

```
In [1]: import pandas as pd
import seaborn as sns
```

```
In [2]: df = pd.read_csv('Admission_Predict.csv')
```

```
In [3]: df.columns
```

```
Out[3]: Index(['Serial No.', 'GRE Score', 'TOEFL Score', 'University Rating', 'SOP',
              'LOR ', 'CGPA', 'Research', 'Chance of Admit '],
              dtype='object')
```

```
In [4]: df.shape
```

```
Out[4]: (400, 9)
```

```
In [5]: df.head()
```

```
Out[5]:
```

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
0	1	337	118	4	4.5	4.5	9.65	1	0.92
1	2	324	107	4	4.0	4.5	8.87	1	0.76
2	3	316	104	3	3.0	3.5	8.00	1	0.72
3	4	322	110	3	3.5	2.5	8.67	1	0.80
4	5	314	103	2	2.0	3.0	8.21	0	0.65

```
In [6]: from sklearn.preprocessing import Binarizer
bi = Binarizer(threshold=0.75)
df['Chance of Admit '] = bi.fit_transform(df[['Chance of Admit ']])
```

```
In [7]: df.head()
```

```
Out[7]:
```

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
0	1	337	118	4	4.5	4.5	9.65	1	1.0
1	2	324	107	4	4.0	4.5	8.87	1	1.0
2	3	316	104	3	3.0	3.5	8.00	1	0.0
3	4	322	110	3	3.5	2.5	8.67	1	1.0
4	5	314	103	2	2.0	3.0	8.21	0	0.0

```
In [8]: x = df.drop('Chance of Admit ', axis = 1)
y = df['Chance of Admit ']
```

```
In [9]: x
```

Out[9]:

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

400 rows × 8 columns

In [13]: `y = y.astype('int')`
`y`

Out[13]:

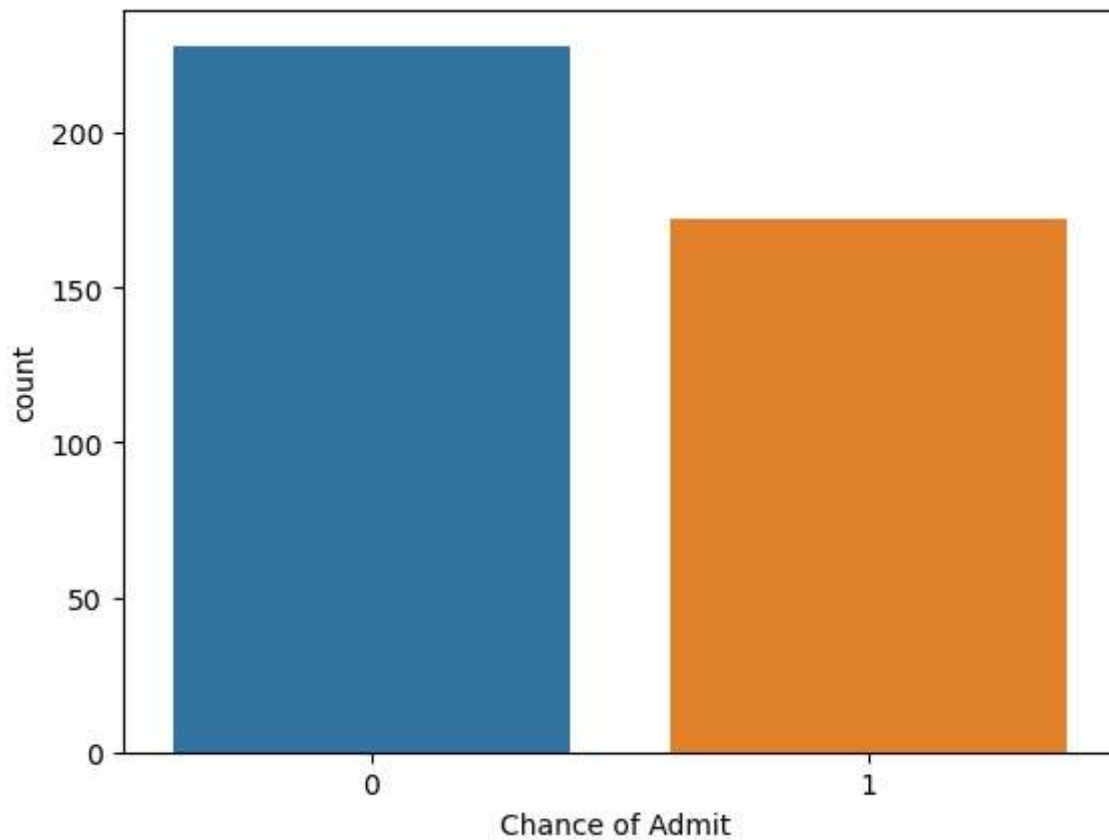
```

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

```

In [14]: `sns.countplot(x = y)`

Out[14]: `<Axes: xlabel='Chance of Admit ', ylabel='count'>`



```
In [15]: y.value_counts()
```

```
Out[15]: 0    228  
         1    172  
         Name: Chance of Admit , dtype: int64
```

```
In [16]: 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)
```

```
In [18]: x_train
```

Out[18]:

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research
250	251	320	104	3	3.0	2.5	8.57	1
63	64	315	107	2	4.0	3.0	8.50	1
312	313	311	107	4	4.5	4.5	9.00	1
159	160	297	100	1	1.5	2.0	7.90	0
283	284	321	111	3	2.5	3.0	8.90	1
...
323	324	305	102	2	2.0	2.5	8.18	0
192	193	322	114	5	4.5	4.0	8.94	1
117	118	290	104	4	2.0	2.5	7.46	0
47	48	339	119	5	4.5	4.0	9.70	0
172	173	322	110	4	4.0	5.0	9.13	1

300 rows × 8 columns

In [19]: `x_train.shape`

Out[19]: (300, 8)

In [20]: `from sklearn.tree import DecisionTreeClassifier`

In [21]: `classifier = DecisionTreeClassifier(random_state=0)`

In [22]: `classifier.fit(x_train, y_train)`

Out[22]:

▼ DecisionTreeClassifier

DecisionTreeClassifier(random_state=0)

In [23]: `y_pred = classifier.predict(x_test)`

In [24]:

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

Out[24]:

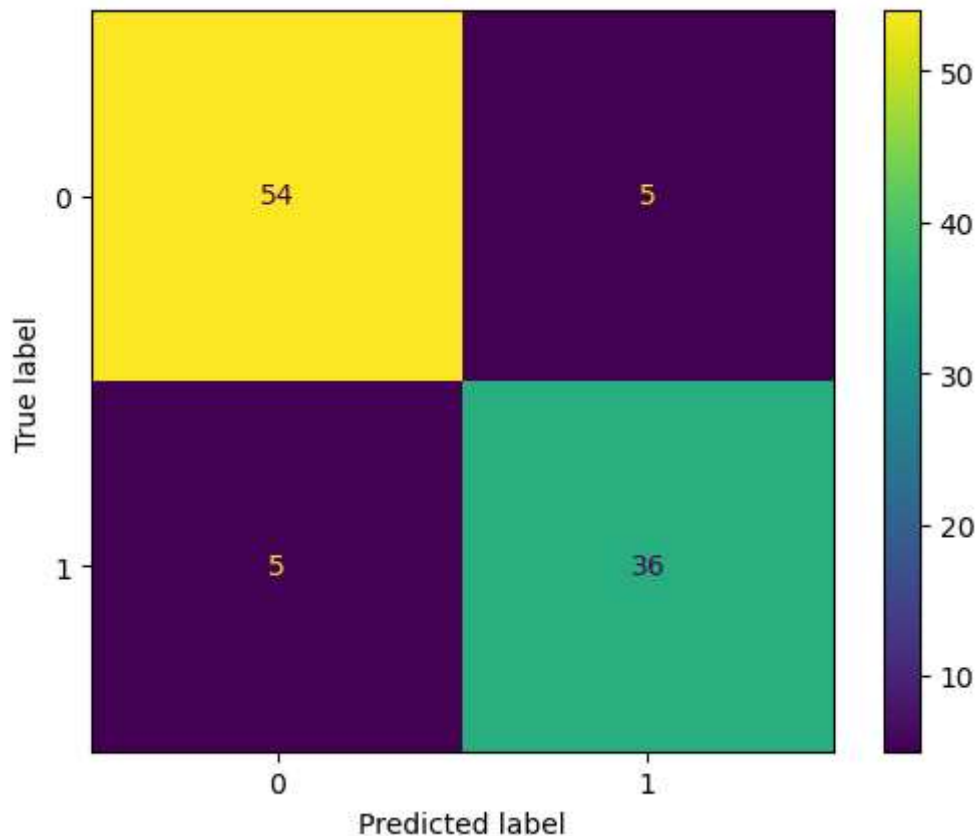
	actual	predicted
132	0	0
309	0	0
341	1	1
196	0	0
246	0	1
...
146	0	0
135	1	1
390	0	0
264	0	0
364	1	1

100 rows × 2 columns

In [25]: `from sklearn.metrics import ConfusionMatrixDisplay, accuracy_score, classification_report`

In [26]: `ConfusionMatrixDisplay.from_predictions(y_test, y_pred)`

Out[26]: `<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x14851a11750>`



In [27]: `accuracy_score(y_test, y_pred)`

Out[27]: 0.9

In [28]: `print(classification_report(y_test, y_pred))`

	precision	recall	f1-score	support
0	0.92	0.92	0.92	59
1	0.88	0.88	0.88	41
accuracy			0.90	100
macro avg	0.90	0.90	0.90	100
weighted avg	0.90	0.90	0.90	100

In [29]: `new = [[48,339,119,5,4.5,4.0,9.70,0]]`
`classifier.predict(new)[0]`

C:\Users\omnic\anaconda3\Lib\site-packages\sklearn\base.py:439: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted with feature names
 warnings.warn(

Out[29]: 1

In [31]: `from sklearn.tree import plot_tree`
`import matplotlib.pyplot as plt`

In [34]: `plt.figure(figsize=(12,12))`
`plot_tree(classifier, fontsize=6, filled=True, rounded=True, feature_names=x.columns,`

```

Out[34]: [Text(0.565625, 0.95, 'CGPA <= 8.655\ngini = 0.492\nsamples = 300\nvalue = [169, 131]\n\nclass = NA'),
Text(0.365625, 0.85, 'LOR <= 4.25\ngini = 0.14\nsamples = 159\nvalue = [147, 12]\n\nclass = NA'),
Text(0.25625, 0.75, 'GRE Score <= 319.5\ngini = 0.089\nsamples = 150\nvalue = [143, 7]\n\nclass = NA'),
Text(0.1625, 0.65, 'SOP <= 4.75\ngini = 0.056\nsamples = 138\nvalue = [134, 4]\n\nclass = NA'),
Text(0.1, 0.55, 'CGPA <= 8.51\ngini = 0.043\nsamples = 135\nvalue = [132, 3]\n\nclass = NA'),
Text(0.05, 0.45, 'SOP <= 3.25\ngini = 0.017\nsamples = 119\nvalue = [118, 1]\n\nclass = NA'),
Text(0.025, 0.35, 'gini = 0.0\nsamples = 89\nvalue = [89, 0]\n\nclass = NA'),
Text(0.075, 0.35, 'GRE Score <= 310.5\ngini = 0.064\nsamples = 30\nvalue = [29, 1]\n\nclass = NA'),
Text(0.05, 0.25, 'gini = 0.0\nsamples = 18\nvalue = [18, 0]\n\nclass = NA'),
Text(0.1, 0.25, 'GRE Score <= 311.5\ngini = 0.153\nsamples = 12\nvalue = [11, 1]\n\nclass = NA'),
Text(0.075, 0.15, 'CGPA <= 8.355\ngini = 0.5\nsamples = 2\nvalue = [1, 1]\n\nclass = NA'),
Text(0.05, 0.05, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]\n\nclass = AD'),
Text(0.1, 0.05, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]\n\nclass = NA'),
Text(0.125, 0.15, 'gini = 0.0\nsamples = 10\nvalue = [10, 0]\n\nclass = NA'),
Text(0.15, 0.45, 'TOEFL Score <= 99.5\ngini = 0.219\nsamples = 16\nvalue = [14, 2]\n\nclass = NA'),
Text(0.125, 0.35, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]\n\nclass = AD'),
Text(0.175, 0.35, 'CGPA <= 8.525\ngini = 0.124\nsamples = 15\nvalue = [14, 1]\n\nclass = NA'),
Text(0.15, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]\n\nclass = AD'),
Text(0.2, 0.25, 'gini = 0.0\nsamples = 14\nvalue = [14, 0]\n\nclass = NA'),
Text(0.225, 0.55, 'Serial No. <= 123.5\ngini = 0.444\nsamples = 3\nvalue = [2, 1]\n\nclass = NA'),
Text(0.2, 0.45, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]\n\nclass = NA'),
Text(0.25, 0.45, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]\n\nclass = AD'),
Text(0.35, 0.65, 'Serial No. <= 341.0\ngini = 0.375\nsamples = 12\nvalue = [9, 3]\n\nclass = NA'),
Text(0.325, 0.55, 'GRE Score <= 326.0\ngini = 0.18\nsamples = 10\nvalue = [9, 1]\n\nclass = NA'),
Text(0.3, 0.45, 'gini = 0.0\nsamples = 7\nvalue = [7, 0]\n\nclass = NA'),
Text(0.35, 0.45, 'University Rating <= 3.5\ngini = 0.444\nsamples = 3\nvalue = [2, 1]\n\nclass = NA'),
Text(0.325, 0.35, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]\n\nclass = NA'),
Text(0.375, 0.35, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]\n\nclass = AD'),
Text(0.375, 0.55, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]\n\nclass = AD'),
Text(0.475, 0.75, 'TOEFL Score <= 104.5\ngini = 0.494\nsamples = 9\nvalue = [4, 5]\n\nclass = AD'),
Text(0.45, 0.65, 'TOEFL Score <= 101.0\ngini = 0.444\nsamples = 6\nvalue = [4, 2]\n\nclass = NA'),
Text(0.425, 0.55, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]\n\nclass = AD'),
Text(0.475, 0.55, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]\n\nclass = NA'),
Text(0.5, 0.65, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]\n\nclass = AD'),
Text(0.765625, 0.85, 'CGPA <= 8.845\ngini = 0.263\nsamples = 141\nvalue = [22, 119]\n\nclass = AD'),
Text(0.63125, 0.75, 'Serial No. <= 127.0\ngini = 0.499\nsamples = 38\nvalue = [18, 20]\n\nclass = AD'),
Text(0.55, 0.65, 'CGPA <= 8.675\ngini = 0.346\nsamples = 9\nvalue = [7, 2]\n\nclass = NA'),
Text(0.525, 0.55, 'SOP <= 2.75\ngini = 0.444\nsamples = 3\nvalue = [1, 2]\n\nclass = AD'),
Text(0.5, 0.45, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]\n\nclass = NA'),

```

```

Text(0.55, 0.45, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]\nclass = AD'),
Text(0.575, 0.55, 'gini = 0.0\nsamples = 6\nvalue = [6, 0]\nclass = NA'),
Text(0.7125, 0.65, 'University Rating <= 3.5\ngini = 0.471\nsamples = 29\nvalue = [1
1, 18]\nclass = AD'),
Text(0.65, 0.55, 'Serial No. <= 225.0\ngini = 0.499\nsamples = 21\nvalue = [10, 11]
\nclass = AD'),
Text(0.6, 0.45, 'CGPA <= 8.78\ngini = 0.278\nsamples = 6\nvalue = [1, 5]\nclass = A
D'),
Text(0.575, 0.35, 'gini = 0.0\nsamples = 5\nvalue = [0, 5]\nclass = AD'),
Text(0.625, 0.35, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]\nclass = NA'),
Text(0.7, 0.45, 'GRE Score <= 325.5\ngini = 0.48\nsamples = 15\nvalue = [9, 6]\nclass
= NA'),
Text(0.675, 0.35, 'CGPA <= 8.815\ngini = 0.426\nsamples = 13\nvalue = [9, 4]\nclass
= NA'),
Text(0.65, 0.25, 'GRE Score <= 317.5\ngini = 0.375\nsamples = 12\nvalue = [9, 3]\nclass
= NA'),
Text(0.625, 0.15, 'TOEFL Score <= 103.5\ngini = 0.5\nsamples = 6\nvalue = [3, 3]\nclass
= NA'),
Text(0.6, 0.05, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]\nclass = NA'),
Text(0.65, 0.05, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]\nclass = AD'),
Text(0.675, 0.15, 'gini = 0.0\nsamples = 6\nvalue = [6, 0]\nclass = NA'),
Text(0.7, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]\nclass = AD'),
Text(0.725, 0.35, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]\nclass = AD'),
Text(0.775, 0.55, 'Serial No. <= 346.0\ngini = 0.219\nsamples = 8\nvalue = [1, 7]\nclass
= AD'),
Text(0.75, 0.45, 'gini = 0.0\nsamples = 7\nvalue = [0, 7]\nclass = AD'),
Text(0.8, 0.45, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]\nclass = NA'),
Text(0.9, 0.75, 'TOEFL Score <= 106.5\ngini = 0.075\nsamples = 103\nvalue = [4, 99]
\nclass = AD'),
Text(0.85, 0.65, 'Serial No. <= 270.5\ngini = 0.444\nsamples = 3\nvalue = [2, 1]\nclass
= NA'),
Text(0.825, 0.55, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]\nclass = NA'),
Text(0.875, 0.55, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]\nclass = AD'),
Text(0.95, 0.65, 'LOR <= 3.25\ngini = 0.039\nsamples = 100\nvalue = [2, 98]\nclass
= AD'),
Text(0.925, 0.55, 'Serial No. <= 116.0\ngini = 0.298\nsamples = 11\nvalue = [2, 9]\nclass
= AD'),
Text(0.9, 0.45, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]\nclass = NA'),
Text(0.95, 0.45, 'gini = 0.0\nsamples = 9\nvalue = [0, 9]\nclass = AD'),
Text(0.975, 0.55, 'gini = 0.0\nsamples = 89\nvalue = [0, 89]\nclass = AD')]

```