import numpy as np  
import pandas as pd  
from sklearn.model\_selection import train\_test\_split  
from sklearn.tree import DecisionTreeClassifier  
  
df = pd.read\_csv('Iris.csv')  
print(df)  
  
X = df[['SepalLengthCm'**,** 'SepalWidthCm'**,** 'PetalLengthCm'**,** 'PetalWidthCm']].values  
y = df['Species'].values  
X\_train**,** X\_test**,** y\_train**,** y\_test = train\_test\_split(X**,** y**,** test\_size = **0.3,**random\_state=**0**)  
  
dtc = DecisionTreeClassifier()  
dtc.fit(X\_train**,**y\_train)  
print(dtc.score(X\_test**,**y\_test))

For the assignment, we need to import a very important library of python, i.e. scikit learn in which there are various machine learning models including svm, knn, liear regression logistic regression etc.

We need to import train\_test\_split and DecisionTreeClassifier which is also a model of ML in scikit learn.

Next, we read the iris dataset in csv format using read\_csv() function.

The dataset includes three iris species with 50 samples each as well as some properties about each flower. The available columns in this dataset are: Id, SepalLengthCm, SepalWidthCm, PetalLengthCm, PetalWidthCm, and Species.

Splitting The Dataset

As there is only one dataset available (no separated training and test dataset) we need to divide the dataset into training and test dataset by ourself. To do this, we can use the train\_test\_split method from the scikit-learn. I configured it to split the dataset into 70:30 for training and test dataset as mentioned in the problem statement. I also define a random\_state equal to 0. The usage of defining random\_state is to make sure the splitted dataset is the same even if we split the dataset again and again.

Classification

Finally, we reach the last part that is the classification itself. We will use the decision tree classifier from the scikit-learn as mentioned in the problem statement.