**HO CHI MINH UNIVERSITY OF TECHNOLOGY AND EDUCATION**

**FALCUTY FOR HIGH QUALITY TRANING**

****

**Lecturer: Ph.D Le Van Vinh**

**REPORT PROJECT 3**

**TOPIC: K-Nearest Neighbors Algorithm**

**Group:**

**DANG TRONG QUOC HUY- 17110035**

**NGUYEN TUAN ANH - 17110002**

**SEMESTER: 1 – SCHOOL YEAR: 2020 – 2021**

**Lecturer’s comment:**

….…………………………………………………………………………………….…………………………………………………………………………………….…………………………………………………………………………………….…………………………………………………………………………………….…………………………………………………………………………………….…………………………………………………………………………………….…………………………………………………………………………………….…………………………………………………………………………………….…………………………………………………………………………………….…………………………………………………………………………………….…………………………………………………………………………………….…………………………………………………………………………………….…………………………………………………………………………………….…………………………………………………………………………………….…………………………………………………………………………………….…………………………………………………………………………………….…………………………………………………………………………………….…………………………………………………………………………………….…………………………………………………………………………………….…………………………………………………………………………………….…………………………………………………………………………………….…………………………………………………………………………………….…………………………………………………………………………………….…………………………………………………………………………………….…………………………………………………………………………………….……………………………………….…………………………………………………………………………………….………………………………………….…………………………………………………………………………………….………………………………………….…………………………………………………………………………………….………………………………………….…………………………………………………………………………………….………………………………………….…………………………………………………………………………………….………………………………………….…………………………………………………………………………………….………………………………………….…………………………………………………………………………………….………………………………………….…………………………………………………………………………………….………………………………………….…………………………………………………………………………………….………………………………………

**K-nearest-neighbor Algorithm**

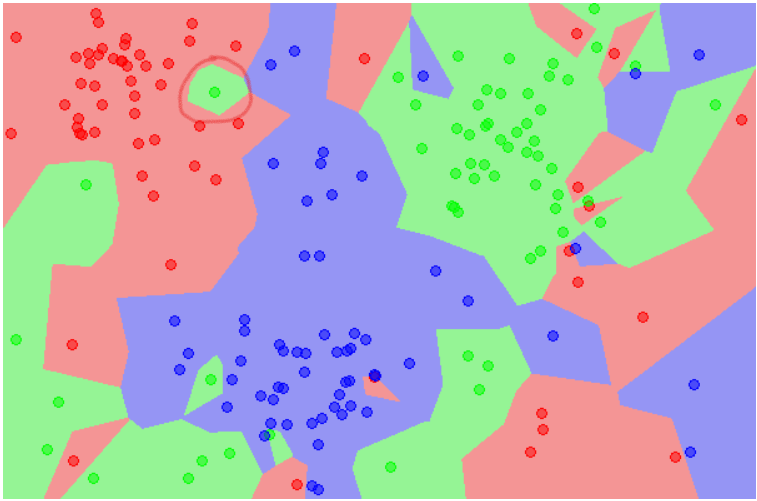
**Definition:**

KNN is one of the most basic Supervised-learning in Machine Learning. When training, this algorithm does not learn anything from the data training, all of the operation is implemented when it needs to guess the result of the new data test. KNN algorithm can be apply to both of the problem of Supervised-learning that is Classification and Regression.

KNN in a Classification problem, the label of the new data test can be deduced from K nearest data point in training set. The label of one test data can be decided by the major voting among the nearest points.

In short, KNN is the algorithm used to find out the result of the new data test by the way that it refers to the information of the K nearest data points.

The picture below illustrates the KNN algorithm in a Classification with K = 1.



The example above is a Classification problem with 3 Classes: Red, Blue and Green. Each data point will be assigned with a label depending on the colour that it belongs to. In this picture, a few little areas lie in the other bigger areas with different colour. For example, we can see the green point lies in between 2 larger areas with red and blue colour (I highlighted it with the red circle). This point can be a interfere point. It leads to, if the data test lies in this area, the result can be the incorrect answer.

**Advantages**:

1. The complicated of the algorithm of the training process is 0.
2. It’s very easy to predict the label of the new data test.
3. It’s unnecessary to assume about the distribution of each class.

**Disadvantages**:

1. It will not actual correctly if K is very small.
2. It will take many times to predict the label of the data test if the data set is very large.
3. The bigger K , the larger complexity.

**Example with Python**:

Iris flower dataset is a small data set. This data set consist of information of 3 kind of Iris flowers: Iris setosa, Iris virginica and Iris versicolor. Each kind has 50 flowers with 4 information about: length and width of sepal, length and width of petal. I will put the illustrate picture of 3 kind of Iris flowers below.

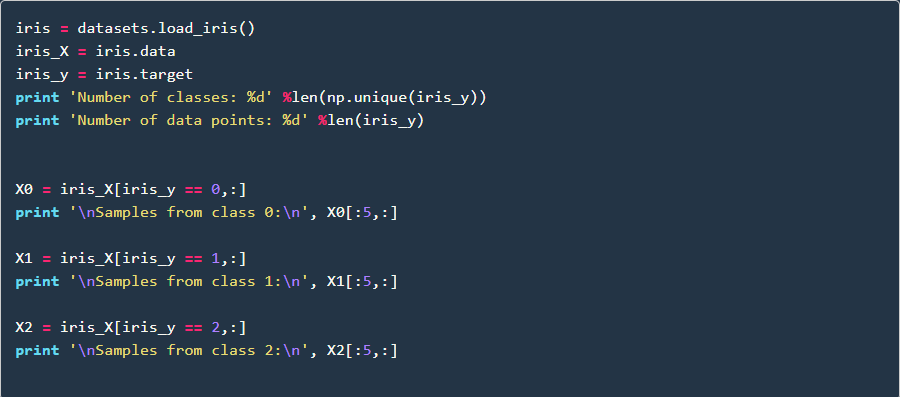


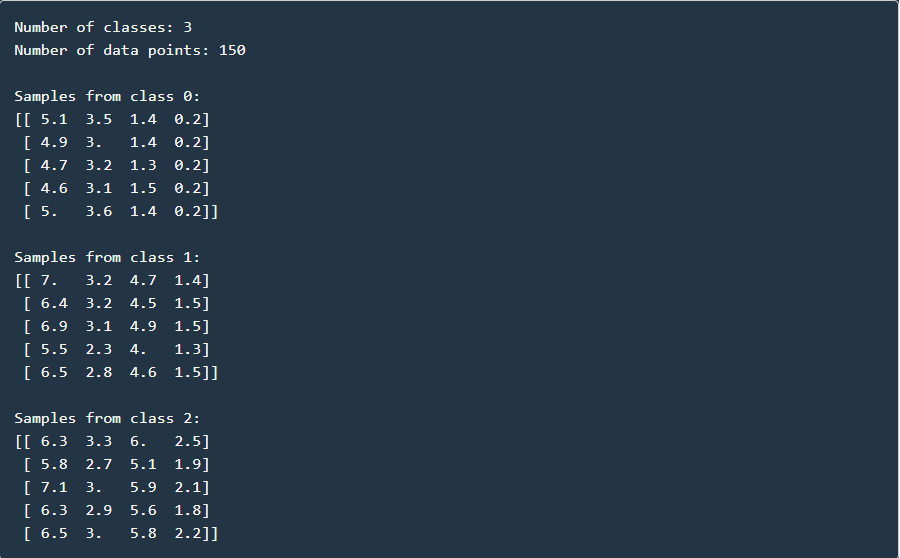
Experiment and coding:

First of all, we have to import some library of Scikit-learn.

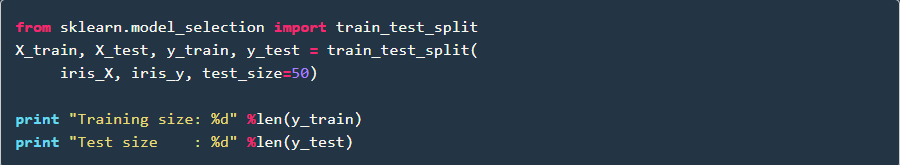


Next, we will load data and show on some data row. Classes are assigned to 0,1 and 2 label.



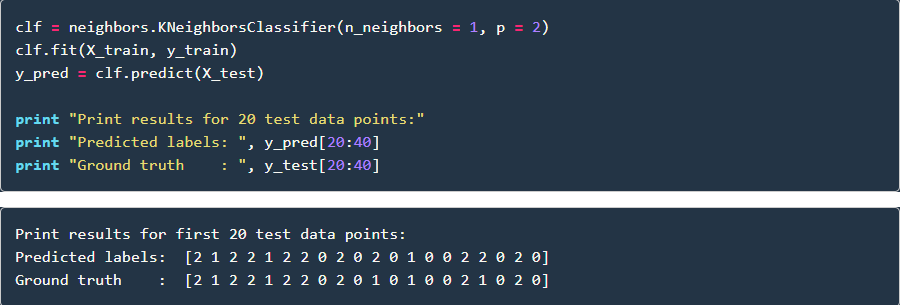


Next step, we have to divide the data set into training set and test set. In this case, I will use 50 data for the test set and 100 data for the training set. In the Scikit-learn library, we already have some functions to help us divide the data set randomly.





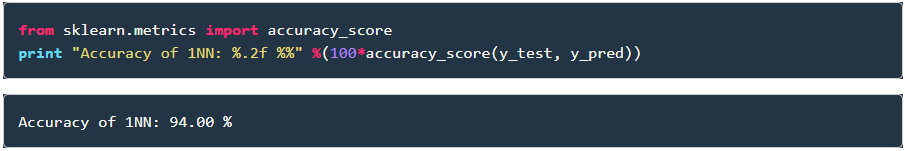
In this case, I will set K = 1. It’s mean that with each data test, we just refer to 1 nearest data point and the label of new data test will be the same with that data point.



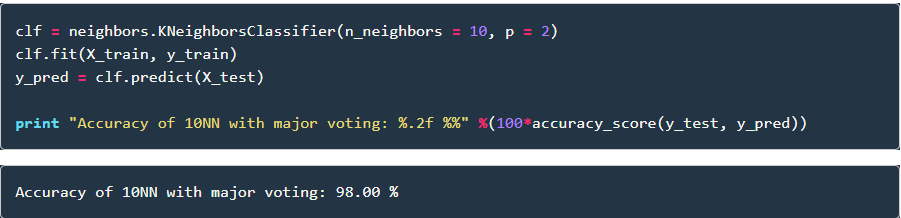
As the result, the prediction label of the data test is almost like the real label of that data test (ground truth).

**Evaluation method**:

In order to evaluate the accuracy of the KNN algorithm classifier, we will check how many data test in the test set had a correct label after the algorithm implemented. And then take that amount and divide with the amount of data test in the test set. The result is accuracy.



In order to increase the accuracy, we should increase the K value. For example, I will set K = 10, so let check the accuracy.



The second evaluation method is recall\_score:

In order to implement this method into your project. Firstly, we have to import recall\_score algorithm already built-in sklearn. And the second step is that we have to choose the type of averaging performed on the data, in recall we have 6 type of averaging: binary , macro , micro , sample , None.

In my project I will use macro.

