# DECISION TREE CLASSIFICATION IMPLEMENTATION

## CODE

```
# Assigning features and label variables
weather=['Sunny','Sunny','Overcast','Rainy','Rainy','Rainy','Overcast','Sunny','Sunny',
'Rainy','Sunny','Overcast','Overcast','Rainy']
temp=['Hot','Hot','Hot','Mild','Cool','Cool','Cool','Mild','Cool','Mild','Mild','Mild','Hot','Mild']

play=['No','No','Yes','Yes','Yes','No','Yes','No','Yes','Yes','Yes','Yes','Yes','No']

# Import LabelEncoder
from sklearn import preprocessing
#creating labelEncoder
le = preprocessing.LabelEncoder()
# Converting string labels into numbers.
weather_encoded=le.fit_transform(weather)
temp_encoded=le.fit_transform(temp)
target=le.fit_transform(play)

print(weather_encoded)
print(temp_encoded)
print(target)

import numpy as np
zipped=zip(weather_encoded,temp_encoded)
features = np.array(list(zipped)).tolist()
print(features)

from sklearn import tree
#Create a Gaussian Classifier
model = tree.DecisionTreeClassifier(criterion='entropy')

# Train the model using the training sets
model.fit(features,target)

#Predict Output
predicted= model.predict([[0, 2]]) # 0:Overcast, 2:Mild
print("Predicted Value:", predicted)

from matplotlib import pyplot as plt
fig, ax = plt.subplots(figsize=(6, 6)) #figsize value changes the size of plot
tree.plot_tree(model,ax=ax,feature_names=['wether','temp'])
plt.show()
```

## OUTPUT

```
[2 2 0 1 1 1 0 2 2 1 2 0 0 1]
[1 1 1 2 0 0 0 2 0 2 2 2 1 2]
[0 0 1 1 1 0 1 0 1 1 1 1 1 0]
[[2, 1], [2, 1], [0, 1], [1, 2], [1, 0], [1, 0], [0, 0], [2, 2], [2, 0], [1, 2], [2,
Predicted Value: [1]
```