**Task 3**

**Data set:**

Iris dataset: Target attribute class:{Iris Setosa, Iris Versicolour, Iris Virginica}.

https://archive.ics.uci.edu/ml/datasets/Iris

Attributes:

sepal length in cm

sepal width in cm

petal length in cm

petal width in cm

We will just use two features for easier visualization, sepal length and width.

**# import iris dataset**

**from sklearn import datasets**

**iris = datasets.load\_iris()**

**Exercise: Implement K-Nearest Neighbor (KNN)**

Your task is to implement KNN algorithm. To implement KNN you have to

* Split data into a train and a test split (70% and 30% respectively).
* Normalize the data:

**from sklearn.preprocessing import Normalizer**

**scaler= Normalizer().fit(x\_train) # the scaler is fitted to the training set**

**normalized\_x\_train= scaler.transform(x\_train) # the scaler is applied to the training set**

**normalized\_x\_test= scaler.transform(x\_test) # the scaler is applied to the test set**

* Implement a similarity (or a distance) measure. (use Euclidean Distance).

**def distance\_ecu(x\_train, x\_test\_point)**

**Input:**

**- x\_train: corresponding to the training data**

**- x\_test\_point: corresponding to the test point**

**Output:**

**-distances: The distances between the test point and each point in the training data.**

* Implement a function that returns top K Nearest Neighbors for a given query (data point).

**def nearest\_neighbors(distance\_point, K)**

**Input:**

**-distance\_point: the distances between the test point and each point in the training data.**

**-K: the number of neighbors**

**Output:**

**-df\_nearest: the nearest K neighbors between the test point and the training data.**

* You should provide the prediction for a given query (for a classification task you can use majority voting).

**def voting(df\_nearest, y\_train):**

**"""**

**Input:**

**-df\_nearest: dataframe contains the nearest K neighbors between the full training dataset and the test point.**

**-y\_train: the labels of the training dataset.**

**Output:**

**-y\_pred: the prediction based on Majority Voting**

* Test the KNN Algorithm on the test dataset
* Compare your implementation with Sklearn library
* Calculate the accuracy of both methods