```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
df = pd.read_csv('Advertising_data.csv')
```

```
df.shape
```

```
(400, 5)
```

```
df.head()
```

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19.0	19000.0	0
1	15810944	Male	35.0	20000.0	0
2	15668575	Female	26.0	43000.0	0
3	15603246	Female	27.0	57000.0	0
4	15804002	Male	19.0	76000.0	0

```
df['Gender'].replace(['Male', 'Female'],
                        [0, 1], inplace = True)
```

```
df.Purchased.unique()
```

```
array([0, 1])
```

```
X = df.loc[:, ['Gender' , 'Age', 'EstimatedSalary']]
```

```
y = df.loc[:, 'Purchased']
```

```
y
```

```
0      0
1      0
2      0
3      0
4      0
      ..
395    1
396    1
397    1
398    0
399    1
Name: Purchased, Length: 400, dtype: int64
```

```
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 = 5)
```

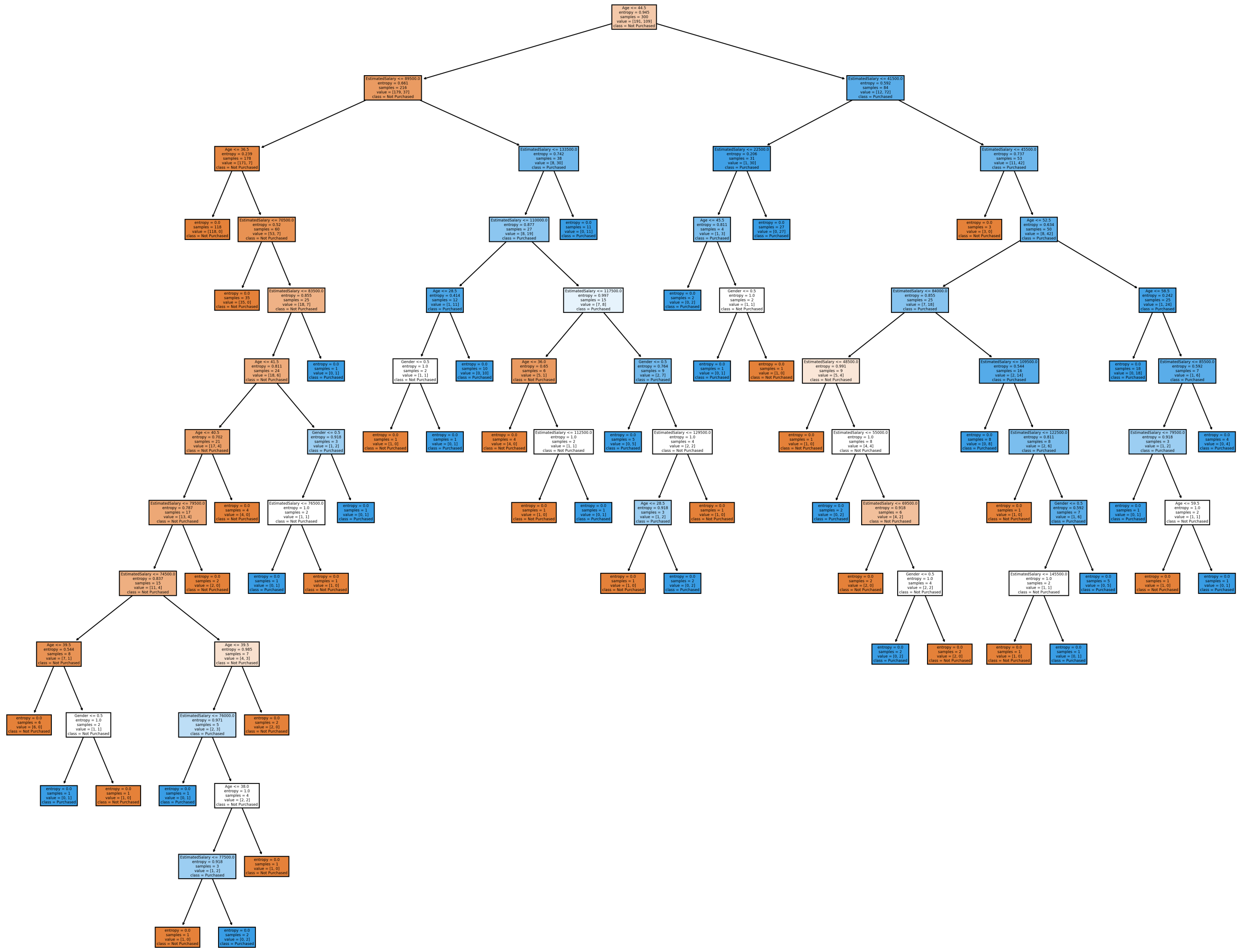
```
from sklearn.tree import DecisionTreeClassifier
dtc = DecisionTreeClassifier(criterion = 'entropy' ,
                             random_state = 0)
dtc.fit(X_train , y_train)
```

```
▾                    DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', random_state=0)
```

```
from sklearn import tree

plt.figure(figsize = ( 25 , 20) , dpi = 300.0)

f1= tree.plot_tree(dtc,
                   feature_names = ['Gender' , 'Age' , 'EstimatedSalary'],
                   class_names = ['Not Purchased', 'Purchased'] ,
                    filled = True )
```


Age <= 44.5 entropy = 0.945 samples = 300 value = [191, 109] class = Not Purchased
EstimatedSalary <= 89500.0 entropy = 0.661 samples = 216 value = [179, 37] class = Not Purchased
EstimatedSalary <= 41500.0 entropy = 0.592 samples = 84 value = [12, 72] class = Purchased
Age <= 36.5 entropy = 0.239 samples = 178 value = [171, 7] class = Not Purchased
EstimatedSalary <= 133500.0 entropy = 0.742 samples = 38 value = [8, 30] class = Purchased
EstimatedSalary <= 22500.0 entropy = 0.206 samples = 31 value = [1, 30] class = Purchased
EstimatedSalary <= 45500.0 entropy = 0.737 samples = 53 value = [11, 42] class = Purchased
entropy = 0.0 samples = 118 value = [118, 0] class = Not Purchased
EstimatedSalary <= 70500.0 entropy = 0.52 samples = 60 value = [53, 7] class = Not Purchased
EstimatedSalary <= 110000.0 entropy = 0.877 samples = 27 value = [8, 19] class = Purchased
entropy = 0.0 samples = 11 value = [0, 11] class = Purchased
Age <= 45.5 entropy = 0.811 samples = 4 value = [1, 3] class = Purchased
entropy = 0.0 samples = 27 value = [0, 27] class = Purchased
entropy = 0.0 samples = 3 value = [3, 0] class = Not Purchased
Age <= 52.5 entropy = 0.634 samples = 50 value = [8, 42] class = Purchased
entropy = 0.0 samples = 35 value = [35, 0] class = Not Purchased
EstimatedSalary <= 83500.0 entropy = 0.855 samples = 25 value = [18, 7] class = Not Purchased
Age <= 28.5 entropy = 0.414 samples = 12 value = [1, 11] class = Purchased
EstimatedSalary <= 117500.0 entropy = 0.997 samples = 15 value = [7, 8] class = Purchased
entropy = 0.0 samples = 2 value = [0, 2] class = Purchased
Gender <= 0.5 entropy = 1.0 samples = 2 value = [1, 1] class = Not Purchased
EstimatedSalary <= 84000.0 entropy = 0.855 samples = 25 value = [7, 18] class = Purchased
Age <= 58.5 entropy = 0.242 samples = 25 value = [1, 24] class = Purchased
Age <= 41.5 entropy = 0.811 samples = 24 value = [18, 6] class = Not Purchased
entropy = 0.0 samples = 1 value = [0, 1] class = Purchased
Gender <= 0.5 entropy = 1.0 samples = 2 value = [1, 1] class = Not Purchased
entropy = 0.0 samples = 10 value = [0, 10] class = Purchased
Age <= 36.0 entropy = 0.65 samples = 6 value = [5, 1] class = Not Purchased
Gender <= 0.5 entropy = 0.764 samples = 9 value = [2, 7] class = Purchased
entropy = 0.0 samples = 1 value = [0, 1] class = Purchased
entropy = 0.0 samples = 1 value = [1, 0] class = Not Purchased
EstimatedSalary <= 48500.0 entropy = 0.991 samples = 9 value = [5, 4] class = Not Purchased
EstimatedSalary <= 109500.0 entropy = 0.544 samples = 16 value = [2, 14] class = Purchased
entropy = 0.0 samples = 18 value = [0, 18] class = Purchased
EstimatedSalary <= 85500.0 entropy = 0.592 samples = 7 value = [1, 6] class = Purchased
Age <= 40.5 entropy = 0.702 samples = 21 value = [17, 4] class = Not Purchased
Gender <= 0.5 entropy = 0.918 samples = 3 value = [1, 2] class = Purchased
entropy = 0.0 samples = 1 value = [1, 0] class = Not Purchased
entropy = 0.0 samples = 1 value = [0, 1] class = Purchased
entropy = 0.0 samples = 4 value = [4, 0] class = Not Purchased
EstimatedSalary <= 112500.0 entropy = 1.0 samples = 2 value = [1, 1] class = Not Purchased
entropy = 0.0 samples = 5 value = [0, 5] class = Purchased
EstimatedSalary <= 129500.0 entropy = 1.0 samples = 4 value = [2, 2] class = Not Purchased
entropy = 0.0 samples = 1 value = [1, 0] class = Not Purchased
EstimatedSalary <= 55000.0 entropy = 1.0 samples = 8 value = [4, 4] class = Not Purchased
entropy = 0.0 samples = 8 value = [0, 8] class = Purchased
EstimatedSalary <= 122500.0 entropy = 0.811 samples = 8 value = [2, 6] class = Purchased
EstimatedSalary <= 79500.0 entropy = 0.918 samples = 3 value = [1, 2] class = Purchased
entropy = 0.0 samples = 4 value = [0, 4] class = Purchased
EstimatedSalary <= 79500.0 entropy = 0.787 samples = 17 value = [13, 4] class = Not Purchased
entropy = 0.0 samples = 4 value = [4, 0] class = Not Purchased
EstimatedSalary <= 76500.0 entropy = 1.0 samples = 2 value = [1, 1] class = Not Purchased
entropy = 0.0 samples = 1 value = [0, 1] class = Purchased
entropy = 0.0 samples = 1 value = [1, 0] class = Not Purchased
entropy = 0.0 samples = 1 value = [0, 1] class = Purchased
Age <= 28.5 entropy = 0.918 samples = 3 value = [1, 2] class = Purchased
entropy = 0.0 samples = 1 value = [1, 0] class = Not Purchased
entropy = 0.0 samples = 2 value = [0, 2] class = Purchased
EstimatedSalary <= 69500.0 entropy = 0.918 samples = 6 value = [4, 2] class = Not Purchased
entropy = 0.0 samples = 1 value = [1, 0] class = Not Purchased
Gender <= 0.5 entropy = 0.592 samples = 7 value = [1, 6] class = Purchased
entropy = 0.0 samples = 1 value = [0, 1] class = Purchased
Age <= 59.5 entropy = 1.0 samples = 2 value = [1, 1] class = Not Purchased
EstimatedSalary <= 74500.0 entropy = 0.837 samples = 15 value = [11, 4] class = Not Purchased
entropy = 0.0 samples = 2 value = [2, 0] class = Not Purchased
entropy = 0.0 samples = 1 value = [0, 1] class = Purchased
entropy = 0.0 samples = 1 value = [1, 0] class = Not Purchased
entropy = 0.0 samples = 1 value = [1, 0] class = Not Purchased
entropy = 0.0 samples = 2 value = [0, 2] class = Purchased
entropy = 0.0 samples = 2 value = [2, 0] class = Not Purchased
Gender <= 0.5 entropy = 1.0 samples = 4 value = [2, 2] class = Not Purchased
EstimatedSalary <= 145500.0 entropy = 1.0 samples = 2 value = [1, 1] class = Not Purchased
entropy = 0.0 samples = 5 value = [0, 5] class = Purchased
entropy = 0.0 samples = 1 value = [1, 0] class = Not Purchased
entropy = 0.0 samples = 1 value = [0, 1] class = Purchased
Age <= 39.5 entropy = 0.544 samples = 8 value = [7, 1] class = Not Purchased
Age <= 39.5 entropy = 0.985 samples = 7 value = [4, 3] class = Not Purchased
entropy = 0.0 samples = 2 value = [0, 2] class = Purchased
entropy = 0.0 samples = 2 value = [2, 0] class = Not Purchased
entropy = 0.0 samples = 1 value = [1, 0] class = Not Purchased
entropy = 0.0 samples = 1 value = [0, 1] class = Purchased
entropy = 0.0 samples = 6 value = [6, 0] class = Not Purchased
Gender <= 0.5 entropy = 1.0 samples = 2 value = [1, 1] class = Not Purchased
EstimatedSalary <= 76000.0 entropy = 0.971 samples = 5 value = [2, 3] class = Purchased
entropy = 0.0 samples = 2 value = [2, 0] class = Not Purchased
entropy = 0.0 samples = 1 value = [0, 1] class = Purchased
entropy = 0.0 samples = 1 value = [1, 0] class = Not Purchased
entropy = 0.0 samples = 1 value = [0, 1] class = Purchased
Age <= 38.0 entropy = 1.0 samples = 4 value = [2, 2] class = Not Purchased
EstimatedSalary <= 77500.0 entropy = 0.918 samples = 3 value = [1, 2] class = Purchased
entropy = 0.0 samples = 1 value = [1, 0] class = Not Purchased
entropy = 0.0 samples = 1 value = [1, 0] class = Not Purchased
entropy = 0.0 samples = 2 value = [0, 2] class = Purchased

```
print ( 'Depth of the tree: ' , dtc.get_depth() )
print ( 'No. of leaves in the tree:' , dtc.get_n_leaves() )
```

```
    Depth of the tree:  13
    No. of leaves in the tree: 47
```

```
from sklearn.metrics import accuracy_score
pred_train = dtc.predict(X_train)
accuracy_train = accuracy_score(y_train, pred_train)
print('% of Accuracy of train data: ', accuracy_train * 100)
pred_test = dtc.predict(X_test)
accuracy_test = accuracy_score(y_test, pred_test)
print('% of Accuracy on test data: ', accuracy_test * 100)
```

```
    % of Accuracy of train data:  100.0
    % of Accuracy on test data:  80.0
```

```
from sklearn.model_selection import GridSearchCV
params = {'criterion': ['entropy', 'gini'],
          'max_depth': range(1, 10)}

gsc = GridSearchCV(estimator = dtc,param_grid = params,scoring = 'accuracy',cv = 10,n_jobs = -1)
gsc = gsc.fit(X_train, y_train)
```

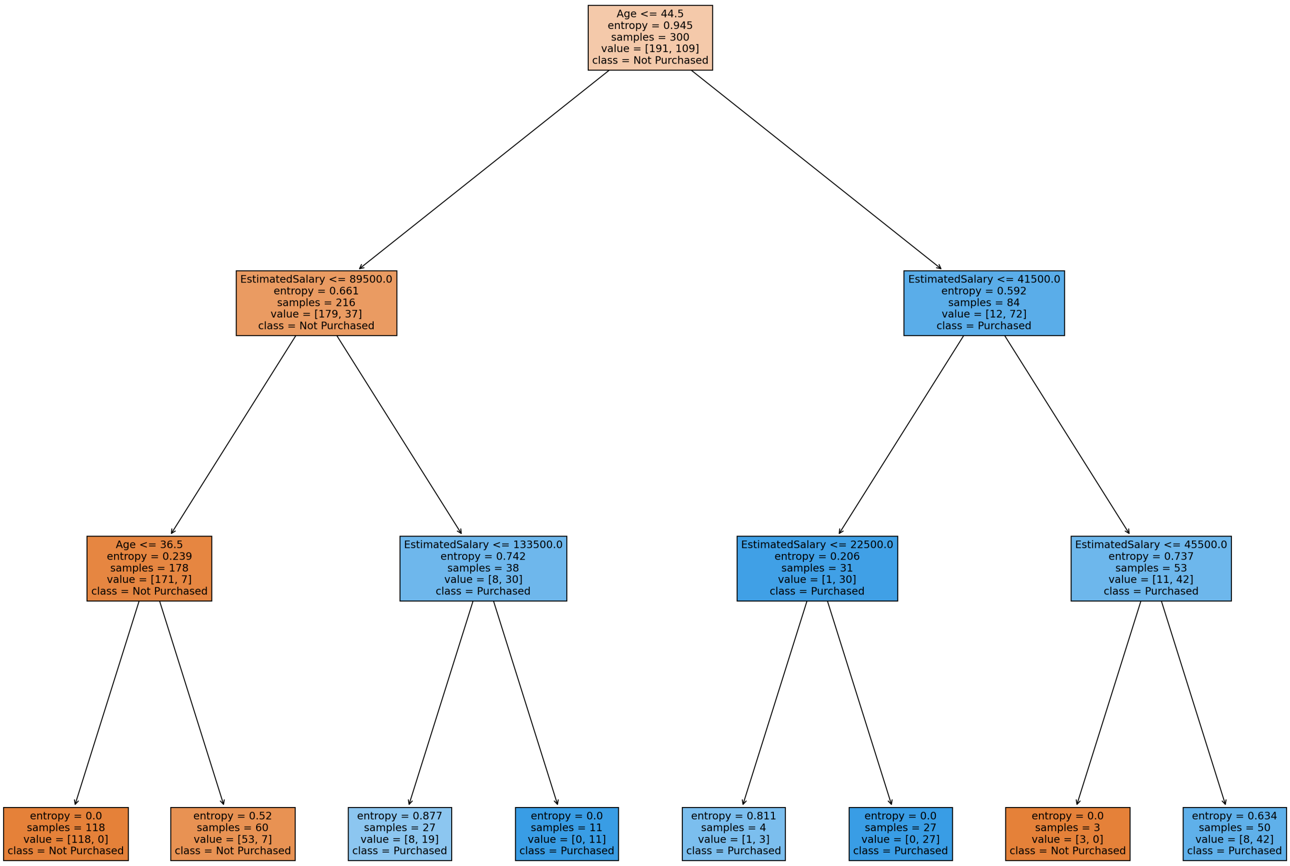
```
gsc.best_estimator_
```

```
▾                        DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', max_depth=3, random_state=0)
```

```
dtc1 = DecisionTreeClassifier(criterion = 'entropy' , max_depth=3, random_state = 0)
dtc1.fit(X_train , y_train)
```

```
▾                        DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', max_depth=3, random_state=0)
```

```
plt.figure(figsize = ( 25 , 20) , dpi = 200.0)
f2= tree.plot_tree(dtc1,feature_names = ['Gender' , 'Age' , 'EstimatedSalary'],class_names = ['Not Purchased', 'Purchased'] ,filled = Tr
```



```
print ( 'Depth of the tree after handling overfitting:' ,dtc1.get_depth() )

print ( 'No. of leaves in the tree after handling overfitting:',dtc1.get_n_leaves() )
```

```
Depth of the tree after handling overfitting: 3
No. of leaves in the tree after handling overfitting: 8
```

```
pred_train = dtc.predict(X_train)
acc_train = accuracy_score(y_train, pred_train)
print('% of Accuracy of train data after handling overfitting: ', acc_train * 100)
pred_test = dtc1.predict(X_test)
acc_test = accuracy_score(y_test, pred_test)
print('% of Accuracy of the tree on test data after handling overfittin : ', accuracy_test * 100)
```

```
     % of Accuracy of train data after handling overfitting:  100.0
     % of Accuracy of the tree on test data after handling overfittin :  90.0
```