**import** **pandas** **as** **pd**

**import** **numpy** **as** **np**

**import** **seaborn** **as** **sns**

**from** **sklearn.model\_selection** **import** train\_test\_split

**from** **sklearn** **import** svm

**from** **sklearn.metrics** **import** accuracy\_score

In [31]:

loan\_data = pd.read\_csv('loan.csv')

In [32]:

print(loan\_data)

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]

In [33]:

loan\_data.head()

Out[33]:

|  | **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 [34]:

type(loan\_data)

Out[34]:

pandas.core.frame.DataFrame

In [35]:

loan\_data.shape

Out[35]:

(614, 13)

In [36]:

loan\_data.describe()

Out[36]:

|  | **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 [37]:

loan\_data.isnull().sum()

Out[37]:

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 [38]:

loan\_data = loan\_data.dropna()

In [39]:

loan\_data.isnull().sum()

Out[39]:

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 [40]:

loan\_data.replace({"Loan\_Status":{"N":0, "Y":1}}, inplace=**True**)

In [41]:

loan\_data.head()

Out[41]:

|  | **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 [42]:

loan\_data['Dependents'].value\_counts()

Out[42]:

0 274

2 85

1 80

3+ 41

Name: Dependents, dtype: int64

In [43]:

loan\_data = loan\_data.replace(to\_replace='3+', value=4)

In [44]:

loan\_data['Dependents'].value\_counts()

Out[44]:

0 274

2 85

1 80

4 41

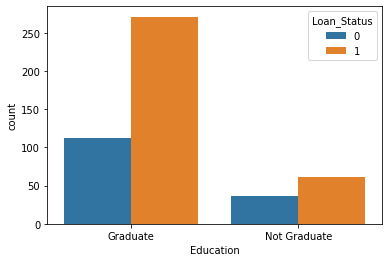
Name: Dependents, dtype: int64

In [49]:

sns.countplot(x='Education', hue='Loan\_Status', data=loan\_data)

Out[49]:

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

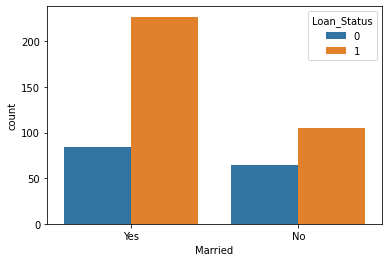


In [50]:

sns.countplot(x='Married', hue='Loan\_Status', data=loan\_data)

Out[50]:

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

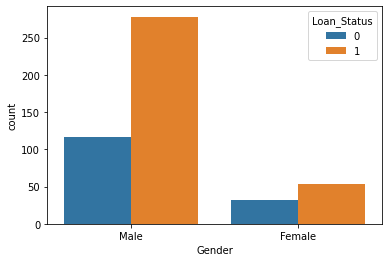


In [51]:

sns.countplot(x='Gender', hue='Loan\_Status', data=loan\_data)

Out[51]:

<AxesSubplot:xlabel='Gender', ylabel='count'>

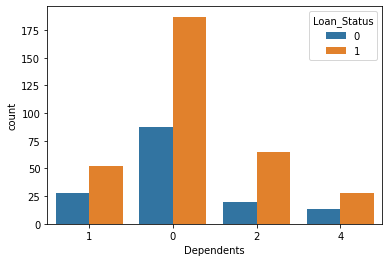


In [54]:

sns.countplot(x='Dependents', hue='Loan\_Status', data=loan\_data)

Out[54]:

<AxesSubplot:xlabel='Dependents', ylabel='count'>

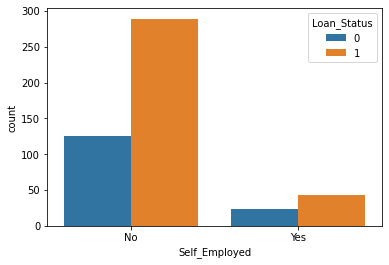


In [55]:

sns.countplot(x='Self\_Employed', hue='Loan\_Status', data=loan\_data)

Out[55]:

<AxesSubplot:xlabel='Self\_Employed', ylabel='count'>

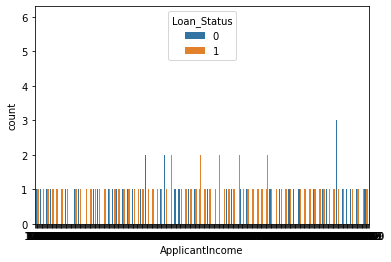


In [56]:

sns.countplot(x='ApplicantIncome', hue='Loan\_Status', data=loan\_data)

Out[56]:

<AxesSubplot:xlabel='ApplicantIncome', ylabel='count'>

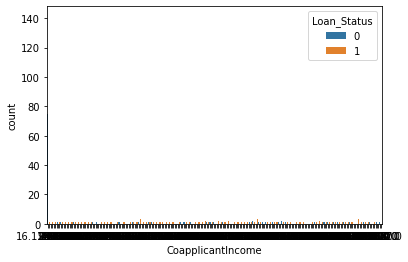


In [58]:

sns.countplot(x='CoapplicantIncome', hue='Loan\_Status', data=loan\_data)

Out[58]:

<AxesSubplot:xlabel='CoapplicantIncome', ylabel='count'>

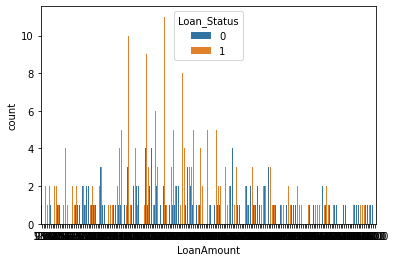


In [59]:

sns.countplot(x='LoanAmount', hue='Loan\_Status', data=loan\_data)

Out[59]:

<AxesSubplot:xlabel='LoanAmount', ylabel='count'>

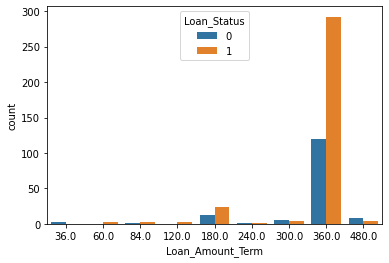


In [60]:

sns.countplot(x='Loan\_Amount\_Term', hue='Loan\_Status', data= loan\_data)

Out[60]:

<AxesSubplot:xlabel='Loan\_Amount\_Term', ylabel='count'>

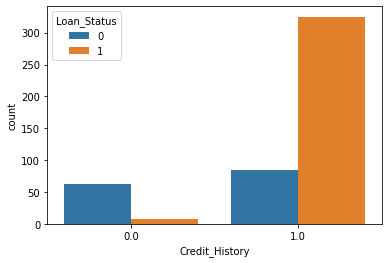


In [61]:

sns.countplot(x='Credit\_History', hue='Loan\_Status', data=loan\_data)

Out[61]:

<AxesSubplot:xlabel='Credit\_History', ylabel='count'>

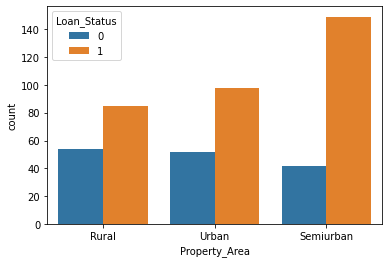


In [62]:

sns.countplot(x='Property\_Area', hue='Loan\_Status', data=loan\_data)

Out[62]:

<AxesSubplot:xlabel='Property\_Area', ylabel='count'>



In [63]:

loan\_data.replace({'Married':{'No':0, 'Yes':1}, 'Gender':{'Male':1, 'Female':0}, 'Self\_Employed':{'No':0, 'Yes':1},

'Property\_Area':{'Rural':0, 'Semiurban':1, 'Urban':2},'Education': {'Graduate':1, 'Not Graduate':0}},inplace=**True**)

In [64]:

loan\_data.head()

Out[64]:

|  | **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 | 2 | 1 |
| **3** | LP001006 | 1 | 1 | 0 | 0 | 0 | 2583 | 2358.0 | 120.0 | 360.0 | 1.0 | 2 | 1 |
| **4** | LP001008 | 1 | 0 | 0 | 1 | 0 | 6000 | 0.0 | 141.0 | 360.0 | 1.0 | 2 | 1 |
| **5** | LP001011 | 1 | 1 | 2 | 1 | 1 | 5417 | 4196.0 | 267.0 | 360.0 | 1.0 | 2 | 1 |

In [68]:

x = loan\_data.drop(columns=['Loan\_ID', 'Loan\_Status'], axis=1)

y = loan\_data['Loan\_Status']

In [69]:

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 2

3 2

4 2

5 2

.. ...

609 0

610 0

611 2

612 2

613 1

[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 [76]:

x\_train,x\_test,y\_train,y\_test = train\_test\_split(x,y, test\_size=0.1,stratify=y, random\_state=2)

In [77]:

print(x.shape, x\_train.shape,x\_test.shape)

(480, 11) (432, 11) (48, 11)

In [84]:

vector= svm.SVC(kernel='linear')

In [85]:

vector.fit(x\_train,y\_train)

Out[85]:

SVC(kernel='linear')

In [86]:

x\_train\_prediction = vector.predict(x\_train)

In [87]:

training\_data\_accuracy = accuracy\_score(y\_train, x\_train\_prediction)

In [89]:

print(training\_data\_accuracy)

0.7986111111111112

In [90]:

x\_test\_prediction=vector.predict(x\_test)

In [92]:

test\_data\_accuracy = accuracy\_score(y\_test, x\_test\_prediction)

In [93]:

print(test\_data\_accuracy)

0.8333333333333334

In [ ]:

In [ ]:

In [ ]:

In [ ]: