```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df=pd.read_csv("C:\\Users\\Keerthi Priya\\OneDrive\\Desktop\\archive\\
loan_sanction_train.csv")

df
```

```
      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

..        ...     ...     ...        ...           ...            ...

609  LP002978  Female      No          0      Graduate             No

610  LP002979    Male     Yes         3+      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
\
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

..               ...                ...         ...               ...

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
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
..              ...           ...         ...
609             1.0         Rural           Y
610             1.0         Rural           Y
611             1.0         Urban           Y
612             1.0         Urban           Y
613             0.0     Semiurban           N

[614 rows x 13 columns]
```

```
df.head()
```

```
    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

   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

   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
```

```
df.shape
```

```
(614, 13)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Loan_ID            614 non-null    object 
 1   Gender             601 non-null    object 
 2   Married            611 non-null    object 
 3   Dependents         599 non-null    object 
 4   Education          614 non-null    object 
 5   Self_Employed      582 non-null    object 
 6   ApplicantIncome    614 non-null    int64  
 7   CoapplicantIncome  614 non-null    float64
 8   LoanAmount         592 non-null    float64
 9   Loan_Amount_Term   600 non-null    float64
 10  Credit_History     564 non-null    float64
 11  Property_Area      614 non-null    object 
 12  Loan_Status        614 non-null    object 
dtypes: float64(4), int64(1), object(8)
memory usage: 62.5+ KB
```

```
df.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
```

```
df=df.dropna()
```

```
df.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

df

      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         3+      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           N
2               1.0         Urban           Y
3               1.0         Urban           Y
4               1.0         Urban           Y
5               1.0         Urban           Y
..              ...           ...         ...
609             1.0         Rural           Y
610             1.0         Rural           Y
611             1.0         Urban           Y
612             1.0         Urban           Y
613             0.0     Semiurban           N

[480 rows x 13 columns]
```

```
df.reset_index(inplace=True)

df
```

```
     index   Loan_ID  Gender Married Dependents     Education
Self_Employed  \
0        1  LP001003    Male     Yes          1      Graduate
No
1        2  LP001005    Male     Yes          0      Graduate
Yes
2        3  LP001006    Male     Yes          0  Not Graduate
No
3        4  LP001008    Male      No          0      Graduate
No
4        5  LP001011    Male     Yes          2      Graduate
Yes
..     ...       ...     ...     ...        ...           ...
...
475    609  LP002978  Female      No          0      Graduate
No
476    610  LP002979    Male     Yes         3+      Graduate
No
477    611  LP002983    Male     Yes          1      Graduate
No
478    612  LP002984    Male     Yes          2      Graduate
No
479    613  LP002990  Female      No          0      Graduate
Yes
```

```
     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term
\
0               4583             1508.0       128.0             360.0

1               3000                0.0        66.0             360.0

2               2583             2358.0       120.0             360.0

3               6000                0.0       141.0             360.0

4               5417             4196.0       267.0             360.0

..               ...                ...         ...               ...

475             2900                0.0        71.0             360.0

476             4106                0.0        40.0             180.0

477             8072              240.0       253.0             360.0

478             7583                0.0       187.0             360.0

479             4583                0.0       133.0             360.0


     Credit_History Property_Area Loan_Status
0               1.0         Rural           N
1               1.0         Urban           Y
2               1.0         Urban           Y
3               1.0         Urban           Y
4               1.0         Urban           Y
..              ...           ...         ...
475             1.0         Rural           Y
476             1.0         Rural           Y
477             1.0         Urban           Y
478             1.0         Urban           Y
479             0.0     Semiurban           N

[480 rows x 14 columns]
```

```
df['Dependents'].unique()
```

```
array(['1', '0', '2', '3+'], dtype=object)
```

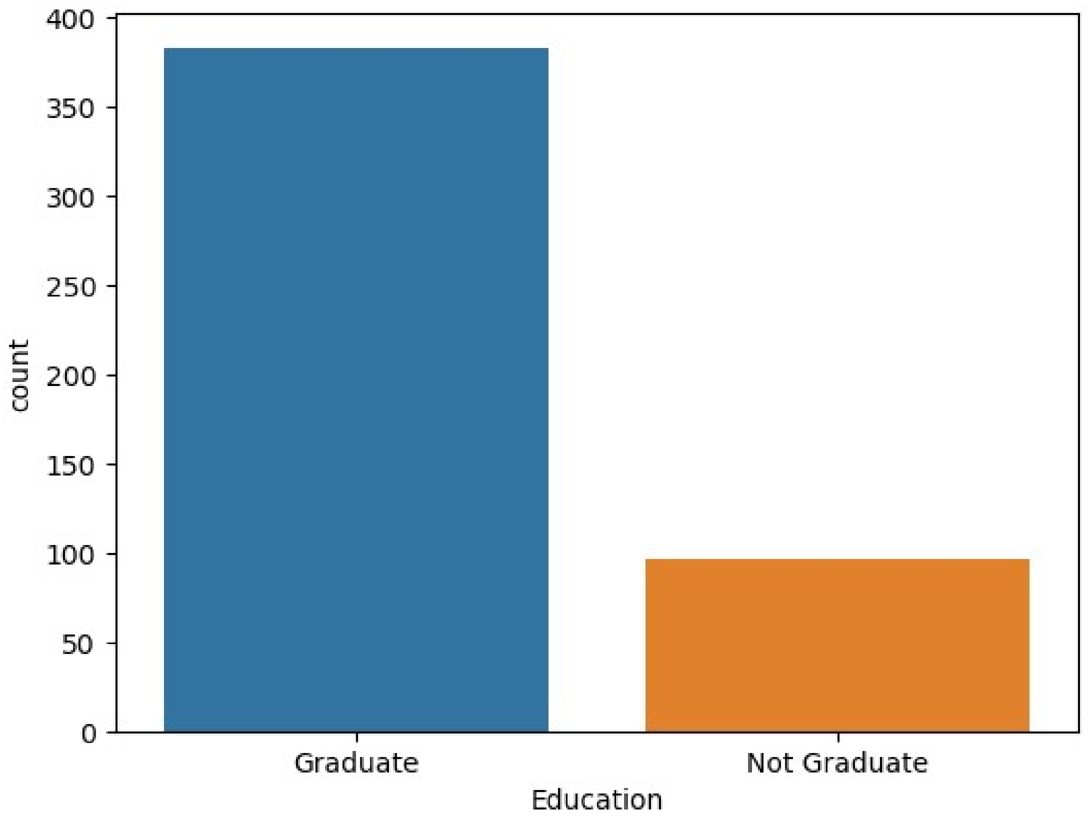
```
df['Dependents'].value_counts()
```

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

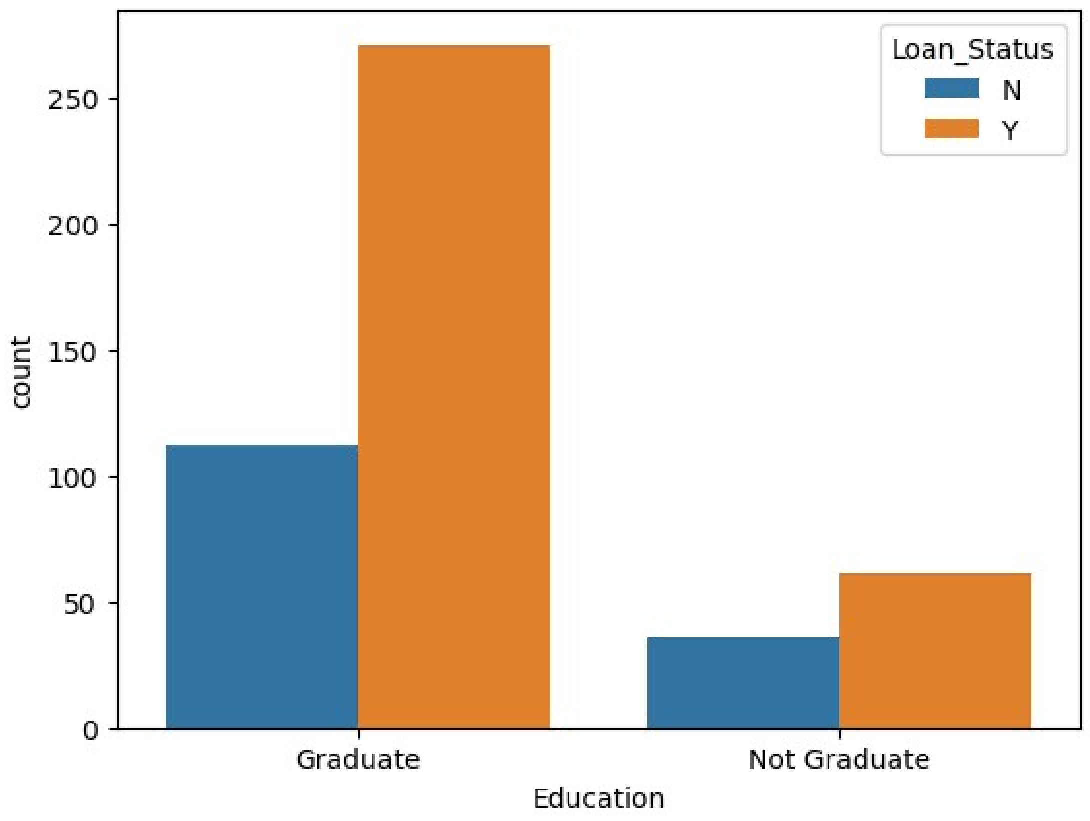
```
df.loc[df['Dependents'] == '3+', 'Dependents'] = 4

df['Dependents'].value_counts()

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

Visualization

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
sns.countplot(data=df, x='Education')

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

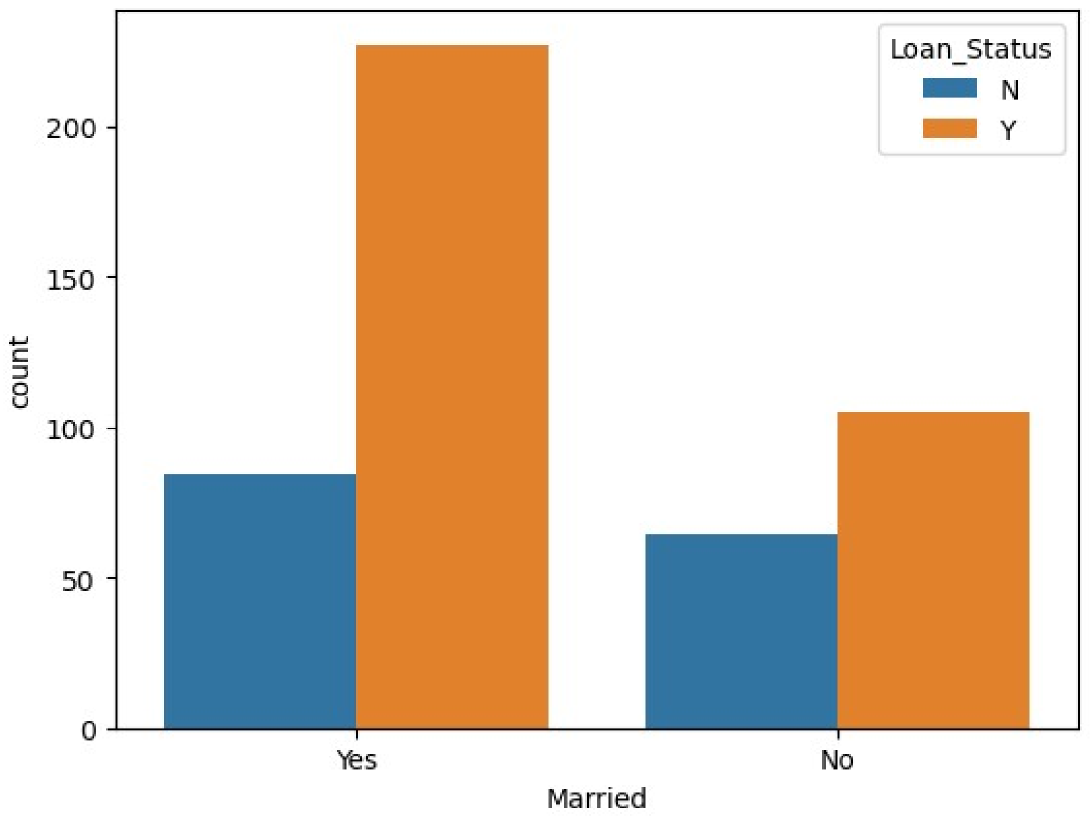


```
sns.countplot(x='Education',hue='Loan_Status',data=df )

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



```
sns.countplot(x='Married',hue='Loan_Status',data=df )
```

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



Encoding categorical values

```
df.head()

   index   Loan_ID Gender Married Dependents     Education
Self_Employed  \
0      1  LP001003   Male     Yes          1      Graduate
No
1      2  LP001005   Male     Yes          0      Graduate
Yes
2      3  LP001006   Male     Yes          0  Not Graduate
No
3      4  LP001008   Male      No          0      Graduate
No
4      5  LP001011   Male     Yes          2      Graduate
Yes

   ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
0             4583             1508.0       128.0             360.0
1             3000                0.0        66.0             360.0
2             2583             2358.0       120.0             360.0
3             6000                0.0       141.0             360.0
4             5417             4196.0       267.0             360.0

   Credit_History Property_Area Loan_Status
```

```
0              1.0         Rural           N
1              1.0         Urban           Y
2              1.0         Urban           Y
3              1.0         Urban           Y
4              1.0         Urban           Y
```

```
df.replace({ 'Married':{'Yes': 1, 'No': 0}, 'Gender':{'Male':1,
'Female': 0}, 'Education':{'Graduate':1, 'Not Graduate':0},
            'Self_Employed':{'Yes': 1, 'No': 0}, 'Property_Area':
{'Rural': 0, 'Urban':1, 'Semiurban':2}}, inplace=True)
```

```
C:\Users\Keerthi Priya\AppData\Local\Temp\
ipykernel_12200\1907869267.py:1: 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
  df.replace({ 'Married':{'Yes': 1, 'No': 0}, 'Gender':{'Male':1,
'Female': 0}, 'Education':{'Graduate':1, 'Not Graduate':0},
```

```
df.head()
```

```
   index   Loan_ID  Gender  Married Dependents  Education
Self_Employed  \
0      1  LP001003       1        1          1          1
0
1      2  LP001005       1        1          0          1
1
2      3  LP001006       1        1          0          0
0
3      4  LP001008       1        0          0          1
0
4      5  LP001011       1        1          2          1
1

   ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
0             4583             1508.0       128.0             360.0
1             3000                0.0        66.0             360.0
2             2583             2358.0       120.0             360.0
3             6000                0.0       141.0             360.0
4             5417             4196.0       267.0             360.0

   Credit_History  Property_Area Loan_Status
0             1.0              0           N
1             1.0              1           Y
2             1.0              1           Y
3             1.0              1           Y
4             1.0              1           Y
```

```
df_copy = df.copy()
df_copy.loc[:, 'Dependents'] = df_copy['Dependents'].astype('int')

X=df.iloc[:,2:-1].values

X[0]

array([1, 1, '1', 1, 0, 4583, 1508.0, 128.0, 360.0, 1.0, 0],
dtype=object)

df.replace({'Loan_Status': {'Y': 1, 'N': 0}}, inplace=True)

C:\Users\Keerthi Priya\AppData\Local\Temp\
ipykernel_12200\3483282469.py:1: 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
  df.replace({'Loan_Status': {'Y': 1, 'N': 0}}, inplace=True)

y=df.iloc[:,-1].values

y

array([0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1,
1,
       0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1,
1,
       0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1,
0,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1,
1,
       1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1,
1,
       1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0,
1,
       0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0,
1,
       0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1,
1,
       0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0,
1,
       1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1,
       1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1,
1,
       1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1,
0,
       0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0,
1,
       0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0,
1,
       1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1,
1,
       1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0,
0,
       0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1,
1,
       1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1,
1,
       1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,
1,
       1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1,
1,
       1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0,
1,
       0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0],
dtype=int64)
```

```
X
```

```
array([[1, 1, '1', ..., 360.0, 1.0, 0],
       [1, 1, '0', ..., 360.0, 1.0, 1],
       [1, 1, '0', ..., 360.0, 1.0, 1],
       ...,
       [1, 1, '1', ..., 360.0, 1.0, 1],
       [1, 1, '2', ..., 360.0, 1.0, 1],
       [0, 0, '0', ..., 360.0, 0.0, 2]], 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 =
0.25,random_state=42)
```

```
x_train.shape
```

```
(384, 11)
```

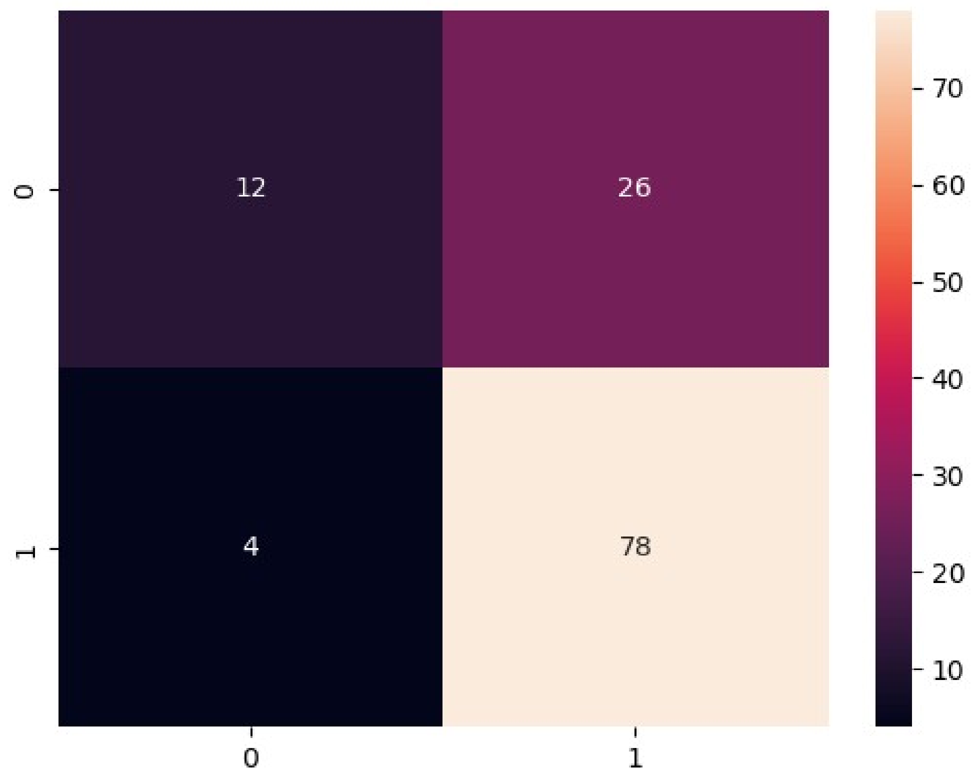
```
x_test.shape
```

```
(120, 11)
```

```
from sklearn.linear_model import LogisticRegression
log_classifier=LogisticRegression()
log_classifier.fit(x_train,y_train)
```

```
C:\Users\Keerthi Priya\anaconda3\lib\site-packages\sklearn\
linear_model\_logistic.py:814: ConvergenceWarning: lbfgs failed to
converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

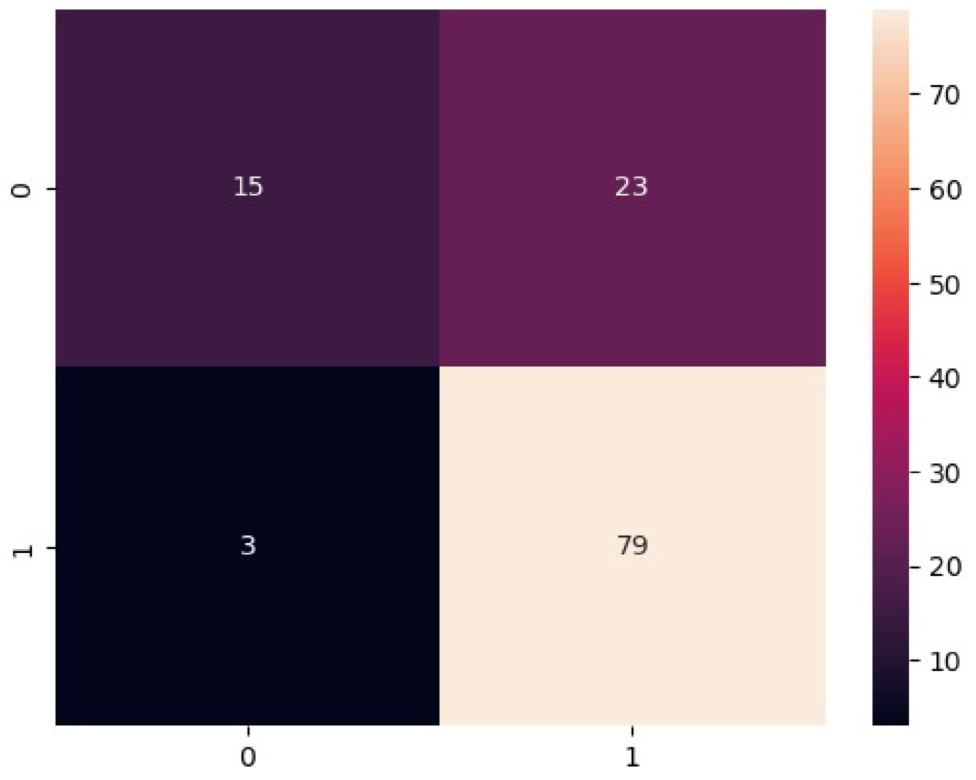
```
Increase the number of iterations (max_iter) or scale the data as
shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(

LogisticRegression()

log_y_pred=log_classifier.predict(x_test)

from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,log_y_pred)
cm
sns.heatmap(cm,annot=True)

<AxesSubplot:>
```



```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,log_y_pred)

0.75
```

Accuracy

```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,log_y_pred)from sklearn.metrics import
accuracy_score
accuracy_score(y_test,log_y_pred)

RandomForestClassifier(criterion='entropy', n_estimators=25)

y_pred=classifier.predict(x_test)

sns.heatmap(confusion_matrix(y_test,y_pred),annot=True)

<AxesSubplot:>
```



```
accuracy_score(y_test,y_pred)

0.7833333333333333
```