CHANDAN KUMAR

ID: GO_STP_13267

```
1 import numpy as np
2 import pandas as pd
```

```
1 df = pd.read_csv("/content/train_ctrUa4K.csv")
2 df.head()
```

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

```
1 df.tail()
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncom
609	LP002978	Female	No	0	Graduate	No	290
610	LP002979	Male	Yes	3+	Graduate	No	410
611	LP002983	Male	Yes	1	Graduate	No	807
612	LP002984	Male	Yes	2	Graduate	No	758
613	LP002990	Female	No	0	Graduate	Yes	458

```
1 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["Gender"].unique()
```

```
array(['Male', 'Female', nan], dtype=object)
```

```
df["Gender"] = df["Gender"].replace(["Female", "Male"], [1, 0])
df["Gender"] = df["Gender"].replace([np.nan],0)
df["Gender"].unique()
```

```
array([0., 1.])
```

```
df["Gender"].isnull().sum()
```

```
0
```

```
df["Married"].unique()
```

```
array(['No', 'Yes', nan], dtype=object)
```

```
df["Married"] = df["Married"].replace(["No", "Yes"], [0, 1])
df["Married"] = df["Married"].replace([np.nan],0)
df["Married"].unique()
```

```
array([0., 1.])
```

```
df["Married"].isnull().sum()
```

```
0
```

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

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

```
df["Dependents"] = df["Dependents"].replace([np.nan, '3+', "1", "2"],[0, 3, 1, 2])
df["Dependents"].unique()
```

```
array([0, 1, 2, 3])
```

```
df["Dependents"].isnull().sum()
```

```
0
```

```
df["Self_Employed"].unique()
```

```
array(['No', 'Yes', nan], dtype=object)
```

```
df["Self_Employed"] = df["Self_Employed"].replace(["No", "Yes"], [0, 1])
```

```
df['Self_Employed'] = df['Self_Employed'].replace(['No', 'Yes'], [0, 1])
df["Self_Employed"] = df["Self_Employed"].replace([np.nan],0)
df["Self_Employed"].unique()
```

```
array([0., 1.])
```

```
df["Self_Employed"].isnull().sum()
```

```
0
```

```
df["LoanAmount"] = df["LoanAmount"].fillna(method = "ffill")
df["LoanAmount"] = df["LoanAmount"].replace([np.nan],0)
df["LoanAmount"].isnull().sum()
```

```
0
```

```
df["LoanAmount"].unique()
```

```
array([  0., 128.,  66., 120., 141., 267.,  95., 158., 168., 349.,  70.,
       109., 200., 114.,  17., 125., 100.,  76., 133., 115., 104., 315.,
       116., 112., 151., 191., 122., 110.,  35., 201.,  74., 106., 320.,
       144., 184.,  80.,  47.,  75., 134.,  96.,  88.,  44., 286.,  97.,
       135., 180.,  99., 165., 258., 126., 312., 136., 172.,  81., 187.,
       113., 176., 130., 111., 167., 265.,  50., 210., 175., 131., 188.,
        25., 137., 160., 225., 216.,  94., 139., 152., 118., 185., 154.,
        85., 259., 194.,  93., 370., 182., 650., 102., 290.,  84., 242.,
       129.,  30., 244., 600., 255.,  98., 275., 121.,  63., 700.,  87.,
       101., 495.,  67.,  73., 260., 108.,  58.,  48., 164., 170.,  83.,
        90., 166., 124.,  55.,  59., 127., 214., 240.,  72.,  60., 138.,
        42., 280., 140., 155., 123., 279., 192., 304., 330., 150., 207.,
       436.,  78.,  54.,  89., 143., 105., 132., 480.,  56., 159., 300.,
       376., 117.,  71., 490., 173.,  46., 228., 308., 236., 570., 380.,
       296., 156., 103.,  45.,  65.,  53., 360.,  62., 218., 178., 239.,
       405., 148., 190., 149., 153., 162., 230.,  86., 234., 246., 500.,
       186., 119., 107., 209., 208., 243.,  40., 250., 311., 400., 161.,
       196., 324., 157., 145., 181.,  26., 211.,   9., 205.,  36.,  61.,
       146., 292., 142., 350., 496., 253.])
```

```
df["Loan_Amount_Term"] = df["Loan_Amount_Term"].fillna(method = "ffill")
df["Loan_Amount_Term"].unique()
```

```
array([360., 120., 240., 180.,  60., 300., 480.,  36.,  84.,  12.])
```

```
df["Loan_Amount_Term"].isnull().sum()
```

```
0
```

```
df["Credit_History"].unique()
```

```
array([ 1.,  0., nan])
```

```
df["Credit_History"] = df["Credit_History"].replace([np.nan],0)
df["Credit_History"].unique()
```

```
array([1., 0.])
```

```
df["Credit_History"].isnull().sum()
```

```
0
```

```
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["Education"].unique()
df["Education"] = df["Education"].replace(["Graduate", "Not Graduate"], [1, 0])
df["Education"].unique()
```

```
array([1, 0])
```

```
df["Property_Area"].unique()
```

```
array(['Urban', 'Rural', 'Semiurban'], dtype=object)
```

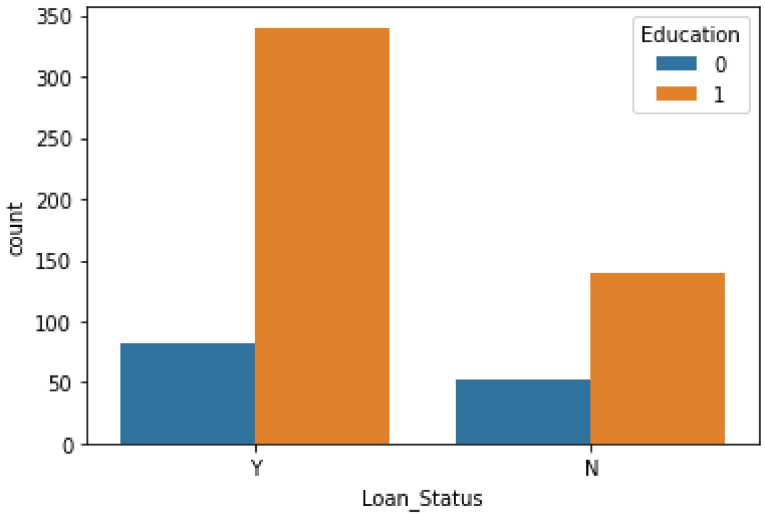
```
df["Property_Area"] = df["Property_Area"].replace(["Urban", "Rural", "Semiurban"], [1,
df["Property_Area"].unique()
```

```
array([1, 2, 3])
```

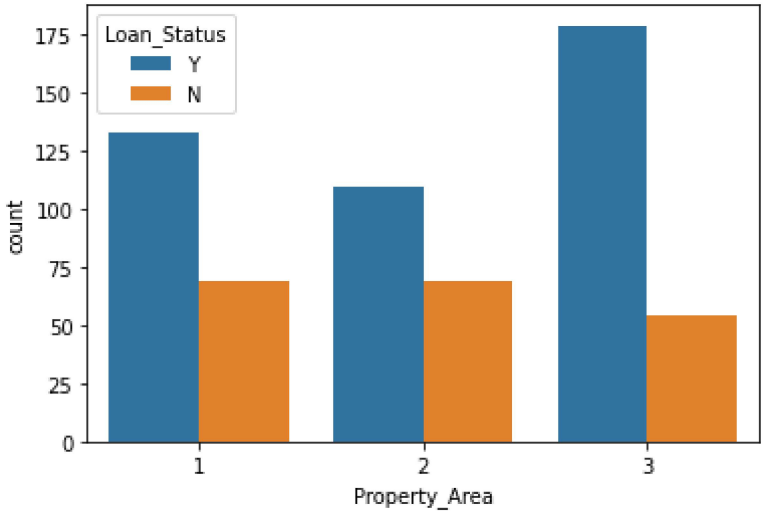
```
import matplotlib.pyplot as plt
import seaborn as sb
```

```
sb.countplot(x = "Education", data = df);
```


```
1 sb.countplot(x = "Loan_Status", hue = "Education", data = df);
```

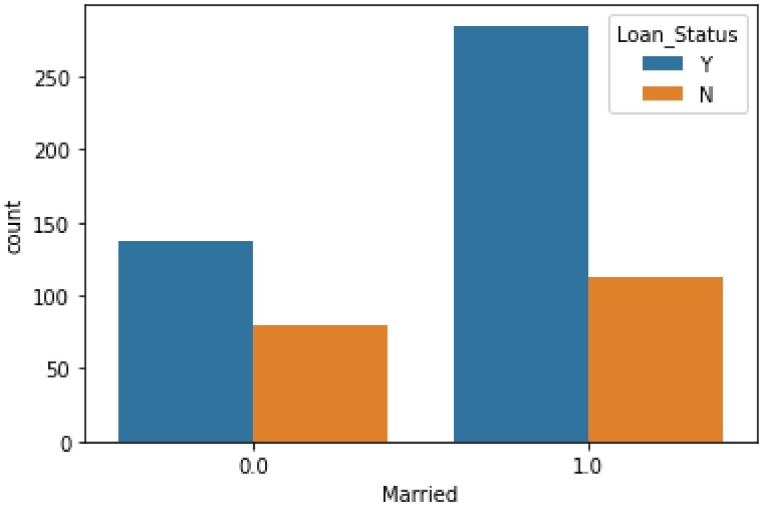



```
1 sb.countplot(x = "Property_Area", hue = "Loan_Status", data = df);
```

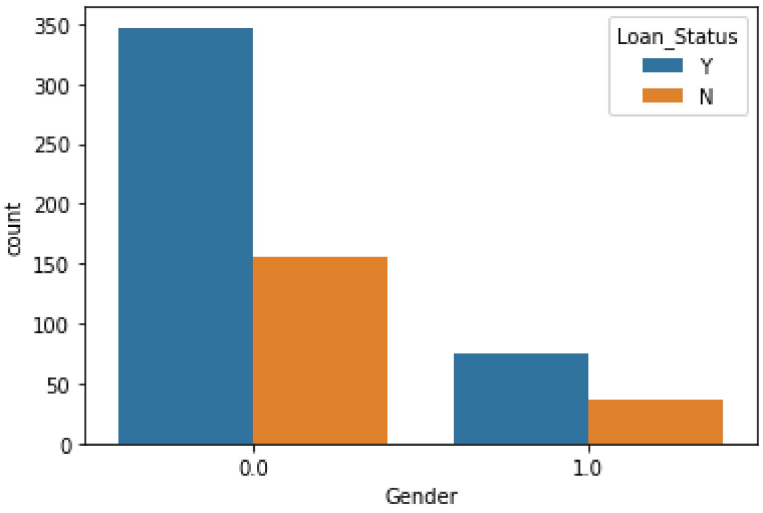



```
1 sb.countplot(x = "Self_Employed", hue = "Loan_Status", data = df);
```


```
1 sb.countplot(x = "Married", hue = "Loan_Status", data = df);
```




```
1 sb.countplot(x = "Gender", hue = "Loan_Status", data = df);
```




```
1 x = df[["Gender", "Married", "Dependents", "Education", "Self_Employed", "ApplicantInco
2 y = df["Loan_Status"]
```

```
1 from sklearn.model_selection import train_test_split
2 xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size = 0.3, random_state = 3
3 print(xtrain.shape)
4 print(ytrain.shape)
5 print(xtest.shape)
6 print(ytest.shape)
```

```
(429, 11)
(429,)
(185, 11)
(185,)
```

```
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier(max_depth = 4)
```

```
model.fit(xtrain, ytrain)
ypred = model.predict(xtest)
```

```
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(ypred, ytest)
print(accuracy*100)
```

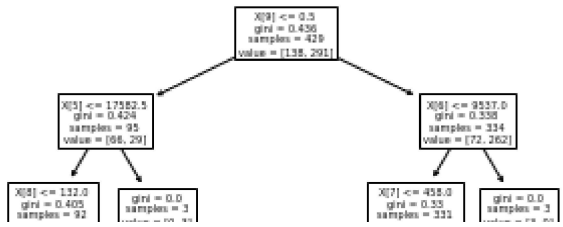
```
76.21621621621621
```

```
from sklearn.metrics import confusion_matrix
matrix = confusion_matrix(ypred, ytest)
print(matrix)
```

```
[[ 28  18]
 [ 26 113]]
```

```
from sklearn import tree
tree.plot_tree(model)
tree.export_graphviz(model)
```

```
'digraph Tree {\nnode [shape=box] ;\n0 [label="X[9] <= 0.5\\ngini = 0.436\\nsamples
= 429\\nvalue = [138, 291]"] ;\n1 [label="X[5] <= 17582.5\\ngini = 0.424\\nsamples =
95\\nvalue = [66, 29]"] ;\n0 -> 1 [labeldistance=2.5, labelangle=45, headlabel="Tru
e"] ;\n2 [label="X[8] <= 132.0\\ngini = 0.405\\nsamples = 92\\nvalue = [66, 26]"]
;\n1 -> 2 ;\n3 [label="gini = 0.0\\nsamples = 2\\nvalue = [0, 2]"] ;\n2 -> 3 ;\n4 [l
abel="X[5] <= 2554.5\\ngini = 0.391\\nsamples = 90\\nvalue = [66, 24]"] ;\n2 -> 4
;\n5 [label="gini = 0.498\\nsamples = 17\\nvalue = [9, 8]"] ;\n4 -> 5 ;\n6 [label="g
ini = 0.342\\nsamples = 73\\nvalue = [57, 16]"] ;\n4 -> 6 ;\n7 [label="gini = 0.0\\n
samples = 3\\nvalue = [0, 3]"] ;\n1 -> 7 ;\n8 [label="X[6] <= 9537.0\\ngini = 0.338
\\nsamples = 334\\nvalue = [72, 262]"] ;\n0 -> 8 [labeldistance=2.5, labelangle=-45,
headlabel="False"] ;\n9 [label="X[7] <= 458.0\\ngini = 0.33\\nsamples = 331\\nvalue
= [69, 262]"] ;\n8 -> 9 ;\n10 [label="X[10] <= 2.5\\ngini = 0.317\\nsamples = 324...'
```



✓ 1s completed at 5:33 PM