```
import numpy as np
import pandas as pd
import seaborn as sns
# first upload file
df=pd.read_csv("/content/LoanApprovalPrediction.csv")
df
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncor
0	LP001002	Male	No	0	Graduate	No	5849	0
1	LP001003	Male	Yes	1	Graduate	No	4583	1508
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358
4	LP001008	Male	No	0	Graduate	No	6000	0
...	...	...	...	...	...	...	...	
609	LP002978	Female	No	0	Graduate	No	2900	0
610	LP002979	Male	Yes	3+	Graduate	No	4106	0
611	LP002983	Male	Yes	1	Graduate	No	8072	240
612	LP002984	Male	Yes	2	Graduate	No	7583	0
613	LP002990	Female	No	0	Graduate	Yes	4583	0

614 rows × 13 columns

```
df.head()
```

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

```
df.tail()
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncor
609	LP002978	Female	No	0	Graduate	No	2900	0
610	LP002979	Male	Yes	3+	Graduate	No	4106	0
611	LP002983	Male	Yes	1	Graduate	No	8072	240
612	LP002984	Male	Yes	2	Graduate	No	7583	0
613	LP002990	Female	No	0	Graduate	Yes	4583	0

```
df.shape
```

```
(614, 13)
```

```
df.isna().sum
```

```
<bound method NDFrame._add_numeric_operations.<locals>.sum of      Loan_ID  Gender  Married  Dependents  Education  Self_Employed  \
0      False   False    False       False      False          False
1      False   False    False       False      False          False
2      False   False    False       False      False          False
3      False   False    False       False      False          False
4      False   False    False       False      False          False
..       ...     ...      ...         ...        ...            ...
609    False   False    False       False      False          False
610    False   False    False       False      False          False
611    False   False    False       False      False          False
612    False   False    False       False      False          False
613    False   False    False       False      False          False

     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
0              False              False        True             False
1              False              False       False             False
2              False              False       False             False
3              False              False       False             False
4              False              False       False             False
..               ...                ...         ...               ...
609            False              False       False             False
610            False              False       False             False
611            False              False       False             False
612            False              False       False             False
613            False              False       False             False

     Credit_History  Property_Area  Loan_Status
0             False          False        False
1             False          False        False
2             False          False        False
3             False          False        False
4             False          False        False
..              ...            ...          ...
609           False          False        False
610           False          False        False
611           False          False        False
612           False          False        False
613           False          False        False

[614 rows x 13 columns]>
```

```
df.columns
```

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

```
df.describe()
```

	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

```
df1=df.drop(['Loan_ID','Gender','Married','Dependents','Education','Self_Employed','Property_Area'],axis=1)
df1
```

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Loan_Status
0	5849	0.0	NaN	360.0	1.0	Y
1	4583	1508.0	128.0	360.0	1.0	N
2	3000	0.0	66.0	360.0	1.0	Y
3	2583	2358.0	120.0	360.0	1.0	Y
4	6000	0.0	141.0	360.0	1.0	Y
...	...	...	...	...	...	...

```
df1.isna().sum()
```

```
ApplicantIncome       0
CoapplicantIncome     0
LoanAmount           22
Loan_Amount_Term     14
Credit_History       50
Loan_Status           0
dtype: int64
```

```
from numpy.core.fromnumeric import mean
df1['LoanAmount']=df1['LoanAmount'].fillna(df1['LoanAmount'].mean())
df1['Loan_Amount_Term']=df1['Loan_Amount_Term'].fillna(df1['Loan_Amount_Term'].mode()[0])
df1['Credit_History']=df1['Credit_History'].fillna(df1['Credit_History'].mode()[0])
```

```
df1.isna().sum()
```

```
ApplicantIncome      0
CoapplicantIncome    0
LoanAmount           0
Loan_Amount_Term     0
Credit_History       0
Loan_Status          0
dtype: int64
```

```
x=df1.drop(['Loan_Status'],axis=True).values
x
```

```
array([[5.84900000e+03, 0.00000000e+00, 1.46412162e+02, 3.60000000e+02,
        1.00000000e+00],
       [4.58300000e+03, 1.50800000e+03, 1.28000000e+02, 3.60000000e+02,
        1.00000000e+00],
       [3.00000000e+03, 0.00000000e+00, 6.60000000e+01, 3.60000000e+02,
        1.00000000e+00],
       ...,
       [8.07200000e+03, 2.40000000e+02, 2.53000000e+02, 3.60000000e+02,
        1.00000000e+00],
       [7.58300000e+03, 0.00000000e+00, 1.87000000e+02, 3.60000000e+02,
        1.00000000e+00],
       [4.58300000e+03, 0.00000000e+00, 1.33000000e+02, 3.60000000e+02,
        0.00000000e+00]])
```

```
y=df1['Loan_Status'].values
y
```

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

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,random_state=42)
x_train
```

```
array([[2.45400000e+03, 2.33300000e+03, 1.81000000e+02, 3.60000000e+02,
        0.00000000e+00],
       [2.89400000e+03, 2.79200000e+03, 1.55000000e+02, 3.60000000e+02,
        1.00000000e+00],
       [2.06000000e+03, 2.20900000e+03, 1.34000000e+02, 3.60000000e+02,
        1.00000000e+00],
       ...,
       [3.23700000e+03, 0.00000000e+00, 3.00000000e+01, 3.60000000e+02,
        1.00000000e+00],
       [1.00470000e+04, 0.00000000e+00, 1.46412162e+02, 2.40000000e+02,
        1.00000000e+00],
       [1.36500000e+04, 0.00000000e+00, 1.46412162e+02, 3.60000000e+02,
        1.00000000e+00]])
```

```
x_test
```

```
        1.00000000e+00],
       [2.60000000e+03, 1.91100000e+03, 1.16000000e+02, 3.60000000e+02,
        0.00000000e+00],
       [4.30100000e+03, 0.00000000e+00, 1.18000000e+02, 3.60000000e+02,
        1.00000000e+00],
       [3.16700000e+03, 2.28300000e+03, 1.54000000e+02, 3.60000000e+02,
        1.00000000e+00],
       [5.69500000e+03, 4.16700000e+03, 1.75000000e+02, 3.60000000e+02,
        1.00000000e+00],
       [1.88000000e+03, 0.00000000e+00, 6.10000000e+01, 3.60000000e+02,
        1.00000000e+00],
       [3.77190000e+04, 0.00000000e+00, 1.52000000e+02, 3.60000000e+02,
        1.00000000e+00],
       [4.88700000e+03, 0.00000000e+00, 1.33000000e+02, 3.60000000e+02,
        1.00000000e+00],
       [3.33300000e+03, 2.50000000e+03, 1.28000000e+02, 3.60000000e+02,
        1.00000000e+00],
       [4.65200000e+03, 0.00000000e+00, 1.10000000e+02, 3.60000000e+02,
        1.00000000e+00],
       [1.17570000e+04, 0.00000000e+00, 1.87000000e+02, 1.80000000e+02,
        1.00000000e+00],
       [3.07500000e+03, 2.41600000e+03, 1.39000000e+02, 3.60000000e+02,
        1.00000000e+00],
       [5.50000000e+03, 0.00000000e+00, 1.05000000e+02, 3.60000000e+02,
        0.00000000e+00]])
```

```
y_train
```

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

```
y_test
```

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

```
from sklearn.preprocessing import StandardScaler
scalar=StandardScaler()
scalar.fit(x_train)
x_train=scalar.transform(x_train)
x_train
```

```
array([[-0.50133384,  0.27865737,  0.40368493,  0.30437507, -2.47991935],
       [-0.42803179,  0.45103751,  0.09632945,  0.30437507,  0.40323892],
       [-0.5669725 ,  0.23208844, -0.15191921,  0.30437507,  0.40323892],
       ...,
       [-0.37088951, -0.59751445, -1.38134113,  0.30437507,  0.40323892],
       [ 0.76362634, -0.59751445, -0.00519051, -1.45542149,  0.40323892],
       [ 1.36387019, -0.59751445, -0.00519051,  0.30437507,  0.40323892]])
```

```
x_test=scalar.transform(x_test)
x_test
```

```
       [-2.27950509e-01,  6.97026595e-01,  4.90439918e-02,
         3.04375070e-01,  4.03238919e-01],
       [ 9.22391921e-01, -5.97514448e-01, -7.54808805e-01,
         3.04375070e-01,  4.03238919e-01],
       [ 7.49632312e-01, -5.97514448e-01,  3.91863567e-01,
         3.04375070e-01,  4.03238919e-01],
       [-4.85340669e-01,  1.69370615e-01, -2.46490125e-01,
         3.04375070e-01,  4.03238919e-01],
       [-8.75174309e-01,  4.97981990e-01, -5.77488335e-01,
         3.04375070e-01,  4.03238919e-01],
       [-3.82717796e-01, -5.97514448e-01, -1.31041294e+00,
         3.04375070e-01,  4.03238919e-01],
       [-1.77138859e-01, -5.97514448e-01, -2.34668760e-01,
         3.04375070e-01, -2.47991935e+00],
       [-2.85425981e-01, -5.97514448e-01, -4.00167865e-01,
         2.06417164e+00,  4.03238919e-01],
       [-3.64892070e-01,  8.59972141e-02, -7.78451535e-01,
         3.04375070e-01,  4.03238919e-01],
       [-2.43777088e-01,  3.41375198e-01, -8.09910195e-02,
         3.04375070e-01,  4.03238919e-01],
       [ 2.72669188e-01, -5.97514448e-01, -2.58311489e-01,
        -4.09511634e+00,  4.03238919e-01],
       [-8.55112940e-02, -5.97514448e-01, -2.58311489e-01,
         3.04375070e-01,  4.03238919e-01],
       [-3.38736565e-01, -1.28069625e-01, -2.22847395e-01,
         3.04375070e-01, -2.47991935e+00],
       [-3.40735712e-01, -5.97514448e-01, -1.40097843e-01,
         3.04375070e-01,  4.03238919e-01],
       [ 5.16898297e-01, -5.97514448e-01,  7.46504507e-01,
         3.04375070e-01,  4.03238919e-01],
       [ 5.85938765e-02,  7.79273328e-01,  1.94046234e+00,
         3.04375070e-01,  4.03238919e-01],
       [-5.41316781e-01, -7.24873581e-02, -7.31166076e-01,
         3.04375070e-01,  4.03238919e-01],
       [-3.64392283e-01, -4.15745412e-01, -1.40097843e-01,
         3.04375070e-01,  4.03238919e-01],
       [-3.69390150e-01, -6.53517968e-02, -1.04633749e-01,
         3.04375070e-01,  4.03238919e-01],
       [-3.27074875e-01,  2.85371678e-02, -3.88346501e-01,
         3.04375070e-01,  4.03238919e-01],
       [ 6.42581259e-02, -5.97514448e-01, -5.19051226e-03,
         3.04375070e-01,  4.03238919e-01],
       [-2.45942831e-01, -6.76051319e-02,  1.19972180e-01,
         3.04375070e-01,  4.03238919e-01],
       [ 8.78180019e-03,  3.64626675e+00,  4.11559344e+00,
         3.04375070e-01, -2.47991935e+00],
       [ 4.78081529e-01,  5.91870955e-01,  2.14543097e-01,
         3.04375070e-01,  4.03238919e-01],
       [-6.60266020e-01,  7.84860969e-02, -5.18381512e-01,
         3.04375070e-01, -2.47991935e+00],
       [ 1.36560605e-01,  1.06094022e+00,  7.34683142e-01,
         3.04375070e-01, -2.47991935e+00],
       [-8.86766099e-02, -5.97514448e-01, -2.22847395e-01,
         3.04375070e-01,  4.03238919e-01],
       [-4.77010890e-01,  1.20172797e-01, -3.64703771e-01,
         3.04375070e-01, -2.47991935e+00],
       [-1.93631821e-01, -5.97514448e-01, -3.41061042e-01,
         3.04375070e-01,  4.03238919e-01],
       [-3.82551200e-01,  2.59879576e-01,  8.45080858e-02,
```

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
knn=KNeighborsClassifier(n_neighbors=7)
nb_model=GaussianNB()
sv_model=SVC()
```

```
lsb_model=[knn,nb_model,sv_model]
```

```
from sklearn.metrics import confusion_matrix,accuracy_score,classification_report
for i in lsb_model:
  print(i)
  i.fit(x_train,y_train)
  y_pred=i.predict(x_test)
  print("***********")
  print(classification_report(y_test,y_pred))
  print("**************")
  print(confusion_matrix(y_test,y_pred))
```

```
KNeighborsClassifier(n_neighbors=7)
***********
              precision    recall  f1-score   support

           N       0.93      0.40      0.56        65
           Y       0.75      0.98      0.85       120

    accuracy                           0.78       185
   macro avg       0.84      0.69      0.71       185
weighted avg       0.81      0.78      0.75       185

**************
[[ 26  39]
 [  2 118]]
GaussianNB()
***********
              precision    recall  f1-score   support

           N       0.88      0.45      0.59        65
           Y       0.76      0.97      0.85       120

    accuracy                           0.78       185
   macro avg       0.82      0.71      0.72       185
weighted avg       0.80      0.78      0.76       185

**************
[[ 29  36]
 [  4 116]]
SVC()
***********
              precision    recall  f1-score   support

           N       0.93      0.42      0.57        65
           Y       0.76      0.98      0.86       120

    accuracy                           0.78       185
   macro avg       0.84      0.70      0.71       185
weighted avg       0.82      0.78      0.76       185

**************
[[ 27  38]
 [  2 118]]
```

✓ 0s completed at 2:37 PM