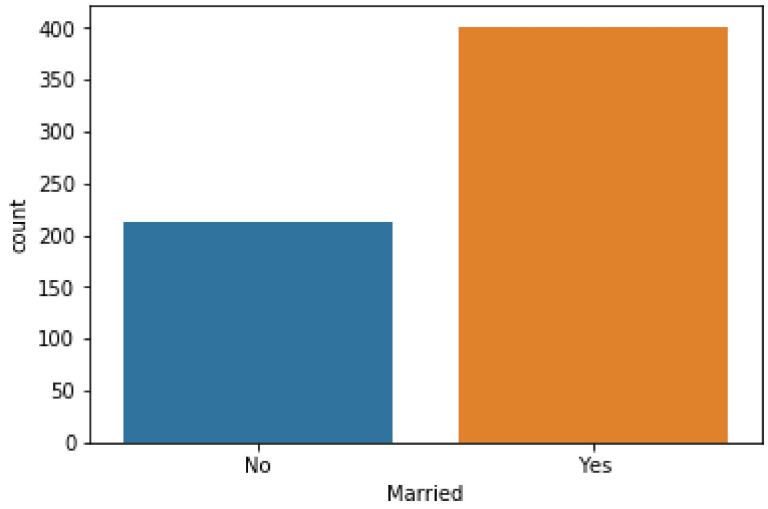
603	Male	No	0	Graduate	No	3676	4301
604	Female	Yes	1	Graduate	No	12000	0
605	Male	Yes	0	Not Graduate	No	2400	3800
606	Male	Yes	1	Graduate	No	3400	2500
607	Male	Yes	2	Not Graduate	No	3987	1411
608	Male	Yes	0	Graduate	No	3232	1950
609	Female	No	0	Graduate	No	2900	0
610	Male	Yes	3	Graduate	No	4106	0
611	Male	Yes	1	Graduate	No	8072	240
612	Male	Yes	2	Graduate	No	7583	0
613	Female	No	0	Graduate	Yes	4583	0

step3

In [90]:
```
import seaborn as sns
import matplotlib.pyplot as plt
```

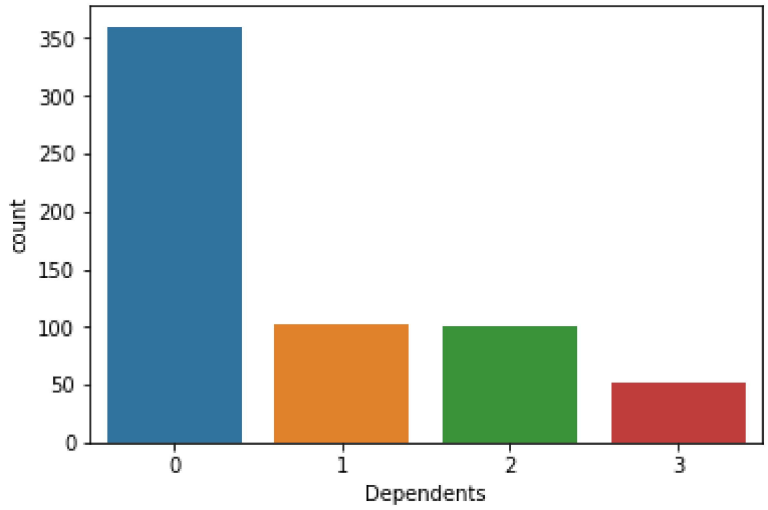
In [101]:
```
sns.countplot(x='Married',data=df)
```

Out[101]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x24647fde160>
```



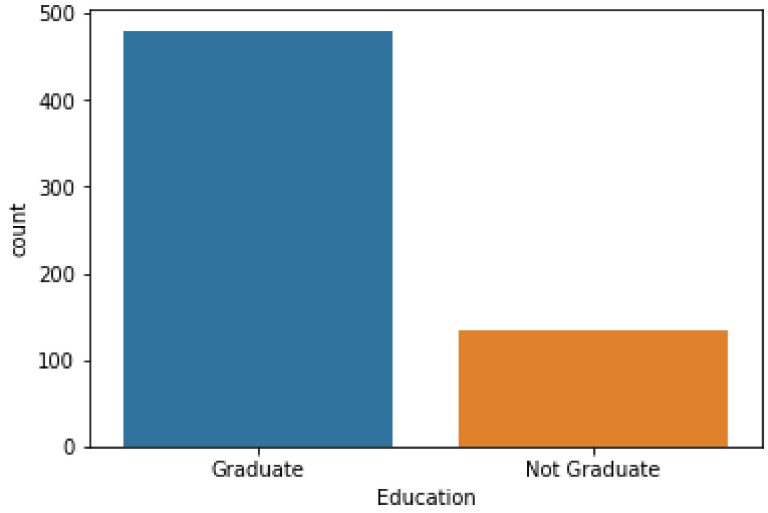
In [102]:
```
sns.countplot(x='Dependents',data=df)
```

Out[102]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x2464ab73550>
```



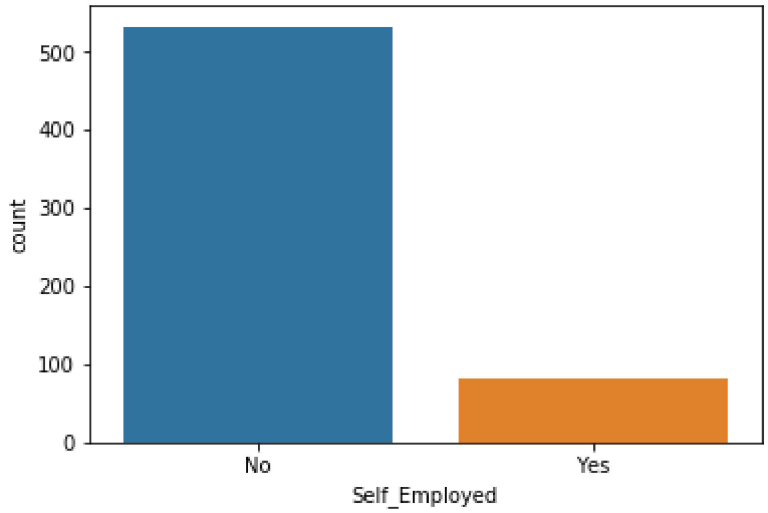
In [105]:
```
sns.countplot(x='Education',data=df)
```

Out[105]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x24647c42d30>
```



In [106]:
```
sns.countplot(x='Self_Employed',data=df)
```

Out[106]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x2464b0fe588>
```



step 4

In [108]:
```
x=df.drop(['Loan_Status'],axis=1)
```

In [110]:
```
y=df.pop('Loan_Status')
```

step 5

In [128]:
```
x=pd.get_dummies(x)
x
```

Out[128]:

	Dependents	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_Hi
0	0	5849	0.0	146.412162	360.0	
1	1	4583	1508.0	128.000000	360.0	
2	0	3000	0.0	66.000000	360.0	
3	0	2583	2358.0	120.000000	360.0	
4	0	6000	0.0	141.000000	360.0	
5	2	5417	4196.0	267.000000	360.0	
6	0	2333	1516.0	95.000000	360.0	
7	3	3036	2504.0	158.000000	360.0	
8	2	4006	1526.0	168.000000	360.0	
9	1	12841	10968.0	349.000000	360.0	
10	2	3200	700.0	70.000000	360.0	
11	2	2500	1840.0	109.000000	360.0	
12	2	3073	8106.0	200.000000	360.0	
13	0	1853	2840.0	114.000000	360.0	
14	2	1299	1086.0	17.000000	120.0	
15	0	4950	0.0	125.000000	360.0	
16	1	3596	0.0	100.000000	240.0	
17	0	3510	0.0	76.000000	360.0	
18	0	4887	0.0	133.000000	360.0	
19	0	2600	3500.0	115.000000	342.0	
20	0	7660	0.0	104.000000	360.0	
21	1	5955	5625.0	315.000000	360.0	
22	0	2600	1911.0	116.000000	360.0	
23	2	3365	1917.0	112.000000	360.0	
24	1	3717	2925.0	151.000000	360.0	
25	0	9560	0.0	191.000000	360.0	
26	0	2799	2253.0	122.000000	360.0	
27	2	4226	1040.0	110.000000	360.0	
28	0	1442	0.0	35.000000	360.0	
29	2	3750	2083.0	120.000000	360.0	
...	...	...	...	...	...	
584	1	2787	1917.0	146.000000	360.0	
585	1	4283	3000.0	172.000000	84.0	

	Dependents	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_Hi
586	0	2297	1522.0	104.000000	360.0	
587	0	2165	0.0	70.000000	360.0	
588	0	4750	0.0	94.000000	360.0	
589	2	2726	0.0	106.000000	360.0	
590	0	3000	3416.0	56.000000	180.0	
591	2	6000	0.0	205.000000	240.0	
592	3	9357	0.0	292.000000	360.0	
593	0	3859	3300.0	142.000000	180.0	
594	0	16120	0.0	260.000000	360.0	
595	0	3833	0.0	110.000000	360.0	
596	2	6383	1000.0	187.000000	360.0	
597	0	2987	0.0	88.000000	360.0	
598	0	9963	0.0	180.000000	360.0	
599	2	5780	0.0	192.000000	360.0	
600	3	416	41667.0	350.000000	180.0	
601	0	2894	2792.0	155.000000	360.0	
602	3	5703	0.0	128.000000	360.0	
603	0	3676	4301.0	172.000000	360.0	
604	1	12000	0.0	496.000000	360.0	
605	0	2400	3800.0	146.412162	180.0	
606	1	3400	2500.0	173.000000	360.0	
607	2	3987	1411.0	157.000000	360.0	
608	0	3232	1950.0	108.000000	360.0	
609	0	2900	0.0	71.000000	360.0	
610	3	4106	0.0	40.000000	180.0	
611	1	8072	240.0	253.000000	360.0	
612	2	7583	0.0	187.000000	360.0	
613	0	4583	0.0	133.000000	360.0	

614 rows × 631 columns

step 6

In [147]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.30,random_state=42)
```

In [148]:
```
from sklearn.preprocessing import StandardScaler
ss=StandardScaler()
```

In [149]:
```
x_train_ss=ss.fit_transform(x_train)
x_train_ss
```

Out[149]:
```
array([[-0.71703534, -0.50133384,  0.27865737, ..., -0.62317695,
        -0.79056942,  1.40682858],
       [-0.71703534, -0.42803179,  0.45103751, ...,  1.60468065,
        -0.79056942, -0.71081865],
       [-0.71703534, -0.5669725 ,  0.23208844, ..., -0.62317695,
         1.26491106, -0.71081865],
       ...,
       [-0.71703534, -0.37088951, -0.59751445, ..., -0.62317695,
        -0.79056942,  1.40682858],
       [-0.71703534,  0.76362634, -0.59751445, ..., -0.62317695,
         1.26491106, -0.71081865],
       [-0.71703534,  1.36387019, -0.59751445, ..., -0.62317695,
        -0.79056942,  1.40682858]])
```

In [156]:
```
x_test_ss=ss.fit_transform(x_test)
x_test_ss
```

Out[156]:
```
array([[-0.78697069,  0.60310661, -0.4897835 , ..., -0.68429085,
         1.31171195, -0.67579058],
       [-0.78697069, -0.1508012 , -0.4897835 , ..., -0.68429085,
         1.31171195, -0.67579058],
       [ 1.15422368, -0.17338842, -0.07075971, ...,  1.4613669 ,
        -0.7623625 , -0.67579058],
       ...,
       [ 1.15422368,  1.02547189, -0.4897835 , ..., -0.68429085,
        -0.7623625 ,  1.47974835],
       [-0.78697069, -0.34587267,  0.20984434, ...,  1.4613669 ,
        -0.7623625 , -0.67579058],
       [-0.78697069,  0.03716241, -0.4897835 , ...,  1.4613669 ,
        -0.7623625 , -0.67579058]])
```

In [157]:
```
from sklearn.svm import LinearSVC
lvc=LinearSVC()
lvc.fit(x_train_ss,y_train)
l_pred=lvc.predict(x_test_ss)
```

In [158]:
```
l_pred
```

Out[158]:
```
array(['Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y',
       'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y',
       'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'N', 'Y', 'Y', 'Y',
       'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'N', 'N', 'N', 'Y', 'Y',
       'N', 'Y', 'Y', 'Y', 'Y', 'Y', 'N', 'Y', 'N', 'Y', 'N', 'Y', 'Y',
       'Y', 'Y', 'Y', 'Y', 'Y', 'N', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'N',
       'N', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y',
       'N', 'N', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y',
       'Y', 'N', 'Y', 'N', 'Y', 'Y', 'Y', 'N', 'Y', 'Y', 'Y', 'Y', 'Y',
       'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'N', 'Y', 'Y',
       'N', 'Y', 'Y', 'N', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y',
       'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y',
       'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y',
       'Y', 'N', 'N', 'Y', 'N', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y',
       'Y', 'Y', 'N'], dtype=object)
```

In [159]:
```
from sklearn.metrics import accuracy_score
lvc_acc=accuracy_score(y_test,l_pred)
print(lvc_acc)
```

```
0.7513513513513513
```

In [160]:
```
from sklearn.metrics import confusion_matrix
c_mat=confusion_matrix(y_test,l_pred)
c_mat
```

Out[160]:
```
array([[ 21,  44],
       [  2, 118]], dtype=int64)
```

In [162]:
```
from sklearn.metrics import classification_report
c_rep=classification_report(y_test,l_pred)
print(c_rep)
```

```
             precision    recall  f1-score   support

          N       0.91      0.32      0.48        65
          Y       0.73      0.98      0.84       120

avg / total       0.79      0.75      0.71       185
```

step 7

In [168]:
```
from  sklearn.linear_model import LogisticRegression
```

In [175]:
```
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression()
lr.fit(x_train_ss,y_train)
lr_pred=lr.predict(x_test_ss)

from sklearn.svm import LinearSVC
lvc=LinearSVC()
lvc.fit(x_train_ss,y_train)
l_pred=lvc.predict(x_test_ss)

from sklearn.metrics import accuracy_score
lvc_acc=accuracy_score(y_test,l_pred)
print("linear_svc_acc_score : ",lvc_acc)

from sklearn.metrics import accuracy_score
lr_acc=accuracy_score(y_test,lr_pred)
print("linear_reg_acc_score : ",lr_acc)
```

```
linear_svc_acc_score :  0.7513513513513513
linear_reg_acc_score :  0.7783783783783784
```

In [ ]: