

```
# Aadesh Gulumbe 71
# LP-3
```

```
import pandas as pd
import seaborn as sns
```

```
df = pd.read_csv('Admission_Predict.csv')
```

```
df
```

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR
CGPA \						
0	1	337	118	4	4.5	4.5
9.65						
1	2	324	107	4	4.0	4.5
8.87						
2	3	316	104	3	3.0	3.5
8.00						
3	4	322	110	3	3.5	2.5
8.67						
4	5	314	103	2	2.0	3.0
8.21						
..
...						
395	396	324	110	3	3.5	3.5
9.04						
396	397	325	107	3	3.0	3.5
9.11						
397	398	330	116	4	5.0	4.5
9.45						
398	399	312	103	3	3.5	4.0
8.78						
399	400	333	117	4	5.0	4.0
9.66						

	Research	Chance of Admit
0	1	0.92
1	1	0.76
2	1	0.72
3	1	0.80
4	0	0.65
..
395	1	0.82
396	1	0.84
397	1	0.91
398	0	0.67
399	1	0.95

```
[400 rows x 9 columns]
```

```
from sklearn.preprocessing import Binarizer
bi = Binarizer(threshold=0.75)
df['Chance of Admit '] = bi.fit_transform(df[['Chance of Admit ']])
df.head()
```

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR
CGPA \						
0	1	337	118	4	4.5	4.5
9.65						
1	2	324	107	4	4.0	4.5
8.87						
2	3	316	104	3	3.0	3.5
8.00						
3	4	322	110	3	3.5	2.5
8.67						
4	5	314	103	2	2.0	3.0
8.21						

	Research	Chance of Admit
0	1	1.0
1	1	1.0
2	1	0.0
3	1	1.0
4	0	0.0

```
x = df.drop('Chance of Admit ',axis=1)
y=df['Chance of Admit ']
```

x

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR
CGPA \						
0	1	337	118	4	4.5	4.5
9.65						
1	2	324	107	4	4.0	4.5
8.87						
2	3	316	104	3	3.0	3.5
8.00						
3	4	322	110	3	3.5	2.5
8.67						
4	5	314	103	2	2.0	3.0
8.21						
...
...						
395	396	324	110	3	3.5	3.5
9.04						
396	397	325	107	3	3.0	3.5
9.11						
397	398	330	116	4	5.0	4.5

```

9.45
398      399      312      103      3  3.5  4.0
8.78
399      400      333      117      4  5.0  4.0
9.66

```

```

      Research
0          1
1          1
2          1
3          1
4          0
..        ...
395        1
396        1
397        1
398        0
399        1

```

```
[400 rows x 8 columns]
```

```
y
```

```

0      1.0
1      1.0
2      0.0
3      1.0
4      0.0
...
395    1.0
396    1.0
397    1.0
398    0.0
399    1.0

```

```
Name: Chance of Admit , Length: 400, dtype: float64
```

```
y=y.astype('int')
```

```
y
```

```

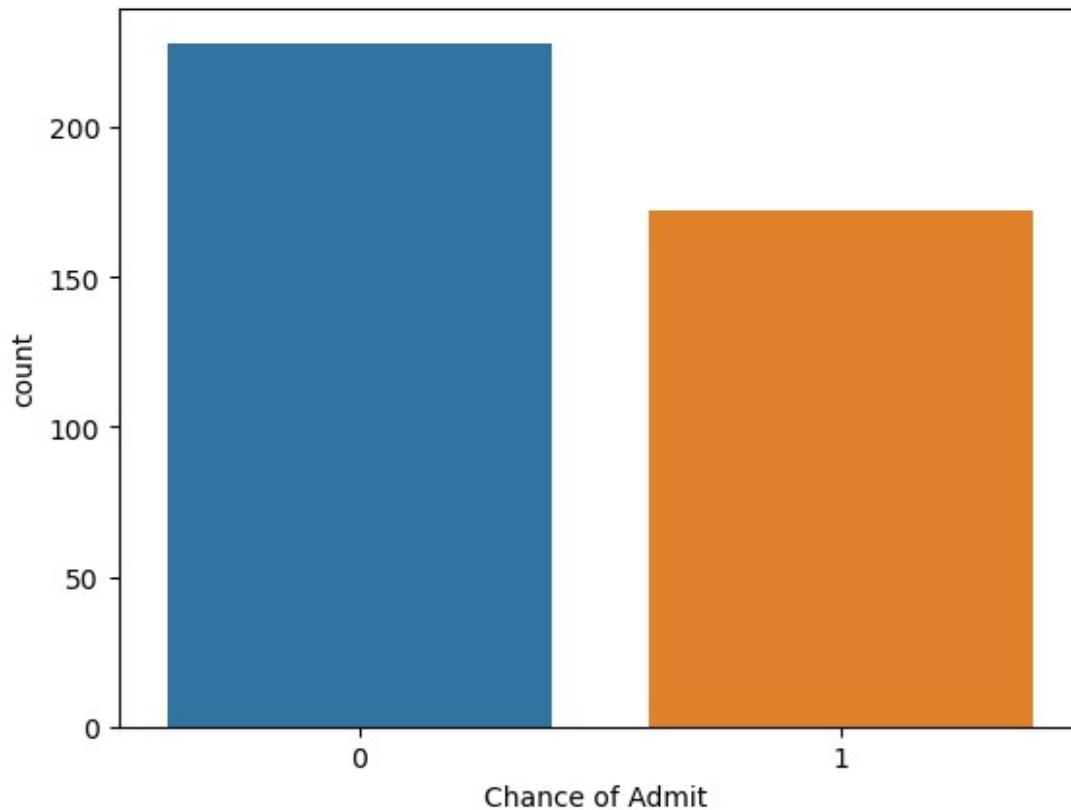
0      1
1      1
2      0
3      1
4      0
..
395    1
396    1
397    1
398    0

```

```
399      1
Name: Chance of Admit , Length: 400, dtype: int32

sns.countplot(x=y)

<Axes: xlabel='Chance of Admit ', ylabel='count'>
```



```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test =
train_test_split(x,y,random_state=0, test_size=0.2)

x_train.shape
(320, 8)

x_test.shape
(80, 8)

y_train.shape
(320,)

y_test.shape
(80,)
```

```

from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(random_state=0)

classifier.fit(x_train,y_train)

DecisionTreeClassifier(random_state=0)

y_pred = classifier.predict(x_test)

result=pd.DataFrame({'actual' : y_test, 'predicted':y_pred})

result

```

	actual	predicted
132	0	0
309	0	0
341	1	1
196	0	0
246	0	1
..
14	0	0
363	0	0
304	0	0
361	1	1
329	0	0

```
[80 rows x 2 columns]
```

```

from sklearn.metrics import ConfusionMatrixDisplay, accuracy_score
from sklearn.metrics import classification_report

```

```
accuracy_score(y_test,y_pred)
```

```
0.9
```

```

from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred, labels = classifier.classes_)

disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels =
classifier.classes_)

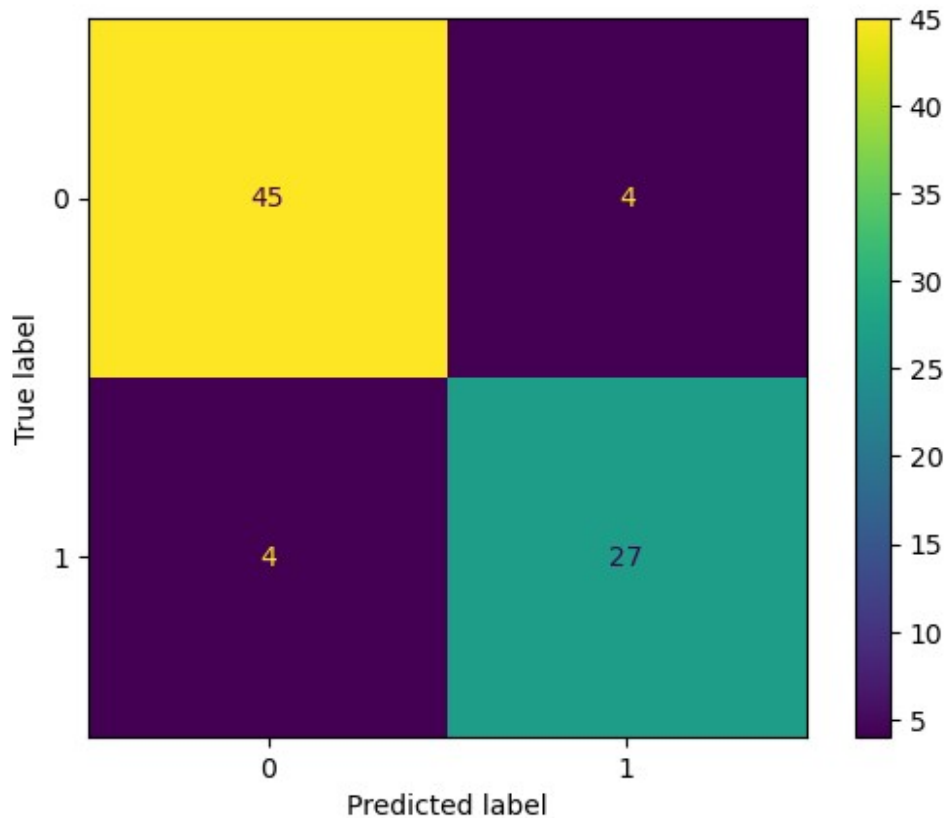
disp.plot()

```

```

<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at
0x1ec0cla25f0>

```



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

	precision	recall	f1-score	support
0	0.92	0.92	0.92	49
1	0.87	0.87	0.87	31
accuracy			0.90	80
macro avg	0.89	0.89	0.89	80
weighted avg	0.90	0.90	0.90	80

```
new = [[140,300,110,5,4.5,4.5,9.2,1]]
```

```
classifier.predict(new)[0]
```

```
C:\Users\STUDENT\anaconda3\lib\site-packages\sklearn\base.py:420:
UserWarning: X does not have valid feature names, but
DecisionTreeClassifier was fitted with feature names
warnings.warn(
```

```
1
```

```
from sklearn.tree import plot_tree
import matplotlib.pyplot as plt
```

