**Code Documentation for Decision Tree Classifier Assignment**

The assignment was to import a dataset about various flower species, split the dataset into 2 subsets: Training data (70% of the rows) and Testing data (30% of the rows) and finally implement a decision tree classifier algorithm on the Iris Training dataset and test the accuracy with Testing Dataset.

**Explaining the code:**

First and foremost, I import all the necessary libraries for working on this project:

*# importing the needed libraries*

from sklearn.metrics import accuracy\_score

from sklearn import tree

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import LabelEncoder

import numpy as np

import pandas as pd

Next, I read the dataset into the code I was working on using the pandas library and print the dataset I imported to check its accessibility:

*# reading the data*

irisData = pd.read\_csv('Iris.csv')

print(irisData)  *# checking the imported dataset*

The next step is to assign the columns of the dataset to dependent and independent variables depending on their uniqueness. For our dataset, since ‘Species’ is the unique member of our dataset, it takes the y variable and all others are assigned to x. I did this using the numpy array functions:

*# setting the dependent and independent variables as x and y*

x = irisData[['SepalLengthCm', 'SepalWidthCm',

              'PetalLengthCm', 'PetalWidthCm']].values

y = irisData['Species'].values

Now, I move forward into pre-processing the ‘Species’ data into a way that can be easily understood by the computer by using the labelEncoder() method. This converts the values of the ‘Species’ data into numerical data based on the index of all occurring values. Hence, this encodes all first 50 species data to 0s since they are all the same species. Next occurring 50 become 1s and last 50 values are encoded as 2s:

*# using LabelEncoder to preprocess the data*

le = LabelEncoder()

y = le.fit\_transform(y)

print(y)  *# checking the preprocessed data of the Species*

The most important part of this project is where I split the dataset into training data and testing data and deciding how much data I was going to assign to both subsets of the data I have now. For this assignment, I assign 70% of the data to training and 30% of the data to testing. I was able to implement this by aid of the sklearn.model\_selection library and importing the train\_test\_split method:

*# splitting the dataset into training an testing data*

*# setting the training size to 70% and testing size to 30%*

x\_train, x\_test, y\_train, y\_test = train\_test\_split(

    x, y, *test\_size*=0.3, *train\_size*=0.7)

Next thing is to import the Decision Tree Classifier Algorithm into the code, assign it to a variable for easy access and use and make a classifier fit of the training data I have, using the various methods that come with the algorithm:

*# assigning the DecisionTreeClasssifier() method to the a variable*

dTreeClasssifier = tree.DecisionTreeClassifier()

*# using the DecisionTreeClassifier() method to fit the training data*

dTreeClasssifier.fit(x\_train, y\_train)

Finally, I attempt to make a prediction using the test data and calculate the accuracy of our prediction using the sklearn.metrics accuracy\_score method:

*# using the DecisionTreeClassifier() method to make a prediction using the x\_test data*

predictions = dTreeClasssifier.predict(x\_test)

*# printing the accuracy of the prediction*

print('The Accuracy of the Decision Tree Classifier is: ' +

      str(accuracy\_score(y\_test, predictions)))

The accuracy of my Decision Tree Classifier is: 0.9777777777777777

This is equivalent to 97%