**Code Documentation**

For the fourth assignment, I was expected to perform classification on the famous iris flower data set available on Kaggle on the link *“https://www.kaggle.com/uciml/iris”.*

I used the scikit learn library of python for this purpose. After importing pandas and numpy, I imported the tree from sklearn which is exactly what I needed to create my Decision Tree for the problem.

Next, I had to import the train\_test\_split from model\_selection of sklearn to split my data set into a 70:30 ratio as was stated in the problem statement. The training set had to be 70% of the entire dataset. On the contrary the test set had to be 30% of the entire data set.

Lastly, I had to import the metrics\_accuracy from sklearn to calculate the accuracy of my model.

Now coming to the main portion of the problem I had to read the data from the csv file to a pandas data frame for which I used the read\_csv function and stored the same in a data frame called df\_iris. After inspecting the data set, I came to the conclusion that the Id column was practically useless so I dropped it using the drop function. Now our data set consisted of only five columns excluding the index.

After this I created a new data frame named df\_X that consisted of our features for the problem. Then I created a numpy array X that consisted of values in the data frame df\_X but in an array format. Then I created an array y that consists of our species values i.e. the target value.

After doing this, we need to split our data set into train and test sets. For accomplishing this task, we now use the train\_test\_split earlier imported from the sklearn library. Now we check the size of our training and testing data set using the shape attribute on our data sets.

Now we create a classifier object for our model named clf. Then we train this classifier using the fit function and, training features and labels, as parameters. Once the model is trained, we can use the predict function with testing features as parameters. Now we have to store our predicted values somewhere and we do so by storing it in an array called pred\_tree.

Finally, to check the accuracy of our model we can use the accuracy\_score from the metrics we imported from sklearn earlier. The parameters for this function are going to be the actual labels for the feature sets and the values that our classifier predicted i.e. y\_test and pred\_tree.

In this case the accuracy that our model showed was 93.33%.