

Assignment-2

```
import csv

from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier
from sklearn.model_selection import train_test_split # Import train_test_split
function
from sklearn import metrics #Import scikit-learn metrics module for accuracy
calculation
from sklearn.tree import export_graphviz
from sklearn.externals.six import StringIO
from IPython.display import Image
import pydotplus

X = []
Y = []

file = "cosmetic.csv"
csvfile = open(file, 'r')
read = csv.reader(csvfile)
read = list(read)
read = read[1:]

for row in read:
    for column in [1,2,3,4,5]:

        if(column == 1):
            if(row[column]=='< 21'):
                row[column] = 1
            elif(row[column]=='21-35'):
                row[column] = 2
            elif(row[column]=='> 35'):
```

```
row[column] = 3
```

```
elif(column == 2):
```

```
    if(row[column]=='High'):
```

```
        row[column] = 1
```

```
    elif(row[column]=='Medium'):
```

```
        row[column] = 2
```

```
    elif(row[column]=='Low'):
```

```
        row[column] = 3
```

```
elif(column == 3):
```

```
    if(row[column]=='Male'):
```

```
        row[column] = 0
```

```
    elif(row[column]=='Female'):
```

```
        row[column] = 1
```

```
elif(column == 4):
```

```
    if(row[column]=='Single'):
```

```
        row[column] = 1
```

```
    elif(row[column]=='Married'):
```

```
        row[column] = 2
```

```
elif(column == 5):
```

```
    if(row[column]=='N'):
```

```
        row[column] = 0
```

```
    elif(row[column]=='Y'):
```

```
        row[column] = 1
```

```
X.append(row[1:5])
```

```
Y.append(row[5])
```

```
print(X)
```

```
print(Y)
```

```
# Split dataset into training set and test set
```

```
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.21,  
random_state = 0) # 70% training and 30% test
```

```
print(X_train)
```

```
print('End X')
```

```
# Create Decision Tree classifier object
```

```
clf = DecisionTreeClassifier()
```

```
# Train Decision Tree Classifier
```

```
clf = clf.fit(X_train,y_train)
```

```
#Predict the response for test dataset
```

```
y_pred = clf.predict(X_test)
```

```
print(y_pred)
```

```
# Model Accuracy, how often is the classifier correct?
```

```
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
dot_data = StringIO()
```

```
export_graphviz(clf, out_file=dot_data,
```

```
filled=True, rounded=True,
```

```
special_characters=True,feature_names = ['Age', 'Income', 'Gender',  
'Marital_Status'],class_names=['0','1'])
```

```
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
```

```
graph.write_png('buys.png')
```

```
Image(graph.create_png())
```

Output:

