

```
import pandas as pd
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]

```
df=df.drop(df.columns[0],axis=1)
df
```

	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						

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

[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
SOP            0
LOR            0
CGPA           0
Research       0
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 \						
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						

	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	NO
396	0.84	NO

```

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

```

```

      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

```

```

      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']]
X
```

	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]
```

```
y=df[['Admission']]
y
```

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

```
[400 rows x 1 columns]
```

```
df.corr()
```

\	GRE Score	TOEFL Score	University Rating	SOP
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,random_state=0)
```

X_Train

	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA
Research						
3360	319	110	3	3.0	2.5	8.79
640	325	111	3	3.0	3.5	8.70
550	320	103	3	3.0	3.0	7.70
1061	329	111	4	4.5	4.5	9.18
3000	309	106	2	2.5	2.5	8.00
...
3230	305	102	2	2.0	2.5	8.18
1921	322	114	5	4.5	4.0	8.94

1170	290	104	4	2.0	2.5	7.46
470	339	119	5	4.5	4.0	9.70
1721	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

	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA
Research						
1320	309	105	5	3.5	3.5	8.56
3090	308	110	4	3.5	3.0	8.60
3411	326	110	3	3.5	3.5	8.76
1960	306	105	2	3.0	2.5	8.26
2460	316	105	3	3.0	3.5	8.73
...
...						
141	311	104	3	3.5	2.0	8.20
3630	306	103	2	2.5	3.0	8.36
3040	313	106	2	2.5	2.0	8.43
3611	334	116	4	4.0	3.5	9.54
329	297	96	2	2.5	1.5	7.89

0

[80 rows x 7 columns]

y_Test

	Admission
132	NO
309	NO
341	NO
196	NO
246	NO
..	...
14	NO
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] <= 9.21\ngini =
0.259\nsamples = 320\nvalue = [271, 49]'),
Text(0.15384615384615385, 0.75, 'X[5] <= 9.155\ngini = 0.007\nsamples
= 266\nvalue = [265, 1]'),
Text(0.07692307692307693, 0.5833333333333333, 'gini = 0.0\nsamples =
258\nvalue = [258, 0]'),
Text(0.23076923076923078, 0.5833333333333333, 'X[0] <= 324.5\ngini =
0.219\nsamples = 8\nvalue = [7, 1]'),
Text(0.15384615384615385, 0.4166666666666667, 'X[1] <= 110.5\ngini =
0.5\nsamples = 2\nvalue = [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,
1]'),
Text(0.3076923076923077, 0.4166666666666667, 'gini = 0.0\nsamples =
6\nvalue = [6, 0]'),
Text(0.6923076923076923, 0.75, 'X[0] <= 329.5\ngini = 0.198\nsamples
= 54\nvalue = [6, 48]'),
Text(0.5384615384615384, 0.5833333333333333, 'X[0] <= 322.5\ngini =
0.496\nsamples = 11\nvalue = [5, 6]'),
Text(0.46153846153846156, 0.4166666666666667, 'gini = 0.0\nsamples =
4\nvalue = [0, 4]'),
Text(0.6153846153846154, 0.4166666666666667, 'X[5] <= 9.36\ngini =
0.408\nsamples = 7\nvalue = [5, 2]'),
```



```

Text(0.5384615384615384, 0.25, 'gini = 0.0\nsamples = 4\nvalue = [4,
0]'),
Text(0.6923076923076923, 0.25, 'X[0] <= 328.5\ngini = 0.444\nsamples
= 3\nvalue = [1, 2]'),
Text(0.6153846153846154, 0.08333333333333333, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
Text(0.7692307692307693, 0.08333333333333333, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
Text(0.8461538461538461, 0.5833333333333334, 'X[6] <= 0.5\ngini =
0.045\nsamples = 43\nvalue = [1, 42]'),
Text(0.7692307692307693, 0.4166666666666667, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
Text(0.9230769230769231, 0.4166666666666667, 'gini = 0.0\nsamples =
42\nvalue = [0, 42]')]

```



```

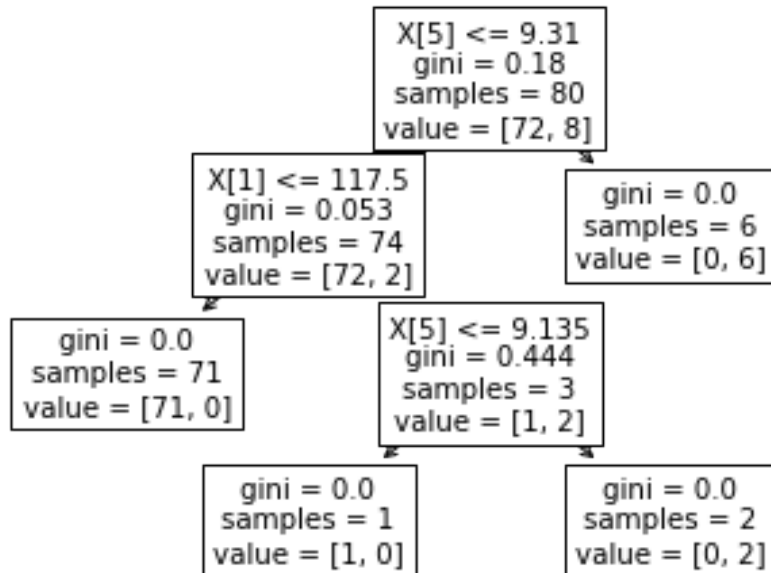
clf1=tree.DecisionTreeClassifier()
clf1=clf1.fit(X_Test,y_Test)
tree.plot_tree(clf1)

```

```

[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]')]

```



```

yp=clf.predict(X_Test)
yp
array(['NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', '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', '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', 'YES', 'NO', 'YES', 'NO',
'NO', 'NO', 'YES', 'NO'], dtype=object)

```

```

yp1=clf.predict(X_Train)
yp1
array(['NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
'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', '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', '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', 'NO', 'YES', 'NO', 'YES',
'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', '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', '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', 'YES', 'NO', 'NO', 'NO', 'NO', 'NO',
'NO', 'NO', 'NO', 'NO', 'YES', 'NO', 'YES', 'NO', 'YES', 'YES',
'NO', 'NO', 'NO', 'YES', 'NO', 'NO', 'YES', 'NO', 'NO', 'NO',
'NO',
'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)
```

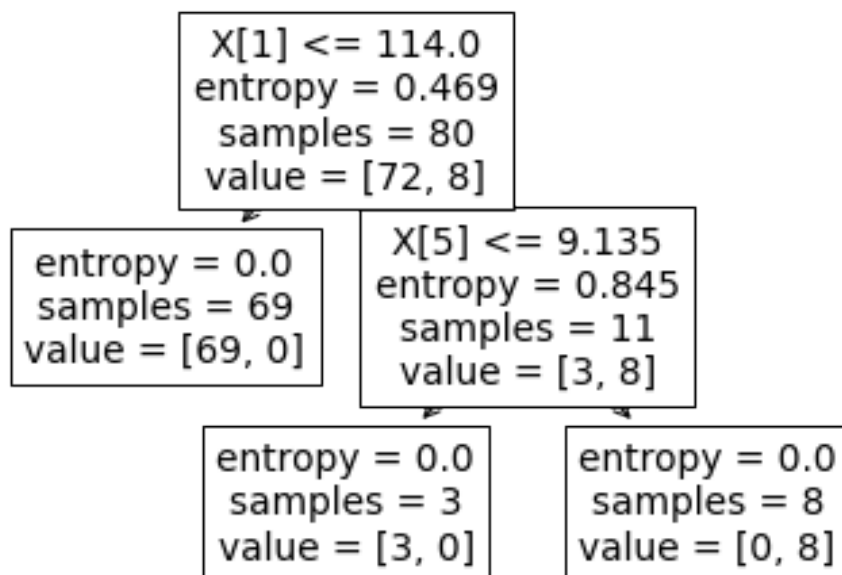
```
1.0
```

```
clf2=tree.DecisionTreeClassifier(criterion="entropy")
```

```
clf2=clf2.fit(X_Test,y_Test)
```

```
tree.plot_tree(clf2)
```

```
[Text(0.4, 0.8333333333333334, 'X[1] <= 114.0\nentropy = 0.469\  
nsamples = 80\nvalue = [72, 8]'),  
 Text(0.2, 0.5, 'entropy = 0.0\nsamples = 69\nvalue = [69, 0]'),  
 Text(0.6, 0.5, 'X[5] <= 9.135\nentropy = 0.845\nsamples = 11\nvalue = [3, 8]'),  
 Text(0.4, 0.16666666666666666, 'entropy = 0.0\nsamples = 3\nvalue = [3, 0]'),  
 Text(0.8, 0.16666666666666666, 'entropy = 0.0\nsamples = 8\nvalue = [0, 8]')]
```



```
clf3=tree.DecisionTreeClassifier(criterion="entropy")
```

```
clf3=clf3.fit(X_Train,y_Train)
```

```
tree.plot_tree(clf3)
```

```
[Text(0.4230769230769231, 0.9166666666666666, 'X[5] <= 9.21\nentropy = 0.618\nsamples = 320\nvalue = [271, 49]'),  
 Text(0.15384615384615385, 0.75, 'X[5] <= 9.155\nentropy = 0.036\  
nsamples = 266\nvalue = [265, 1]'),  
 Text(0.07692307692307693, 0.5833333333333334, 'entropy = 0.0\nsamples = 258\nvalue = [258, 0]')]
```

```
Text(0.23076923076923078, 0.5833333333333334, 'X[5] <= 9.165\nentropy
= 0.544\nsamples = 8\nvalue = [7, 1]'),
Text(0.15384615384615385, 0.4166666666666667, 'X[2] <= 4.5\nentropy =
1.0\nsamples = 2\nvalue = [1, 1]'),
Text(0.07692307692307693, 0.25, 'entropy = 0.0\nsamples = 1\nvalue =
[1, 0]'),
Text(0.23076923076923078, 0.25, 'entropy = 0.0\nsamples = 1\nvalue =
[0, 1]'),
Text(0.3076923076923077, 0.4166666666666667, 'entropy = 0.0\nsamples
= 6\nvalue = [6, 0]'),
Text(0.6923076923076923, 0.75, 'X[0] <= 329.5\nentropy = 0.503\
nsamples = 54\nvalue = [6, 48]'),
Text(0.5384615384615384, 0.5833333333333334, 'X[0] <= 322.5\nentropy
= 0.994\nsamples = 11\nvalue = [5, 6]'),
Text(0.46153846153846156, 0.4166666666666667, 'entropy = 0.0\nsamples
= 4\nvalue = [0, 4]'),
Text(0.6153846153846154, 0.4166666666666667, 'X[5] <= 9.36\nentropy =
0.863\nsamples = 7\nvalue = [5, 2]'),
Text(0.5384615384615384, 0.25, 'entropy = 0.0\nsamples = 4\nvalue =
[4, 0]'),
Text(0.6923076923076923, 0.25, 'X[0] <= 328.5\nentropy = 0.918\
nsamples = 3\nvalue = [1, 2]'),
Text(0.6153846153846154, 0.08333333333333333, 'entropy = 0.0\nsamples
= 2\nvalue = [0, 2]'),
Text(0.7692307692307693, 0.08333333333333333, 'entropy = 0.0\nsamples
= 1\nvalue = [1, 0]'),
Text(0.8461538461538461, 0.5833333333333334, 'X[6] <= 0.5\nentropy =
0.159\nsamples = 43\nvalue = [1, 42]'),
Text(0.7692307692307693, 0.4166666666666667, 'entropy = 0.0\nsamples
= 1\nvalue = [1, 0]'),
Text(0.9230769230769231, 0.4166666666666667, 'entropy = 0.0\nsamples
= 42\nvalue = [0, 42]')]
```

```
yp4=clf3.predict(X_Train)
yp4
array(['NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
       'NO', 'YES', 'NO', 'NO', 'YES', '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',
       'NO', 'NO', 'YES', 'YES', 'NO', 'NO', 'NO', 'YES', 'NO', 'NO',
       'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
       'NO',
```

```

'YES', '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',
'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', 'NO', 'YES',
'NO', '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', 'NO', 'NO',
'NO', 'NO', 'NO', 'NO', 'YES', 'NO', 'NO', 'NO', 'YES', 'NO',
'NO', 'YES', 'YES', 'NO', 'NO', 'NO', 'NO', 'YES', 'NO', 'YES',
'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',
'NO', 'NO', 'NO', 'NO', 'YES', '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', 'NO', 'NO', 'NO', 'NO', 'NO',
'NO', 'NO', 'NO', 'YES', 'NO', 'YES', 'NO', 'YES', 'NO', 'NO',
'YES', 'NO', 'YES', 'NO', 'YES', 'NO', 'NO', 'NO', 'NO', 'NO',
'NO', 'NO', 'NO', 'NO', 'YES', 'NO', 'YES', 'NO', 'YES', 'YES',
'NO', 'NO', 'NO', 'YES', 'NO', 'NO', 'YES', 'NO', 'NO', 'NO',
'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO'],
dtype=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())
```