Code for Decision Tree :

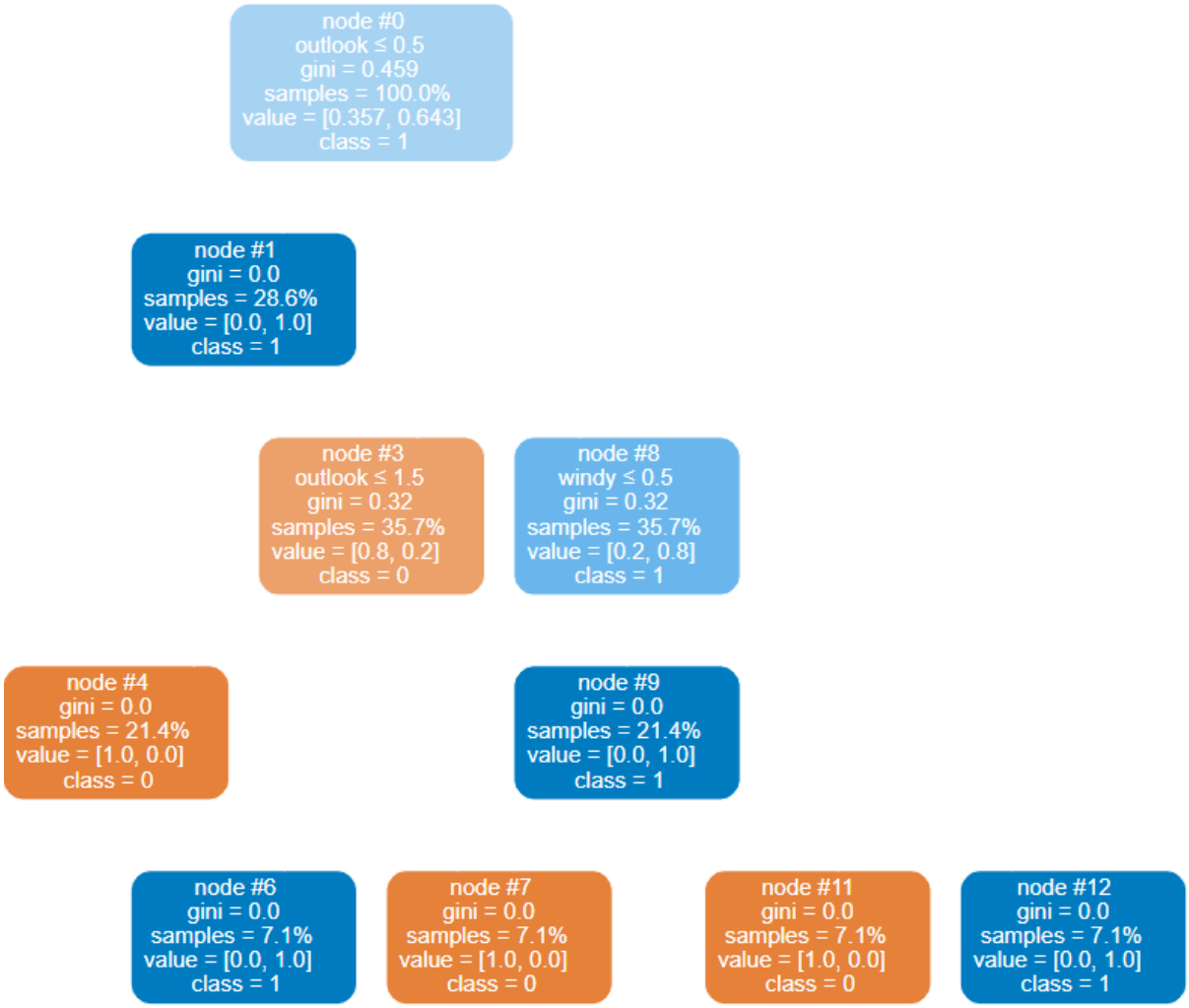
```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier,export_graphviz
from sklearn.model_selection import train_test_split
from sklearn import metrics
import numpy as np
df_golf = pd.read_csv('playgolf.csv')
df_golf
df_golf.info()
from sklearn import preprocessing
def toNumerical(base) -> pd.DataFrame:
    result = base.copy()
    new = preprocessing.LabelEncoder()
    for n in result:
        old = result[n]
        if old.dtype != 'int64' and old.dtype != 'float64':
            new.fit(old)
            result[n] = new.transform(old)
    return result
    base = toNumerical(df_golf)
base.head()
X_train, X_test, y_train, y_test =
train_test_split(base.drop('playgolf',axis=1),base['playgolf'],test_siz
e=0.01,shuffle=False)
X_train.shape,X_test.shape
clf = DecisionTreeClassifier(random_state=0)
clf = clf.fit(X_train,y_train)
for feature, importance in zip(base.columns, clf.feature_importances_):
    print("{}:{}".format(feature, importance))
result = clf.predict(X_test)
print(result)
print(base.values[-1])
from sklearn import metrics
print(metrics.classification_report(y_test,result))
import pydot
import graphviz
dotfile = open("./graph_golf.dot", 'w')
dot_datafile = export_graphviz(
         clf,
         out_file=dotfile,
         feature_names=base.drop('playgolf',axis=1).columns,
         class_names=['0','1'],
         filled=True, rounded=True,
         proportion=True,
         node_ids=True,
         rotate=False,
         label='all',
         special_characters=True
```

```
dotfile.close()
from subprocess import check_call
check_call(['dot','-Tpng','./graph_golf.dot','-o','./graph_golf.png'])

dot_data = export_graphviz(
         clf,
         out_file=None,
         feature_names=base.drop('playgolf',axis=1).columns,
         class_names=['0','1'],
         filled=True, rounded=True,
         proportion=True,
         node_ids=True,
         rotate=False,
         label='all',
         special_characters=True
        )
graph = graphviz.Source(dot_data)
graph
```

	Outlook	Temperature	Humidity	Windy	Play
0	sunny	hot	high	False	no
1	sunny	hot	high	True	no
2	overcast	hot	high	False	yes
3	rainy	mild	high	False	yes
4	rainy	cool	normal	False	yes
5	rainy	cool	normal	True	no
6	overcast	cool	normal	True	yes
7	sunny	mild	high	False	no
8	sunny	cool	normal	False	yes
9	rainy	mild	normal	False	yes
10	sunny	mild	normal	True	yes
11	overcast	mild	high	True	yes
12	overcast	hot	normal	False	yes
13	rainy	mild	high	True	no

Output :


node #0
outlook ≤ 0.5
gini = 0.459
samples = 100.0%
value = [0.357, 0.643]
class = 1
node #1
gini = 0.0
samples = 28.6%
value = [0.0, 1.0]
class = 1
node #3
outlook ≤ 1.5
gini = 0.32
samples = 35.7%
value = [0.8, 0.2]
class = 0
node #8
windy ≤ 0.5
gini = 0.32
samples = 35.7%
value = [0.2, 0.8]
class = 1
node #4
gini = 0.0
samples = 21.4%
value = [1.0, 0.0]
class = 0
node #9
gini = 0.0
samples = 21.4%
value = [0.0, 1.0]
class = 1
node #6
gini = 0.0
samples = 7.1%
value = [0.0, 1.0]
class = 1
node #7
gini = 0.0
samples = 7.1%
value = [1.0, 0.0]
class = 0
node #11
gini = 0.0
samples = 7.1%
value = [1.0, 0.0]
class = 0
node #12
gini = 0.0
samples = 7.1%
value = [0.0, 1.0]
class = 1