decisiontree

September 24, 2024

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

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

[2]:
```
     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]
```

[3]:
```
df.isnull().sum()
```

```
[3]: 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
```

```
[4]: for i in range(len(df)):
         if df.at[i, "Chance of Admit "] >= 0.75:
             df.at[i, "Chance of Admit "] = 1
         else:
             df.at[i, "Chance of Admit "] = 0
```

```
[5]: df
```

```
[5]:      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               1.0
     1           1               1.0
     2           1               0.0
     3           1               1.0
     4           0               0.0
     ..        …               …
     395         1               1.0
     396         1               1.0
     397         1               1.0
     398         0               0.0
     399         1               1.0

     [400 rows x 9 columns]
```

[6]:

```
df.columns
```

[6]:

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

[7]:

```
X=df[['GRE Score', 'TOEFL Score', 'University Rating', 'SOP','LOR ', 'CGPA',␣
 ↪'Research']]
Y=df[['Chance of Admit ']]
```

[8]:

```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier , plot_tree
```

[9]:

```
X_train, X_test, Y_train, Y_test = train_test_split(  X, Y, test_size=0.3)
Model=DecisionTreeClassifier(criterion='entropy')
```

[10]:

```
Model.fit(X_train,Y_train)
```

[10]:

```
DecisionTreeClassifier(criterion='entropy')
```

[11]:

```
Y_pred=Model.predict(X_test)
```

[12]:

```
from sklearn.metrics import classification_report
```

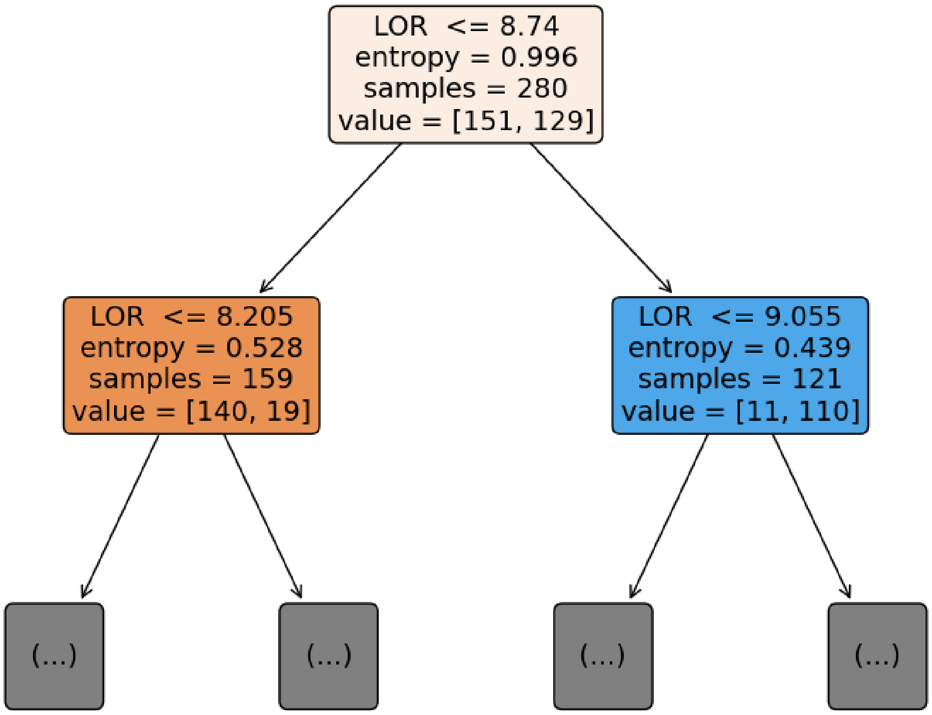
[13]:

```
print('', classification_report(Y_test, Y_pred))
```

```
               precision    recall  f1-score   support

         0.0       0.82      0.86      0.84        69
         1.0       0.79      0.75      0.77        51

    accuracy                           0.81       120
   macro avg       0.81      0.80      0.80       120
weighted avg       0.81      0.81      0.81       120
```

[14]:

```
plt.figure(figsize=(10, 8))
plot_tree(Model, filled=True, feature_names=df.columns,max_depth=1,␣
 ↪rounded=True)

plt.show()
```


LOR <= 8.74
entropy = 0.996
samples = 280
value = [151, 129]
LOR <= 8.205
entropy = 0.528
samples = 159
value = [140, 19]
LOR <= 9.055
entropy = 0.439
samples = 121
value = [11, 110]
(...)
(...)
(...)
(...)

[ ]: