In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn.metrics import accuracy_score
from matplotlib import pyplot as plt
```

In [2]:
```
df=pd.read_csv("C:\\Users\\Pranav\\Desktop\\DATA SCIENCE DATA\\CVZ file\\train_u6lujuX_CVtuZ9i (1).csv")
```

In [3]:
```
df.head()
```

Out[3]:

	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 [4]:
```
#number of rows and columns
df.shape
```

Out[4]:
```
(614, 13)
```

In [5]:
```
#describe mathamatical data
df.describe()
```

Out[5]:

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	614.000000	614.000000	592.000000	600.00000	564.000000
mean	5403.459283	1621.245798	146.412162	342.00000	0.842199
std	6109.041673	2926.248369	85.587325	65.12041	0.364878
min	150.000000	0.000000	9.000000	12.00000	0.000000
25%	2877.500000	0.000000	100.000000	360.00000	1.000000
50%	3812.500000	1188.500000	128.000000	360.00000	1.000000
75%	5795.000000	2297.250000	168.000000	360.00000	1.000000
max	81000.000000	41667.000000	700.000000	480.00000	1.000000

In [6]:
```
#number of missing value
df.isnull().sum()
```

Out[6]:
```
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 [7]:
```
#droping the missing values
loan_dataset=df.dropna()
```

In [8]:
```
loan_dataset.isnull().sum()
```

Out[8]:
```
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 [9]:
```
#label encoding
loan_dataset.replace({'Loan_Status':{'N':0,'Y':1}},inplace=True)
```

```
C:\Users\Pranav\AppData\Local\Temp\ipykernel_9956\4297967148.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  loan_dataset.replace({'Loan_Status':{'N':0,'Y':1}},inplace=True)
```

In [10]:
```
loan_dataset.head()
```

Out[10]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	360.0	1.0	Rural	0
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	360.0	1.0	Urban	1
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	360.0	1.0	Urban	1
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.0	360.0	1.0	Urban	1
5	LP001011	Male	Yes	2	Graduate	Yes	5417	4196.0	267.0	360.0	1.0	Urban	1

In [11]:
```
loan_dataset['Dependents'].value_counts()
```

Out[11]:
```
0     274
2      85
1      80
3+     41
Name: Dependents, dtype: int64
```

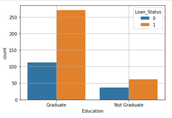
In [12]:
```
loan_dataset= loan_dataset.replace(to_replace='3+',value=4)
print(loan_dataset)
```

```
     Loan_ID  Gender Married Dependents     Education Self_Employed  \
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
5   LP001011    Male     Yes          2      Graduate           Yes
..       ...     ...     ...        ...           ...           ...
609 LP002978  Female      No          0      Graduate            No
610 LP002979    Male     Yes          4      Graduate            No
611 LP002983    Male     Yes          1      Graduate            No
612 LP002984    Male     Yes          2      Graduate            No
613 LP002990  Female      No          0      Graduate           Yes

     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
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
5               5417             4196.0       267.0             360.0
..               ...                ...         ...               ...
609             2900                0.0        71.0             360.0
610             4106                0.0        40.0             180.0
611             8072             240.0       253.0             360.0
612             7583                0.0       187.0             360.0
613             4583                0.0       133.0             360.0

     Credit_History Property_Area  Loan_Status
1               1.0         Rural            0
2               1.0         Urban            1
3               1.0         Urban            1
4               1.0         Urban            1
5               1.0         Urban            1
..              ...           ...          ...
609             1.0         Rural            1
610             1.0         Rural            1
611             1.0         Urban            1
612             1.0         Urban            1
613             0.0     Semiurban            0

[480 rows x 13 columns]
```

In [13]:
```
loan_dataset['Dependents'].value_counts()
```

Out[13]:
```
0    274
2     85
1     80
4     41
Name: Dependents, dtype: int64
```

In [14]:
```
#data visualization
sns.countplot(x='Education',hue='Loan_Status',data= loan_dataset)
plt.grid(True)
plt.show()
```



In [15]:
```
#0-->loan is not aprove    1--> loan is aprove
```

In [16]:
```
loan_dataset.groupby('Loan_Status').mean()
```

Out[16]:

Loan_Status	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
0	5730.189189	1773.081081	153.378378	342.810811	0.574324
1	5201.093373	1495.508795	140.882530	341.710843	0.978916

In [17]:
```
#data visualization
sns.countplot(x='Married',hue='Loan_Status',data= loan_dataset)
plt.grid(True)
plt.show()
```

In [32]:
```
loan_dataset.replace({'Gender':{'Male':1,'Female':0},'Married':{'Yes':1,'No':0},'Education':{'Graduate':1,'Not Graduate':0},'Self_Employed':{'Yes':1,'No':0},'Property_Area':{'Urban
```

In [33]:
```
loan_dataset
```

Out[33]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
1	LP001003	1	1	1	1	0	4583	1508.0	128.0	360.0	1.0	0	0
2	LP001005	1	1	0	1	1	3000	0.0	66.0	360.0	1.0	1	1
3	LP001006	1	1	0	0	0	2583	2358.0	120.0	360.0	1.0	1	1
4	LP001008	1	0	0	1	0	6000	0.0	141.0	360.0	1.0	1	1
5	LP001011	1	1	2	1	1	5417	4196.0	267.0	360.0	1.0	1	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...
609	LP002978	0	0	0	1	0	2900	0.0	71.0	360.0	1.0	0	1
610	LP002979	1	1	4	1	0	4106	0.0	40.0	180.0	1.0	0	1
611	LP002983	1	1	1	1	0	8072	240.0	253.0	360.0	1.0	1	1
612	LP002984	1	1	2	1	0	7583	0.0	187.0	360.0	1.0	1	1
613	LP002990	0	0	0	1	1	4583	0.0	133.0	360.0	0.0	2	0

480 rows × 13 columns

In [34]:
```
loan_dataset['Dependents'].value_counts()
```

Out[34]:
```
0    274
2     85
1     80
4     41
Name: Dependents, dtype: int64
```

In [35]:
```
#seperting data and labels
X=loan_dataset.drop(columns=['Loan_Status','Loan_ID'],axis=1)
y=loan_dataset['Loan_Status']
print(X)
print(y)
```

```
     Gender  Married Dependents  Education  Self_Employed  ApplicantIncome  \
1         1        1          1          1              0             4583
2         1        1          0          1              1             3000
3         1        1          0          0              0             2583
4         1        0          0          1              0             6000
5         1        1          2          1              1             5417
..      ...      ...        ...        ...            ...              ...
609       0        0          0          1              0             2900
610       1        1          4          1              0             4106
611       1        1          1          1              0             8072
612       1        1          2          1              0             7583
613       0        0          0          1              1             4583

     CoapplicantIncome  LoanAmount  Loan_Amount_Term  Credit_History  \
1               1508.0       128.0             360.0             1.0
2                  0.0        66.0             360.0             1.0
3               2358.0       120.0             360.0             1.0
4                  0.0       141.0             360.0             1.0
5               4196.0       267.0             360.0             1.0
..                 ...         ...               ...             ...
609                0.0        71.0             360.0             1.0
610                0.0        40.0             180.0             1.0
611              240.0       253.0             360.0             1.0
612                0.0       187.0             360.0             1.0
613                0.0       133.0             360.0             0.0

     Property_Area
1                0
2                1
3                1
4                1
5                1
..             ...
609              0
610              0
611              1
612              1
613              2

[480 rows x 11 columns]
1      0
2      1
3      1
4      1
5      1
      ..
609    1
610    1
611    1
612    1
613    0
Name: Loan_Status, Length: 480, dtype: int64
```

In [36]:
```
#training and test data
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.1,random_state=2,stratify=y)
print("shape of X_train= ",X_train.shape)
print("shape of X_test= ",X_test.shape)
print("shape of y_train= ",y_train.shape)
print("shape of y_test= ",y_test.shape)
```

```
shape of X_train=  (432, 11)
shape of X_test=  (48, 11)
shape of y_train=  (432,)
shape of y_test=  (48,)
```

In [37]:
```
#Training model
classifier=svm.SVC(kernel='linear')
classifier
```

Out[37]:
```
SVC(kernel='linear')
```

In [38]:
```
# trainin the support vector Machine classifier
classifier.fit(X_train,y_train)
```

Out[38]:
```
SVC(kernel='linear')
```

In [39]:
```
#model evaluation
#Accuracy score
#accuracy of training data
X_train_prediction=classifier.predict(X_train)
training_data_accuracy=accuracy_score(X_train_prediction,y_train)
print('Accuracy on training data:',training_data_accuracy)
```

```
Accuracy on training data: 0.7708333333333334
```

In [40]:
```
#accuracy of testing data
X_test_prediction=classifier.predict(X_test)
testing_data_accuracy=accuracy_score(X_test_prediction,y_test)
print('Accuracy on test data:',testing_data_accuracy)
```

```
Accuracy on test data: 0.8125
```