```
import pandas as pd
df=pd.read_csv("Admission_Predict.csv")
df
```

uı						
CGPA		GRE Score	TOEFL Score	University Rating	S0P	L0R
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		312	103	3	3.5	4.0
8.78 399	400	333	117	4	5.0	4.0
9.66						
0 1 2 3	Research C 1 1 1 1		mit 0.92 0.76 0.72 0.80			

Research	Chance	of	Admit
1			0.92
1			0.76
1			0.72
1			0.80
0			0.65
1			0.82
1			0.84
1			0.91
0			0.67
1			0.95
	1 1 1 0  1 1	1 1 1 0  1 1	1 1 1 0  1 1

[400 rows x 9 columns]

 $\begin{array}{l} \texttt{df=df.drop(df.columns[0],axis=1)} \\ \texttt{df} \end{array}$ 

GRE	Score	TOEFL Score	University Rating	S0P	L0R	CGPA
Research	\					
0	337	118	4	4.5	4.5	9.65
1						

324	107		4	4.0	4.5	8.87
316	104		3	3.0	3.5	8.00
322	110		3	3.5	2.5	8.67
314	103		2	2.0	3.0	8.21
324	110		3	3.5	3.5	9.04
325	107		3	3.0	3.5	9.11
330	116		4	5.0	4.5	9.45
312	103		3	3.5	4.0	8.78
333	117		4	5.0	4.0	9.66
Chance of Admit 0.92 0.76 0.72 0.80 0.65 0.82 0.84 0.91 0.67 0.95						
	316 322 314 324 325 330 312 333  Chance of Admit 0.92 0.76 0.72 0.80 0.65 0.82 0.84 0.91 0.67	316 104 322 110 314 103 324 110 325 107 330 116 312 103 333 117  Chance of Admit 0.92 0.76 0.72 0.80 0.65 0.82 0.84 0.91 0.67	316 104 322 110 314 103 324 110 325 107 330 116 312 103 333 117  Chance of Admit 0.92 0.76 0.72 0.80 0.65 0.82 0.84 0.91 0.67	316 104 3 322 110 3 314 103 2 324 110 3 325 107 3 330 116 4 312 103 3 333 117 4  Chance of Admit 0.92 0.76 0.72 0.80 0.65 0.82 0.84 0.91 0.67	316 104 3 3.0 322 110 3 3.5 314 103 2 2.0 324 110 3 3.5 325 107 3 3.0 330 116 4 5.0 312 103 3 3.5 333 117 4 5.0  Chance of Admit 0.92 0.76 0.72 0.80 0.65 0.82 0.84 0.91 0.67	316 104 3 3.0 3.5 322 110 3 3.5 2.5 314 103 2 2.0 3.0 324 110 3 3.5 3.5 325 107 3 3.0 3.5 330 116 4 5.0 4.5 312 103 3 3.5 4.0 333 117 4 5.0 4.0  Chance of Admit 0.92 0.76 0.72 0.80 0.65 0.82 0.84 0.91 0.67

## [400 rows x 8 columns]

## df.isna().sum()

GRE Score	0
TOEFL Score	0
University Rating	0
SOP	0
LOR	0
CGPA	0
Research	0
Chance of Admit	0
dtype: int64	

df.isnull().sum()

```
GRE Score
                          0
TOEFL Score
                          0
University Rating
                          0
S<sub>0</sub>P
                          0
LOR
                          0
CGPA
                          0
                          0
Research
Chance of Admit
                          0
dtype: int64
# df.loc[df['Chance of Admit '] >= 0.9, 'Admission']=1
# df.loc[df['Chance of Admit '] < 0.9, 'Admission']=0
df.loc[df['Chance of Admit '] >= 0.9, 'Admission']='YES'
df.loc[df['Chance of Admit '] < 0.9, 'Admission']='NO'
df
      GRE Score TOEFL Score University Rating SOP
                                                                 LOR
                                                                         CGPA
Research \
                                                        4 4.5
                                                                   4.5
0
             337
                              118
                                                                         9.65
1
1
                              107
                                                        4
                                                           4.0
                                                                   4.5 8.87
             324
1
2
                              104
                                                        3
                                                           3.0
                                                                   3.5 8.00
             316
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
                                                           3.0
                                                                   3.5 9.11
             325
                              107
                                                        3
1
397
             330
                              116
                                                        4
                                                           5.0
                                                                   4.5 9.45
1
398
             312
                              103
                                                           3.5
                                                                   4.0 8.78
0
399
             333
                              117
                                                        4 5.0
                                                                   4.0 9.66
1
      Chance of Admit Admission
0
                     0.92
                                  YES
1
                     0.76
                                    NO
2
                     0.72
                                    NO
3
                     0.80
                                    NO
4
                     0.65
                                    NO.
                      . . .
                                   . . .
. .
395
                     0.82
                                    N0
396
                     0.84
                                    N0
```

397	0.91	YES
398	0.67	NO
399	0.95	YES

[400 rows x 9 columns]

df=df.drop(df.columns[7],axis=1)
df

	Score	TOEFL Score	University Rating	SOP	L0R	CGPA
Research	\	110	4	4 5	4 5	0 65
0 1	337	118	4	4.5	4.5	9.65
1	324	107	4	4.0	4.5	8.87
1 2	316	104	3	3.0	3.5	8.00
1						
2 1 3 1	322	110	3	3.5	2.5	8.67
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 1	333	117	4	5.0	4.0	9.66

	Admission
0	YES
1	NO
2	NO
3	NO
4	NO
395	NO
396	NO
397	YES
398	NO
399	YES

[400 rows x 8 columns]

```
X=df[['GRE Score', 'TOEFL Score', 'University Rating', 'SOP', 'LOR
','CGPA','Research']]
     GRE Score TOEFL Score University Rating SOP LOR
                                                              CGPA
Research
           337
                         118
                                               4 4.5
                                                        4.5 9.65
1
1
           324
                         107
                                               4 4.0
                                                        4.5 8.87
1
2
           316
                         104
                                                  3.0
                                                        3.5 8.00
1
3
           322
                         110
                                               3
                                                  3.5
                                                         2.5 8.67
1
4
           314
                         103
                                                  2.0
                                                         3.0 8.21
                                               2
0
. .
                                                         . . .
                         . . .
                                                  3.5
                                                        3.5 9.04
           324
                         110
                                               3
395
1
396
           325
                         107
                                                  3.0
                                                         3.5 9.11
                                               3
1
397
           330
                         116
                                               4
                                                  5.0
                                                        4.5 9.45
1
398
           312
                         103
                                                  3.5
                                                        4.0 8.78
0
                                                         4.0 9.66
399
           333
                         117
                                               4 5.0
1
[400 rows x 7 columns]
y=df[['Admission']]
У
    Admission
          YES
0
           NO
1
2
           NO
3
           NO
4
           NO
           . . .
395
           N0
396
           NO
397
          YES
398
           NO
399
          YES
[400 rows x 1 columns]
df.corr()
```

	GRE Score	TOEFL Score	University Rating	S0P
\ GRE Score	1.000000	0.835977	0.668976	0.612831
TOEFL Score	0.835977	1.000000	0.695590	0.657981
University Rating	0.668976	0.695590	1.000000	0.734523
SOP	0.612831	0.657981	0.734523	1.000000
LOR	0.557555	0.567721	0.660123	0.729593
CGPA	0.833060	0.828417	0.746479	0.718144
Research	0.580391	0.489858	0.447783	0.444029

	LOR	CGPA	Research
GRE Score	0.557555	0.833060	0.580391
TOEFL Score	0.567721	0.828417	0.489858
University Rating	0.660123	0.746479	0.447783
SOP	0.729593	0.718144	0.444029
LOR	1.000000	0.670211	0.396859
CGPA	0.670211	1.000000	0.521654
Research	0.396859	0.521654	1.000000

from sklearn.model\_selection import train\_test\_split
X\_Train,X\_Test,y\_Train,y\_Test=train\_test\_split(X,y,test\_size=0.2,rando
m\_state=0)

## X\_Train

		TOEFL Score	University Rating	S0P	L0R	CGPA
Resear 336 0	319	110	3	3.0	2.5	8.79
64 0	325	111	3	3.0	3.5	8.70
55 0	320	103	3	3.0	3.0	7.70
106 1	329	111	4	4.5	4.5	9.18
300 0	309	106	2	2.5	2.5	8.00
• •						
323 0	305	102	2	2.0	2.5	8.18
192 1	322	114	5	4.5	4.0	8.94

117 0	290	104	4	2.0	2.5	7.46
47 0	339	119	5	4.5	4.0	9.70
172 1	322	110	4	4.0	5.0	9.13

[320 rows x 7 columns]

y\_Train

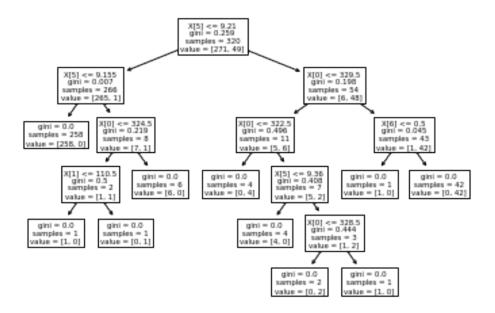
	Admission
336	NO
64	NO
55	NO
106	NO
300	NO
323	NO
192	NO
117	NO
47	NO
172	NO

[320 rows x 1 columns]

X\_Test

	Score	TOEFL Score	University Rating	S0P	L0R	CGPA
Research 132 0	309	105	5	3.5	3.5	8.56
309 0	308	110	4	3.5	3.0	8.60
341 1	326	110	3	3.5	3.5	8.76
196 0	306	105	2	3.0	2.5	8.26
246 0	316	105	3	3.0	3.5	8.73
• •						
14 1	311	104	3	3.5	2.0	8.20
363 0	306	103	2	2.5	3.0	8.36
304 0	313	106	2	2.5	2.0	8.43
361 1	334	116	4	4.0	3.5	9.54
329	297	96	2	2.5	1.5	7.89

```
[80 rows x 7 columns]
y Test
               Admission
132
309
                                          NO
341
                                          NO
196
                                          NO
246
                                          NO
                                        . . .
14
                                          N0
363
                                          NO
304
                                          NO
361
                                      YES
329
                                          NO
[80 rows x 1 columns]
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
clf=tree.DecisionTreeClassifier()
clf=clf.fit(X Train,y Train)
tree.plot tree(clf)
[Text(0.4230769230769231, 0.9166666666666666, 'X[5] \le 9.21 
0.259 \times = 320 \times = [271, 49]'),
   Text(0.15384615384615385, 0.75, 'X[5] \le 9.155  regin = 0.007 \ \ nsamples
= 266\nvalue = [265, 1]'),
   Text(0.07692307692307693, 0.5833333333333334, 'qini = 0.0 \nsamples =
258\nvalue = [258, 0]'),
   Text(0.23076923076923078, 0.5833333333333334, 'X[0] <= 324.5 \neq = 324.5
0.219 \times = 8 \times = [7, 1]'
   Text(0.15384615384615385, 0.416666666666667, 'X[1] <= 110.5 \ngini =
0.5 \times = 2 \times = [1, 1]'
   Text(0.07692307692307693, 0.25, 'gini = 0.0 \nsamples = 1 \nvalue = [1, ]
0]'),
   Text(0.23076923076923078, 0.25, 'gini = 0.0 \nsamples = 1 \nvalue = [0, ]
   Text(0.3076923076923077, 0.416666666666667, 'gini = 0.0 \nsamples = 0.0 \nsa
6\nvalue = [6, 0]'),
   Text(0.6923076923076923, 0.75, 'X[0] \le 329.5 \rangle = 0.198 \rangle
= 54 \nvalue = [6, 48]'),
   Text(0.5384615384615384, 0.5833333333333334, 'X[0] <= 322.5 \ngini =
0.496 \times 11 = [5, 6]'
   Text(0.46153846153846156, 0.416666666666667, 'gini = 0.0 \nsamples = 0.0 \ns
4\nvalue = [0, 4]'),
   Text(0.6153846153846154, 0.4166666666666667, 'X[5] <= 9.36 
0.408 \times = 7 \times = [5, 2]'
```



```
clf1=tree.DecisionTreeClassifier()
clf1=clf1.fit(X_Test,y_Test)
tree.plot_tree(clf1)

[Text(0.6, 0.875, 'X[5] <= 9.31\ngini = 0.18\nsamples = 80\nvalue =
[72, 8]'),
   Text(0.4, 0.625, 'X[1] <= 117.5\ngini = 0.053\nsamples = 74\nvalue =
[72, 2]'),
   Text(0.2, 0.375, 'gini = 0.0\nsamples = 71\nvalue = [71, 0]'),
   Text(0.6, 0.375, 'X[5] <= 9.135\ngini = 0.444\nsamples = 3\nvalue =
[1, 2]'),
   Text(0.4, 0.125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
   Text(0.8, 0.125, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
   Text(0.8, 0.625, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]')]</pre>
```

```
X[5] \le 9.31
                              gini = 0.18
                             samples = 80
                            value = [72, 8]
                  X[1] \le 117.5
                                         gini = 0.0
                   aini = 0.053
                                        samples = 6
                  samples = 74
                                        value = [0, 6]
                  value = [72, 2]
                             X[5] \le 9.135
         gini = 0.0
                              gini = 0.444
       samples = 71
                             samples = 3
       value = [71, 0]
                             value = [1, 2]
                    gini = 0.0
                                         gini = 0.0
                   samples = 1
                                        samples = 2
                  value = [1, 0]
                                        value = [0, 2]
yp=clf.predict(X Test)
уp
array(['N0', 'N0', 'N0', 'N0', 'N0', 'N0', 'N0', 'YES', 'YES',
'NO',
       'YES', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
'NO',
       'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
'NO',
       'NO', 'NO', 'NO', 'NO', 'NO', 'YES', 'NO', 'NO', 'NO',
'NO',
       'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
'NO',
       'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
'NO',
             'YES', 'NO', 'NO', 'NO', 'YES', 'NO', 'YES', 'NO',
       'NO',
       'NO', 'NO', 'YES', 'NO'], dtype=object)
yp1=clf.predict(X Train)
yp1
array(['N0', 'N0', 'N0', 'N0', 'N0', 'N0', 'N0', 'N0', 'N0', 'N0',
'NO',
       'YES', 'NO', 'NO', 'YES', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
'NO',
       'NO', 'NO', 'NO', 'NO', 'YES', 'NO', 'NO', 'NO', 'NO', 'NO',
'NO',
       'NO',
              'NO', 'YES',
                           'YES', 'NO', 'NO', 'NO', 'NO', 'NO',
                          'YES', 'NO', 'NO', 'NO', 'YES',
                                                            'NO', 'NO',
                    'YES',
              'NO',
       ' NO '
       'NO',
             'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
'NO',
       'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'YES', 'NO', 'NO',
```

```
'YES',
                                    'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'YES', 'NO', 'NO',
 'YES',
                                   'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
 'YES',
                                    'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
 'NO',
                                    'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'YES',
 'NO',
                                    'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
 'NO',
                                    'NO', 'YES', 'NO', 'YES', 'NO', 'YES', 'NO', 'NO', 'NO', 'NO',
                                    'NO', 'NO', 'NO', 'NO', 'YES', 'NO', 'NO', 'NO', 'YES', 'NO',
 'NO',
                                    'YES', 'YES', 'NO', 'NO', 'NO', 'NO', 'YES', 'NO', 'YES', 'NO', 'N
 'NO',
                                   'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
 'NO',
                                    'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
 'NO',
                                   'NO', 'NO', 'NO', 'NO', 'YES', 'NO', 'NO', 'NO', 'NO', 'NO',
 'NO',
                                   'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
 'YES',
                                    'YES', 'YES', 'NO', 'YES', 'NO', 'NO', 'YES', 'NO', 'NO', 'NO',
                                    'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
 'NO',
                                    'YES', 'YES', 'NO', 'NO', 'NO', 'NO', 'NO', 'YES', 'YES',
                                    'YES', 'NO', 'NO', 'NO', 'YES', 'NO', 'NO', 'NO', 'NO', 'NO',
 'NO',
                                   'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
 'NO',
                                   'NO', 'NO', 'NO', 'YES', 'NO', 'YES', 'NO', 'YES', 'NO', 'NO', 'YES', 'NO', 'YES', 'NO', '
                                    'NO', 'NO', 'NO', 'YES', 'NO', 'YES', 'NO', 'YES', 'YES', 'NO', 'NO', 'NO', 'YES', 'NO', '
                                                                                                                                                                                                                                                                                         'YES', 'YES',
 'NO',
                                    'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO'],
                              dtype=object)
from sklearn.metrics import confusion matrix
confusion matrix(y Test,yp)
array([[70, 2],
                                   [ 2, 6]])
from sklearn.metrics import accuracy score
accuracy score(y Test,yp)
0.95
```

```
confusion matrix(y Train,yp1)
array([[271,
                                                                                  0],
                                       [0, 49]]
accuracy_score(y_Train,yp1)
1.0
clf2=tree.DecisionTreeClassifier(criterion="entropy")
clf2=clf2.fit(X Test,y Test)
 tree.plot tree(clf2)
 [Text(0.4, 0.83333333333333334, 'X[1] \le 114.0 \neq 0.469
 nsamples = 80 \setminus nvalue = [72, 8]'),
     Text(0.2, 0.5, 'entropy = 0.0 \times = 69 \times = 69 \times = [69, 0]'),
     Text(0.6, 0.5, 'X[5] \le 9.135 \setminus entropy = 0.845 \setminus entropy = 11 \setminus 
  [3, 8]'),
     [3, 0]'),
     [0, 8]')]
                                                                                     X[1] \le 114.0
                                                                                entropy = 0.469
                                                                                       samples = 80
                                                                                    value = [72, 8]
                                                                                                                                              X[5] <= 9.135
                              entropy = 0.0
                                                                                                                                         entropy = 0.845
                              samples = 69
                                                                                                                                                 samples = 11
                            value = [69. 0]
                                                                                                                                                 value = [3, 8]
                                                                                        entropy = 0.0
                                                                                                                                                                                                         entropy = 0.0
                                                                                                                                                                                                            samples = 8
                                                                                           samples = 3
                                                                                             /alue = [3, 0]
                                                                                                                                                                                                         value = [0, 8]
clf3=tree.DecisionTreeClassifier(criterion="entropy")
 clf3=clf3.fit(X Train,y Train)
tree.plot tree(clf3)
 0.618 \times = 320 \times = [271, 49]'
     Text(0.15384615384615385, 0.75, 'X[5] \le 9.155 \setminus entropy = 0.036 
nsamples = 266 \setminus nvalue = [265, 1]'),
     Text(0.07692307692307693, 0.5833333333333334, 'entropy = 0.0 \nsamples
= 258 \text{ nvalue} = [258, 0]'),
```

```
Text(0.23076923076923078, 0.5833333333333334, 'X[5] \le 9.165 \setminus Particle (0.23076923076923078)
= 0.544 \setminus samples = 8 \setminus samples = [7, 1]'),
    Text(0.15384615384615385, 0.416666666666667, 'X[2] <= 4.5 \nentropy =
1.0 \rangle = 2 \rangle = [1, 1]'
    Text(0.07692307692307693, 0.25, 'entropy = 0.0 \nsamples = 1 \nvalue = 0.0 \nsamples = 0.0 \nsamples = 1 \nvalue = 0.0 \nsamples = 0.0 \nsamples
 [1, 0]'),
   Text(0.23076923076923078, 0.25, 'entropy = 0.0 \nsamples = 1 \nvalue = 0.0 \nsamples = 0.0 \nsamples = 1 \nvalue = 0.0 \nsamples = 0.0 \nsamples
 [0, 1]'),
    = 6 \ln e = [6, 0]'),
    Text(0.6923076923076923, 0.75, 'X[0] \le 329.5 \neq 0.503
nsamples = 54 \setminus nvalue = [6, 48]'),
    Text(0.5384615384615384, 0.5833333333333333, 'X[0] \le 322.5 \nentropv
= 0.994 \setminus samples = 11 \setminus samples = [5, 6]'),
    Text(0.46153846153846156, 0.416666666666667, 'entropy = 0.0 \nsamples
= 4 \ln e = [0, 4]'
    Text(0.6153846153846154, 0.4166666666666667, 'X[5] <= 9.36 \ nentropy =
0.863 \times = 7 \times = [5, 2]'
    Text(0.5384615384615384, 0.25, 'entropy = 0.0 \nsamples = 4 \nvalue = 0.0 \nsamples = 4 \nvalue = 0.0 \nsamples = 0.0 \nsamp
 [4, 0]'),
    Text(0.6923076923076923, 0.25, 'X[0] \le 328.5 \le 0.918
nsamples = 3 \setminus nvalue = [1, 2]'),
    Text(0.6153846153846154, 0.0833333333333333, 'entropy = 0.0 \nsamples
= 2 \ln u = [0, 2]'),
    Text(0.7692307692307693, 0.0833333333333333, 'entropy = 0.0 \nsamples
= 1 \setminus nvalue = [1, 0]'),
    Text(0.8461538461538461, 0.5833333333333334, 'X[6] <= 0.5 \nentropy =
0.159 \times = 43 \times = [1, 42]'
    = 1 \setminus nvalue = [1, 0]'),
    = 42 \setminus nvalue = [0, 42]')
```

```
X[5] <= 9.21
entropy = 0.618
samples = 320
value = [271, 49]
                                                                                                                           X[0] <= 329.5
entrapy = 0.503
                X[5] <= 9.155
entropy = 0.036
              samples = 266
value = [265, 1]
                                                                                                                            samples = 54
value = [6, 45]
                                                                                                                                                           X[6] <= 0.5
strong = 0.159
                                                                                             X[0] <= 322.5
intropy = 0.994
entropy = 0.0
samples = 258
                                 samples = 5
                                                                                              samples = 11
                                                                                                                                                           samples = 43
value = [256, 0]
                                value = [7, 1]
                                                                                              value = [5, 6]
                                                                                                                                                          value = (1.421)
                                             entropy = 0.0
samples = 6
value = [6, 0]
                                                                             entropy = 0.0
samples = 4
value = [0, 4]
                                                                                                                                           entropy = 0.0
samples = 1
value = [1, 0]
                                                                                                                                                                          entropy = 0.0
                                                                                                                                                                         samples = 42
value = [0, 42]
                 value = [1, 1]
                                                                                                             value = [5, 2]
                                                                                                                          X[0] <= 326.5
entropy = 0.916
 entropy = 0.0
samples = 1
value = [1, 0]
                               entropy = 0.0
samples = 1
value = [0, 1]
                                                                                             entropy = 0.0
                                                                                             samples = 4
value = [4, 0]
                                                                                                                            samples = 3
value = [1, 2]
                                                                                                               intrapy = 0.0
                                                                                                                                             entropy = 0.0
                                                                                                              samples = 2
                                                                                                                                             samples = 1
                                                                                                                                           value = [1, 0]
```

```
yp3=clf2.predict(X Test)
yp3
array(['N0', 'N0', 'N0', 'N0', 'N0', 'N0', 'N0', 'YES', 'YES',
'NO',
       'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'YES',
'NO',
       'NO', 'NO', 'NO', 'NO', 'NO', 'YES', 'NO', 'NO', 'NO',
'NO',
       'NO', 'NO', 'NO', 'NO', 'NO', 'YES', 'NO', 'NO', 'NO',
'NO',
       'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
'NO',
       'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
'NO',
       'NO', 'YES', 'NO', 'NO', 'NO', 'NO', 'NO', 'YES', 'NO',
'NO',
       'NO', 'YES', 'NO'], dtype=object)
yp4=clf3.predict(X Train)
yp4
array(['N0', 'N0', 'N0', 'N0', 'N0', 'N0', 'N0', 'N0', 'N0', 'N0',
'NO',
       'YES', 'NO', 'NO', 'YES', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
'NO',
       'NO', 'NO', 'NO', 'NO', 'YES', 'NO', 'NO', 'NO', 'NO', 'NO',
'NO',
             'NO', 'YES',
                          'YES', 'NO', 'NO', 'NO', 'NO', 'NO', 'YES', 'YES', 'NO', 'NO', 'NO', 'YES', 'NO', 'NO',
       'NO',
             'NO',
                   'YES',
             'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
'NO',
```

```
'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'YES', 'NO', 'NO',
 'YES',
                                        'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'YES', 'NO', 'NO',
 'YES',
                                        'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
 'YES',
                                        'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
 'NO',
                                        'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'YES',
 'NO',
                                         'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
 'NO',
                                        'NO', 'YES', 'NO', 'YES', 'NO', 'YES', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'YES', 'NO', 'NO', 'YES', 'NO',
 'NO',
                                        'YES', 'YES', 'NO', 'NO', 'NO', 'NO', 'YES', 'NO', 'YES', 'NO', 'N
 'NO',
                                        'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
 'NO',
                                        'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
 'NO',
                                        'NO', 'NO', 'NO', 'NO', 'YES', 'NO', 'NO', 'NO', 'NO', 'NO',
 'NO',
                                        'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
 'YES',
                                         'YES', 'YES', 'NO', 'YES', 'NO', 'NO', 'YES', 'NO', 'NO', 'NO',
                                         'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
 'NO',
                                        'YES', 'YES', 'NO', 'NO', 'NO', 'NO', 'NO', 'YES', 'YES', 'YES', 'NO', '
 'NO',
                                        'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
 'NO',
                                        'NO', 'NO', 'NO', 'YES', 'NO', 'YES', 'NO', 'YES', 'NO', 'NO', 'NO', 'YES', 'NO', 'YES', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'YES', 'NO', 'YES', 'NO', 'YES', 'NO', 'YES', 'NO', 'NO
 'NO',
                                        'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO'],
                                 dtvpe=object)
from sklearn.metrics import confusion matrix
confusion matrix(y Test,yp3)
array([[72,
                                                                        0],
                                        [ 0,
                                                                        8]])
confusion matrix(y Train,yp4)
```

```
array([[271, 0], [ 0, 49]])
accuracy_score(y_Test,yp3)
1.0
accuracy score(y Train,yp4)
1.0
import sklearn.tree as tree
import pydotplus
from six import StringIO
from IPython.display import Image
from sklearn.tree import export graphviz
dot data=StringIO()
export_graphviz(clf,out_file=dot_data,filled=True,rounded=True,special
characters=True, class names=['0','1'])
graph=pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write png('admission.png')
Image(graph.create png())
```

