

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import plot_tree
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_recall_fscore_support
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import ConfusionMatrixDisplay
```

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

```
In [3]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype  
---  -
0   Serial No.            400 non-null   int64   
1   GRE Score              400 non-null   int64   
2   TOEFL Score            400 non-null   int64   
3   University Rating      400 non-null   int64   
4   SOP                    400 non-null   float64  
5   LOR                    400 non-null   float64  
6   CGPA                   400 non-null   float64  
7   Research                400 non-null   int64   
8   Chance of Admit        400 non-null   float64  
dtypes: float64(4), int64(5)
memory usage: 28.2 KB
```

```
In [4]: df.isna().sum()
```

```
Out[4]: 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 [5]: threshold = 0.90
df.loc[df['Chance of Admit '] > threshold, 'Chance of Admit '] = 1
df.loc[df['Chance of Admit '] <= threshold, 'Chance of Admit '] = 0
```

```
In [6]: X = df.drop(['Serial No.', 'Chance of Admit '], axis = 1)
y = df['Chance of Admit ']
```

```
In [7]: (train_X, test_X, train_y, test_y) = train_test_split(X, y, random_state=2, test_size=0.2)
```

```
In [8]: model = DecisionTreeClassifier()
model.fit(train_X, train_y)
pred_y = model.predict(test_X)
```

```
In [9]: print('Classification Report =\n',classification_report(test_y, pred_y))
print('Accuracy =', accuracy_score(test_y, pred_y))
```

```
Classification Report =
              precision    recall  f1-score   support

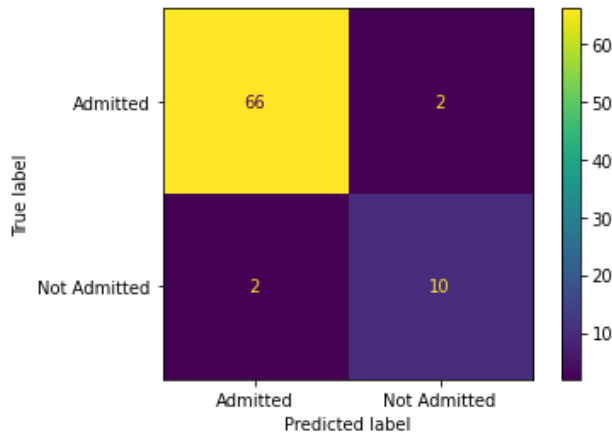
      0.0         0.97      0.97      0.97         68
      1.0         0.83      0.83      0.83         12

 accuracy
macro avg         0.90      0.90      0.90         80
weighted avg         0.95      0.95      0.95         80

Accuracy = 0.95
```

```
In [10]: ConfusionMatrixDisplay(confusion_matrix(test_y, pred_y), display_labels= ['Admitted', 'Not Admitted']).plot()
```

```
Out[10]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7f76865d49b0>
```



```
In [18]: f_name = ['GRE Score', 'TOEFL Score', 'University Rating', 'SOP', 'LOR', 'CGPA', 'Research']
plt.figure(figsize=(30,20))
img = plot_tree(model, filled=True, feature_names=f_name, class_names=[str(x) for x in model.classes_])
```

