

```
import pandas as pd
import numpy as np
from sklearn import datasets
from sklearn import model_selection
from sklearn import tree
import graphviz

wine = datasets.load_wine()
print('Dataset structure= ', dir(wine))
```

Dataset structure= ['DESCR', 'data', 'feature_names', 'frame', 'target', 'target_names']

```
df = pd.DataFrame(wine.data, columns = wine.feature_names)
df['target'] = wine.target
df['feature_names'] = df.target.apply(lambda x : wine.target_names[x])

print('Unique target values=',df['target'].unique())
```

Unique target values= [0 1 2]

```
df.sample(5)
```

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids
24	13.50	1.81	2.61	20.0	96.0	2.53	2.61
130	12.86	1.35	2.32	18.0	122.0	1.51	1.25
63	12.37	1.13	2.16	19.0	87.0	3.50	3.10
47	13.90	1.68	2.12	16.0	101.0	3.10	3.39
98	12.37	1.07	2.10	18.5	88.0	3.52	3.75



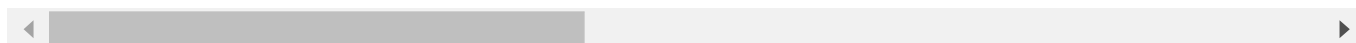
```
# label = class 0
df[df.target == 0].head(3)
```

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	n
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	
1	12.20	1.78	2.14	11.2	100.0	2.65	2.76	

```
# label = class 1
```

```
df[df.target == 1].head(3)
```

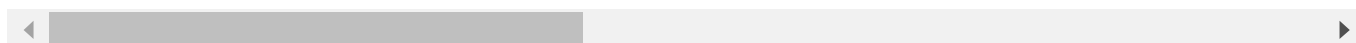
	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids
59	12.37	0.94	1.36	10.6	88.0	1.98	0.57
60	12.33	1.10	2.28	16.0	101.0	2.05	1.09
61	12.64	1.36	2.02	16.8	100.0	2.02	1.41



```
# label = class 2
```

```
df[df.target == 2].head(3)
```

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids
130	12.86	1.35	2.32	18.0	122.0	1.51	1.25
131	12.88	2.99	2.40	20.0	104.0	1.30	1.22
132	12.81	2.31	2.40	24.0	98.0	1.15	1.09



```
X = df[['alcohol','malic_acid']]
```

```
y = df[['target']]
```

```
print('X shape=', X.shape)
```

```
X shape= (178, 2)
```

```
print('Y shape=', y.shape)
```

```
Y shape= (178, 1)
```

```
X_train,X_test, y_train, y_test = model_selection.train_test_split(X, y, test_size= 0.2, rand
```

```
print('X_train dimension= ', X_train.shape)
```

```
print('X_test dimension= ', X_test.shape)
print('y_train dimension= ', y_train.shape)
print('y_train dimension= ', y_test.shape)
```

```
X_train dimension= (142, 2)
X_test dimension= (36, 2)
y_train dimension= (142, 1)
y_test dimension= (36, 1)
```

```
cls = tree.DecisionTreeClassifier(random_state= 1)
cls.fit(X_train ,y_train)
```

```
DecisionTreeClassifier(random_state=1)
```

```
print('Actual value for 10th training example=',wine.target_names[y_test.iloc[10]][0])
print('Predicted value for 10th training example=', wine.target_names[cls.predict([X_test.ilo
```

```
Actual value for 10th training example= class_0
Predicted value for 10th training example= class_0
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not have
    "X does not have valid feature names, but"
```



```
print('\nActual value of species for 20th training example=',wine.target_names[y_test.iloc[20]
print('Predicted value of species for 20th training example=', wine.target_names[cls.predict(
```

```
Actual value of species for 20th training example= class_0
Predicted value of species for 20th training example= class_0
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not have
    "X does not have valid feature names, but"
```



```
print('\nActual value of species for 30th training example=',wine.target_names[y_test.iloc[29]
print('Predicted value of species for 30th training example=', wine.target_names[cls.predict(
```

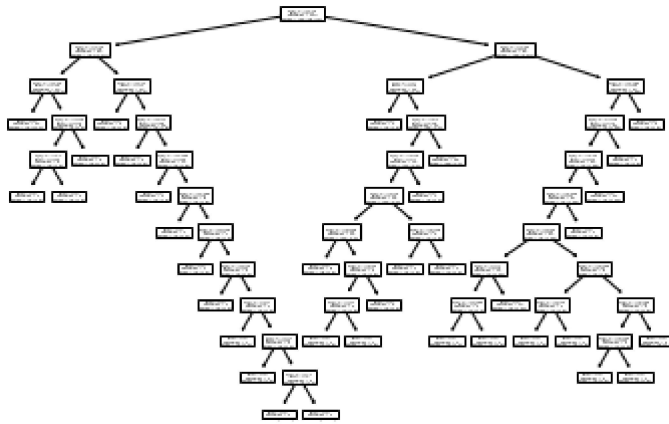
```
Actual value of species for 30th training example= class_2
Predicted value of species for 30th training example= class_0
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not have
    "X does not have valid feature names, but"
```



```
cls.score(X_test, y_test)
```

```
0.7222222222222222
```

```
tree.plot_tree(cls);
```



+ Code

+ Text

```
dot_data = tree.export_graphviz(cls, out_file=None)
graph = graphviz.Source(dot_data)
graph.render("wine_decision_tree")
```

```
'wine_decision_tree.pdf'
```

```
import matplotlib.pyplot as plt
```

```
fn=['alcohol','malic_acid']
cn=['Class 0', 'Class 1', 'Class 2']
fig, axes = plt.subplots(nrows = 1,ncols = 1,figsize = (10,10), dpi=300)
tree.plot_tree(cls,feature_names = fn,class_names=cn,filled = True);
fig.savefig('imagenname.png')
```

