

Assignment 03

Roll No. 33331

Title : Classification using Machine Learning using Decision Tree

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
In [25]: df=pd.read_csv('Admission_Predict.csv')
df.head()
```

```
Out[25]:
```

	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 [8]: df.tail()
```

```
Out[8]:
```

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
395	396	324	110	3	3.5	3.5	9.04	1	1
396	397	325	107	3	3.0	3.5	9.11	1	1
397	398	330	116	4	5.0	4.5	9.45	1	1
398	399	312	103	3	3.5	4.0	8.78	0	0
399	400	333	117	4	5.0	4.0	9.66	1	1

```
In [5]: df.isnull().sum()
```

```
Out[5]: Serial No.      0
GRE Score      0
TOEFL Score    0
University Rating  0
SOP            0
LOR            0
CGPA           0
Research       0
Chance of Admit  0
dtype: int64
```

```
In [6]: df.columns
```

```
Out[6]: Index(['Serial No.', 'GRE Score', 'TOEFL Score', 'University Rating', 'SOP',
              'LOR ', 'CGPA', 'Research', 'Chance of Admit '],
              dtype='object')
```

```
In [7]: #Converting chance of admit into binary format
#Label Encoding
df['Chance of Admit '] = df['Chance of Admit '].apply(lambda x : 0 if x < 0.8 else 1)
df
```

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
1	2	324	107	4	4.0	4.5	8.87	1	0
2	3	316	104	3	3.0	3.5	8.00	1	0
3	4	322	110	3	3.5	2.5	8.67	1	1
4	5	314	103	2	2.0	3.0	8.21	0	0
...
395	396	324	110	3	3.5	3.5	9.04	1	1
396	397	325	107	3	3.0	3.5	9.11	1	1
397	398	330	116	4	5.0	4.5	9.45	1	1
398	399	312	103	3	3.5	4.0	8.78	0	0
399	400	333	117	4	5.0	4.0	9.66	1	1

400 rows x 9 columns

```
In [9]: x=df.drop(['Chance of Admit ','Serial No.'],axis=1)
y=df['Chance of Admit ']
```

In [10]: x

Out[10]:

	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research
0	337	118	4	4.5	4.5	9.65	1
1	324	107	4	4.0	4.5	8.87	1
2	316	104	3	3.0	3.5	8.00	1
3	322	110	3	3.5	2.5	8.67	1
4	314	103	2	2.0	3.0	8.21	0
...
395	324	110	3	3.5	3.5	9.04	1
396	325	107	3	3.0	3.5	9.11	1
397	330	116	4	5.0	4.5	9.45	1
398	312	103	3	3.5	4.0	8.78	0
399	333	117	4	5.0	4.0	9.66	1

400 rows x 7 columns

In [11]: y

Out[11]:

```
0      1
1      0
2      0
3      1
4      0
..
395     1
396     1
397     1
398     0
399     1
Name: Chance of Admit , Length: 400, dtype: int64
```

```
In [27]: from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
from sklearn.metrics import classification_report

X_train,X_test,Y_train,Y_test = train_test_split(x,y,test_size=0.27,random_state=143)
```

```
In [28]: clf = DecisionTreeClassifier()

clf = clf.fit(X_train,Y_train)

y_pred = clf.predict(X_test)

print("Confusion matrix : ")
print(metrics.confusion_matrix(Y_test, y_pred))
```

```
Confusion matrix :
[[67  5]
 [ 9 27]]
```

```
Out[38]: [Text(0.3072916666666667, 0.9583333333333334, 'x[5] <= 8.845\ngini = 0.432\nsamples = 292\nvalue = [200,
Text(0.0833333333333333, 0.875, 'x[5] <= 8.63\ngini = 0.079\nsamples = 195\nvalue = [187, 8]'),
Text(0.0416666666666666, 0.7916666666666666, 'gini = 0.0\nsamples = 144\nvalue = [144, 0]'),
Text(0.125, 0.7916666666666666, 'x[6] <= 0.5\ngini = 0.265\nsamples = 51\nvalue = [43, 8]'),
Text(0.0833333333333333, 0.7083333333333334, 'gini = 0.0\nsamples = 23\nvalue = [23, 0]'),
Text(0.1666666666666666, 0.7083333333333334, 'x[3] <= 3.25\ngini = 0.408\nsamples = 28\nvalue = [20, 8]'),
Text(0.125, 0.625, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),
Text(0.2083333333333334, 0.625, 'x[1] <= 105.5\ngini = 0.48\nsamples = 20\nvalue = [12, 8]'),
Text(0.1666666666666666, 0.5416666666666666, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
Text(0.25, 0.5416666666666666, 'x[5] <= 8.775\ngini = 0.5\nsamples = 16\nvalue = [8, 8]'),
Text(0.2083333333333334, 0.4583333333333333, 'x[2] <= 4.5\ngini = 0.49\nsamples = 14\nvalue = [6, 8]'),
Text(0.1666666666666666, 0.375, 'x[4] <= 3.25\ngini = 0.5\nsamples = 12\nvalue = [6, 6]'),
Text(0.0833333333333333, 0.2916666666666667, 'x[1] <= 114.0\ngini = 0.375\nsamples = 4\nvalue = [1, 3]'),
Text(0.0416666666666666, 0.2083333333333334, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
Text(0.125, 0.2083333333333334, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.25, 0.2916666666666667, 'x[4] <= 4.5\ngini = 0.469\nsamples = 8\nvalue = [5, 3]'),
Text(0.2083333333333334, 0.2083333333333334, 'x[5] <= 8.725\ngini = 0.408\nsamples = 7\nvalue = [5, 2]'),
Text(0.1666666666666666, 0.125, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
Text(0.25, 0.125, 'x[2] <= 3.5\ngini = 0.444\nsamples = 3\nvalue = [1, 2]'),
Text(0.2083333333333334, 0.0416666666666666, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.2916666666666667, 0.0416666666666666, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
Text(0.2916666666666667, 0.2083333333333334, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.25, 0.375, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
Text(0.2916666666666667, 0.4583333333333333, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
Text(0.53125, 0.875, 'x[0] <= 319.0\ngini = 0.232\nsamples = 97\nvalue = [13, 84]'),
Text(0.2916666666666667, 0.7916666666666666, 'x[5] <= 9.02\ngini = 0.245\nsamples = 7\nvalue = [6, 1]'),
Text(0.25, 0.7083333333333334, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
Text(0.3333333333333333, 0.7083333333333334, 'x[0] <= 312.0\ngini = 0.444\nsamples = 3\nvalue = [2, 1]'),
Text(0.2916666666666667, 0.625, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.375, 0.625, 'x[5] <= 9.13\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.3333333333333333, 0.5416666666666666, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.4166666666666667, 0.5416666666666666, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.7708333333333334, 0.7916666666666666, 'x[5] <= 9.105\ngini = 0.143\nsamples = 90\nvalue = [7, 83]'),
Text(0.625, 0.7083333333333334, 'x[0] <= 323.5\ngini = 0.408\nsamples = 21\nvalue = [6, 15]'),
Text(0.5416666666666666, 0.625, 'x[0] <= 321.5\ngini = 0.48\nsamples = 5\nvalue = [3, 2]'),
Text(0.5, 0.5416666666666666, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
Text(0.5833333333333334, 0.5416666666666666, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
Text(0.7083333333333334, 0.625, 'x[5] <= 9.03\ngini = 0.305\nsamples = 16\nvalue = [3, 13]'),
Text(0.6666666666666666, 0.5416666666666666, 'gini = 0.0\nsamples = 8\nvalue = [0, 8]'),
Text(0.75, 0.5416666666666666, 'x[1] <= 109.0\ngini = 0.469\nsamples = 8\nvalue = [3, 5]'),
Text(0.7083333333333334, 0.4583333333333333, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.7916666666666666, 0.4583333333333333, 'x[3] <= 4.25\ngini = 0.408\nsamples = 7\nvalue = [2, 5]'),
Text(0.75, 0.375, 'x[1] <= 111.0\ngini = 0.5\nsamples = 4\nvalue = [2, 2]'),
Text(0.7083333333333334, 0.2916666666666667, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.7916666666666666, 0.2916666666666667, 'x[5] <= 9.07\ngini = 0.444\nsamples = 3\nvalue = [2, 1]'),
Text(0.75, 0.2083333333333334, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.8333333333333334, 0.2083333333333334, 'x[3] <= 3.75\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.7916666666666666, 0.125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.875, 0.125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.8333333333333334, 0.375, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
Text(0.9166666666666666, 0.7083333333333334, 'x[5] <= 9.165\ngini = 0.029\nsamples = 69\nvalue = [1, 68]'),
Text(0.875, 0.625, 'x[5] <= 9.155\ngini = 0.133\nsamples = 14\nvalue = [1, 13]'),
Text(0.8333333333333334, 0.5416666666666666, 'gini = 0.0\nsamples = 13\nvalue = [0, 13]'),
Text(0.9166666666666666, 0.5416666666666666, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.9583333333333334, 0.625, 'gini = 0.0\nsamples = 55\nvalue = [0, 55]')]
```

