

Based on our discussion related to the two decision tree classifiers on Chapter 6, continue working on our previous code, by constructing a C5.0 model using the same test data set and follow the same target variable, predictor variables, and minimum cases criterion. At the end, create a visualization of the decision tree, which is included in your notebook (not a separate file).

In [1]: *#import the nessary libraries needed*

```
import os
import graphviz
import pandas as pd
import numpy as np
import statsmodels.tools.tools as stattools
from sklearn.tree import DecisionTreeClassifier, export_graphviz
from sklearn import tree
```

In [2]: `adult_tr = pd.read_csv(r"C:\Users\admin\Desktop\Python\adult_ch6_training")`

In [3]: `adult_test = pd.read_csv(r"C:\Users\admin\Desktop\Python\adult_ch6_test")`

In [4]: *#create a target variable using from adult test.*

```
y = adult_test['Income']
```

In [5]: *#convert categorial varuiable to dummy variables*

```
mar_np = np.array(adult_test['Marital status'])
(mar_cat, mar_cat_dict) = stattools.categorical(mar_np, drop=True, dictnames=True)
```

c:\users\admin\appdata\local\programs\python\python39\lib\site-packages\statsmodels\tools\tools.py:158: Future Warning: categorical is deprecated. Use pandas Categorical to represent categorical data and can get_dummies to construct dummy arrays. It will be removed after release 0.13.
warnings.warn(

In [6]: *# combine the dummy variables back with the*

```
mar_cat_pd = pd.DataFrame(mar_cat)
X = pd.concat((adult_test[['Cap_Gains_Losses']], mar_cat_pd), axis = 1)
```

In [7]: *#specify names of combined matrix and the target variable*

```
X_names = ["Cap_Gains_Losses", "Divorced", "Married", "Never-Married", "Separated", "Widowed"]
y_names = ["<=50K", ">50K"]
```

```
In [8]: # train C5.0 model
c50_01 = DecisionTreeClassifier(criterion = "entropy", min_samples_split=75, max_leaf_nodes=5).fit(X,y)
```

```
In [9]: # obtain tree structure with labeles
data = export_graphviz(c50_01, out_file=None, feature_names=X_names, class_names=y_names)
```

```
In [10]: # display the plot
model = DecisionTreeClassifier()
model.fit(X, y)
dot_data = tree.export_graphviz(c50_01, out_file=None, feature_names=X_names, class_names=y_names
                                )
graph = graphviz.Source(dot_data)
graph
```

Out[10]:

```
In [11]: #run and indicate your answer on the given space...
Ans1 = input("describe the first two split: ")
```

describe the first two split: The first two split is into low and high entropy nodes

```
In [12]: #run and indicate your answer on the given space...
Ans2 = input("How about the third and final, compare the differences if any: ")
```

How about the third and final, compare the differences if any: The same

```
In [13]: #run and indicate your answer on the given space...
Ans3 = input("Does the test data result match the training data result?: ")
```

Does the test data result match the training data result?: yes

```
In [14]: #Student information
Name = input("Name: ")
Section = input("Section: ")
```

Name: Cristan Kim Xavier S. Lansang
Section: 01151

```
In [15]: import datetime
import socket
def get_Host_name_IP():
    try:
        host_name = socket.gethostname()
        host_ip = socket.gethostbyname(host_name)
        print("Hostname:", host_name)
        print("IP Address:", host_ip)
    except:
        print("No visible IP Address1")
get_Host_name_IP()
now = datetime.datetime.now()
print ("Time Stamp:", now.strftime("%Y-%m-%d %H:%M:%S"))
```

Hostname: DESKTOP-LSGH01V
IP Address: 192.168.254.104
Time Stamp: 2021-02-04 16:28:46

```
In [ ]:
```

