pr3-admission-gs

November 2, 2024

[1]:
```
pip install pandas
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages
(2.2.2)
Requirement already satisfied: numpy>=1.22.4 in /usr/local/lib/python3.10/dist-
packages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-
packages (from pandas) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-
packages (from pandas) (2024.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-
packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
```

[2]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

[3]:
```
df = pd.read_csv('Admission_Predict.csv')
```

[4]:
```
df.head(10)
```

[4]:

	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		
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		
5	6	330	115	5	4.5	3.0	9.34		
6	7	321	109	3	3.0	4.0	8.20		
7	8	308	101	2	3.0	4.0	7.90		
8	9	302	102	1	2.0	1.5	8.00		
9	10	323	108	3	3.5	3.0	8.60		

```
1          1               0.76
2          1               0.72
3          1               0.80
4          0               0.65
5          1               0.90
6          1               0.75
7          0               0.68
8          0               0.50
9          0               0.45
```

[5]:
```
df.shape
```

[5]:
```
(400, 9)
```

[17]:
```
df['Chance of Admit '] = [1 if each > 0.75 else 0 for each in df['Chance of␣
 ↪Admit ']]
df.head()
```

[17]:
```
   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

   Chance of Admit
0                1
1                1
2                0
3                1
4                0
```

========== Alternate Method

from sklearn.preprocessing import Binarizer

bi = Binarizer(threshold=0.75) # here we are changing values less than 0.75 to 0 and above 0.75 to 1

df[‘Chance of Admit’] = bi.fit_transform(df[[‘Chance of Admit’]])

df.head()

[18]:
```
#df = df.drop('Serial No.',axis=1)
```

[14]:
```
df.shape
```

[14]:
```
(400, 8)
```

[19]:
```
x = df.drop('Chance of Admit ',axis=1) # dropping the admitted column
y = df['Chance of Admit ']
```

x = df[[‘GRE Score’, ‘TOEFL Score’, ‘University Rating’, ‘SOP’, ‘LOR’, ‘CGPA’, ‘Research’]]

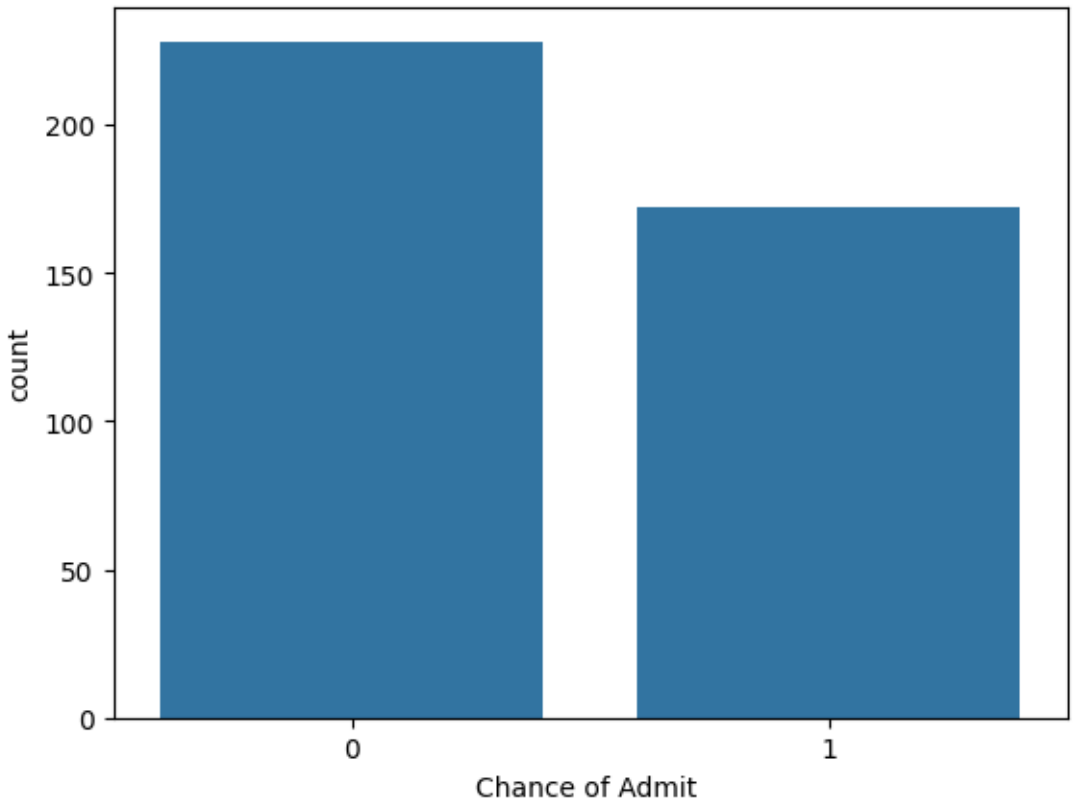
y = df[‘Chance of Admit’]

[21]:
```
#y

#if not int then
#y = y.astype('int')
```

[22]:
```
sns.countplot(x=y)
```

[22]:
```
<Axes: xlabel='Chance of Admit ', ylabel='count'>
```



[23]:
```
y.value_counts()
```

[23]:
```
Chance of Admit
0    228
1    172
Name: count, dtype: int64
```

[24]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.
 ↪25,random_state=0)
```

[25]:
```
x_train.shape
```

[25]:
```
(300, 7)
```

[26]:
```
x_test.shape
```

[26]:
```
(100, 7)
```

[31]:
```
x_test.head()
```

[31]:

	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research
132	309	105	5	3.5	3.5	8.56	0
309	308	110	4	3.5	3.0	8.60	0
341	326	110	3	3.5	3.5	8.76	1
196	306	105	2	3.0	2.5	8.26	0
246	316	105	3	3.0	3.5	8.73	0

[32]:
```
from sklearn.tree import DecisionTreeClassifier
```

[33]:
```
classifier = DecisionTreeClassifier(random_state=0)
```

[34]:
```
classifier.fit(x_train,y_train)
```

[34]:
```
DecisionTreeClassifier(random_state=0)
```

[35]:
```
y_pred = classifier.predict(x_test)
```

[36]:
```
result = pd.DataFrame(
{
    'actual':y_test,
    'predicted':y_pred
})
```

[37]:
```
result
```

[37]:

	actual	predicted
132	0	0
309	0	0
341	1	1
196	0	0
246	0	1
..	…	…
146	0	0

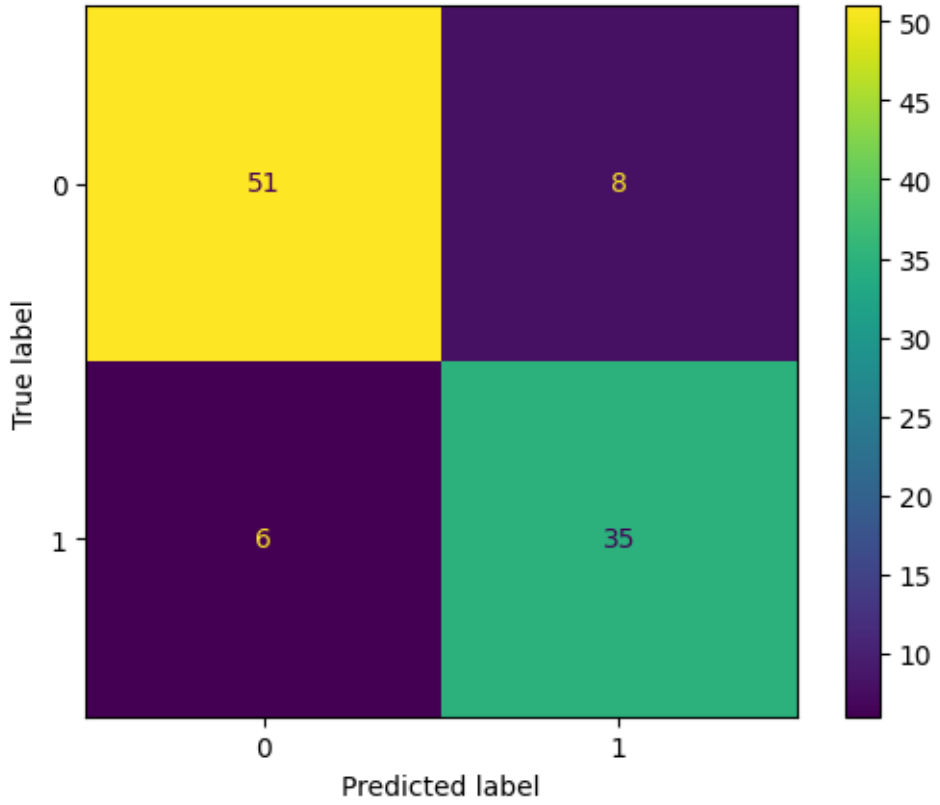
```
135       1          1
390       0          0
264       0          0
364       1          0

[100 rows x 2 columns]
```

[40]:
```
from sklearn.metrics import␣
  ↪confusion_matrix,accuracy_score,ConfusionMatrixDisplay
from sklearn.metrics import classification_report
```

[41]:
```
ConfusionMatrixDisplay.from_predictions(y_test,y_pred)
```

[41]:
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at
0x7c15862bee60>
```



[42]:
```
accuracy_score(y_test,y_pred)
```

[42]:
```
0.86
```

[45]:
```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.89      0.86      0.88        59
           1       0.81      0.85      0.83        41

    accuracy                           0.86       100
   macro avg       0.85      0.86      0.86       100
weighted avg       0.86      0.86      0.86       100
```

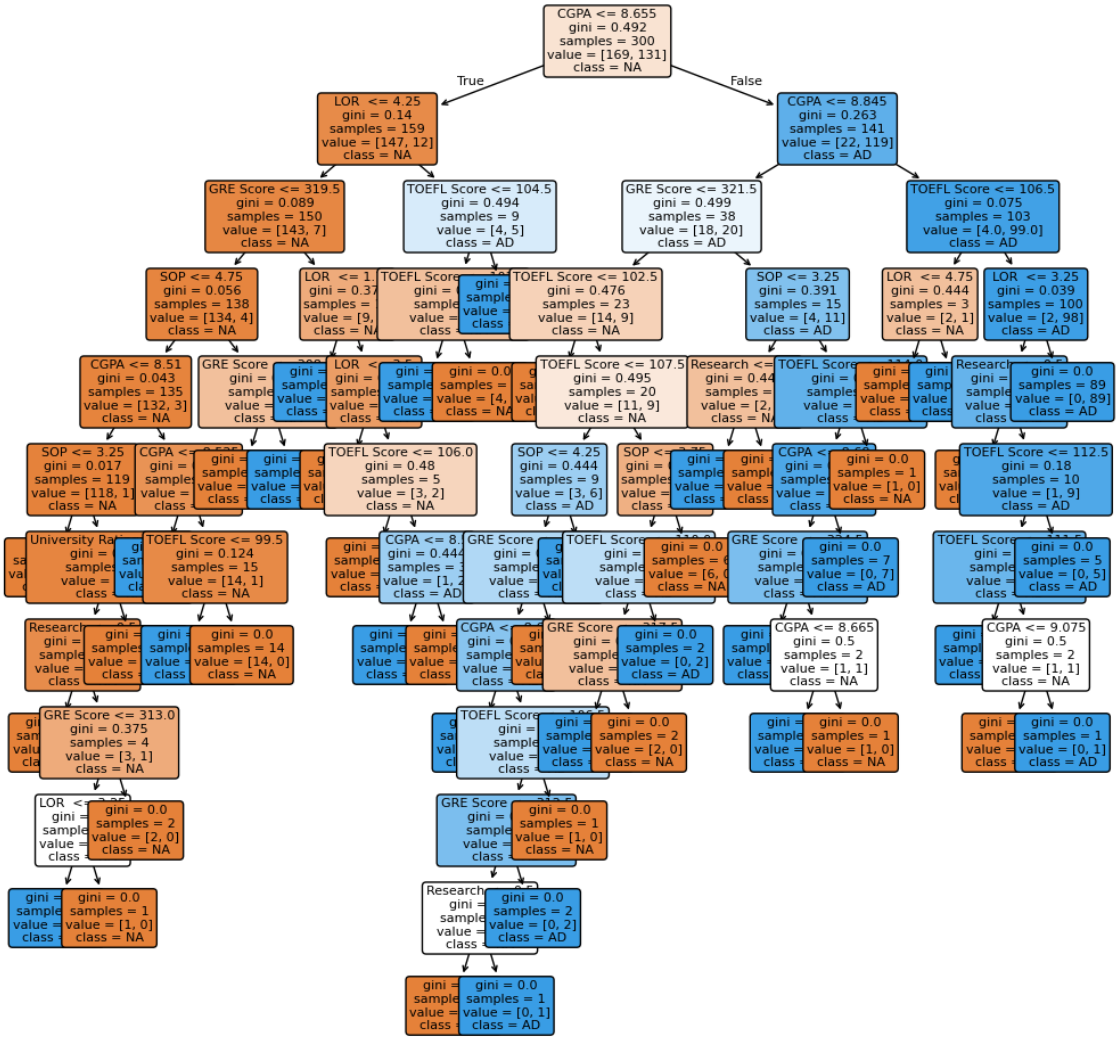
[51]:
```
new=[[322,110,3,3.5,2.5,8.67,1]]
classifier.predict(new)[0]
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:493: UserWarning: X does
not have valid feature names, but DecisionTreeClassifier was fitted with feature
names
  warnings.warn(
```

[51]:
```
1
```

[53]:
```
from sklearn.tree import plot_tree

plt.figure(figsize=(12,12))
plot_tree(classifier,fontsize=8,filled=True,rounded=True,feature_names=x.
 ↪columns,class_names=['NA','AD']);
```


CGPA <= 8.655
gini = 0.492
samples = 300
value = [169, 131]
class = NA
True
False
LOR <= 4.25
gini = 0.14
samples = 159
value = [147, 12]
class = NA
CGPA <= 8.845
gini = 0.263
samples = 141
value = [22, 119]
class = AD
GRE Score <= 319.5
gini = 0.089
samples = 150
value = [143, 7]
class = NA
TOEFL Score <= 104.5
gini = 0.494
samples = 9
value = [4, 5]
class = AD
GRE Score <= 321.5
gini = 0.499
samples = 38
value = [18, 20]
class = AD
TOEFL Score <= 106.5
gini = 0.075
samples = 103
value = [4.0, 99.0]
class = AD
SOP <= 4.75
gini = 0.056
samples = 138
value = [134, 4]
class = NA
TOEFL Score <= 102.5
gini = 0.476
samples = 23
value = [14, 9]
class = NA
SOP <= 3.25
gini = 0.391
samples = 15
value = [4, 11]
class = AD
LOR <= 4.75
gini = 0.444
samples = 3
value = [2, 1]
class = NA
LOR <= 3.25
gini = 0.039
samples = 100
value = [2, 98]
class = AD
CGPA <= 8.51
gini = 0.043
samples = 135
value = [132, 3]
class = NA
TOEFL Score <= 107.5
gini = 0.495
samples = 20
value = [11, 9]
class = NA
gini = 0.0
samples = 89
value = [0, 89]
class = AD
SOP <= 3.25
gini = 0.017
samples = 119
value = [118, 1]
class = NA
TOEFL Score <= 106.0
gini = 0.48
samples = 5
value = [3, 2]
class = NA
SOP <= 4.25
gini = 0.444
samples = 9
value = [3, 6]
class = AD
gini = 0.0
samples = 1
value = [1, 0]
class = NA
TOEFL Score <= 112.5
gini = 0.18
samples = 10
value = [1, 9]
class = AD
TOEFL Score <= 99.5
gini = 0.124
samples = 15
value = [14, 1]
class = NA
gini = 0.0
samples = 14
value = [14, 0]
class = NA
GRE Score <= 313.0
gini = 0.375
samples = 4
value = [3, 1]
class = NA
gini = 0.0
samples = 2
value = [2, 0]
class = NA
gini = 0.0
samples = 1
value = [1, 0]
class = NA
gini = 0.0
samples = 2
value = [0, 2]
class = AD
gini = 0.0
samples = 1
value = [0, 1]
class = AD
gini = 0.0
samples = 7
value = [0, 7]
class = AD
CGPA <= 8.665
gini = 0.5
samples = 2
value = [1, 1]
class = NA
gini = 0.0
samples = 1
value = [1, 0]
class = NA
gini = 0.0
samples = 5
value = [0, 5]
class = AD
CGPA <= 9.075
gini = 0.5
samples = 2
value = [1, 1]
class = NA
gini = 0.0
samples = 1
value = [0, 1]
class = AD

[ ]:

[ ]:

[ ]: