

```
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
```

```
loan_dataset = pd.read_csv("loan.csv")
```

```
loan_dataset.head()
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Am
0	LP001002	Male	No	0	Graduate	No	5849	0.0	NaN	
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.0	

```
loan_dataset.isnull().sum()
```

```
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
```

```
loan_dataset["Loan_Status"].describe()
```

```
count      614
unique       2
top         Y
freq       422
Name: Loan_Status, dtype: object
```

```
loan_dataset["Loan_Status"] = loan_dataset["Loan_Status"].map({"Y":1, "N":0})
```

```
loan_dataset.shape
```

```
(614, 13)
```

```
loan_dataset.head()
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Am
0	LP001002	Male	No	0	Graduate	No	5849	0.0	NaN	
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.0	

```
loan_dataset = loan_dataset.dropna()
```

```
loan_dataset.isnull().sum()
```

```
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
```

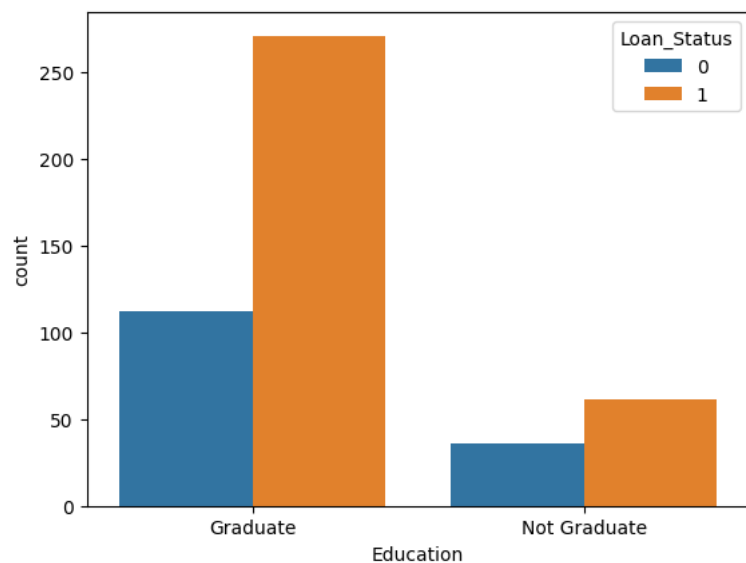
```
loan_dataset = loan_dataset.replace(to_replace = '3+', value=4)
```

```
loan_dataset["Dependents"].value_counts()
```

```
0    274
2     85
1     80
4     41
Name: Dependents, dtype: int64
```

```
sns.countplot(x='Education', hue="Loan_Status", data = loan_dataset)
```

```
<Axes: xlabel='Education', ylabel='count'>
```



```
sns.countplot(x="Married", hue="Loan_Status", data = loan_dataset)
```

```
<Axes: xlabel='Married', ylabel='count'>

loan_dataset = loan_dataset.drop("Loan_ID", axis =1)
loan_dataset = pd.get_dummies(loan_dataset, drop_first = True)

loan_dataset.head()
```

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Loan_Status	Gender_Male	Married_Yes
1	4583	1508.0	128.0	360.0	1.0	0	1	1
2	3000	0.0	66.0	360.0	1.0	1	1	1
3	2583	2358.0	120.0	360.0	1.0	1	1	1
4	6000	0.0	141.0	360.0	1.0	1	1	0
5	5417	4196.0	267.0	360.0	1.0	1	1	1

```
loan_dataset.shape

(480, 15)
```

```
X = loan_dataset.drop("Loan_Status", axis =1).values
y = loan_dataset["Loan_Status"].values

X_train, X_test, y_train,y_test = train_test_split(X,y, stratify = y, test_size = 0.25)
```

```
classifier = svm.SVC(kernel = 'linear')
```

```
classifier.fit(X_train, y_train)
```

▼ SVC

SVC(kernel='linear')

```
y_pred = classifier.predict(X_test)
```

```
from sklearn.metrics import classification_report
```

```
print(classification_report(y_pred,y_test))
```

	precision	recall	f1-score	support
0	0.43	0.89	0.58	18
1	0.98	0.79	0.88	102
accuracy			0.81	120
macro avg	0.70	0.84	0.73	120
weighted avg	0.89	0.81	0.83	120

✓ 0s completed at 10:30 PM

● ×