COMP 462 ASSIGNMENT #1

k-Nearest Neighbor Classification

02.16.2020
MEF University
Department of Computer Engineering

Artun Burak Meçik
mecika@mef.edu.tr
#041501021

Abstract

In this assignment, the iris dataset was studied by using a k-NN classification algorithm. In the k-NN algorithm, the effect of a number of neighbors and distance metrics on the system was examined. Also, the method created was compared with the "sklearn.neighbors.KNeighborsClassifier" library and visualized.

1.Dataset

1.1 Iris Dataset

As part of the project, I worked on the iris dataset. The dataset consists of 3 classes of 50 samples, each referring to a kind of iris plant. Each sample has 4 different features except for the name of the plant:

1. sepal length
2. sepal width
3. petal length
4. petal width.

Iris plant species:

1. Iris-setosa
2. Iris-versicolor
3. Iris-virginica

The length and width of the petal and sepal of the three iris species are shown in Figure 1.

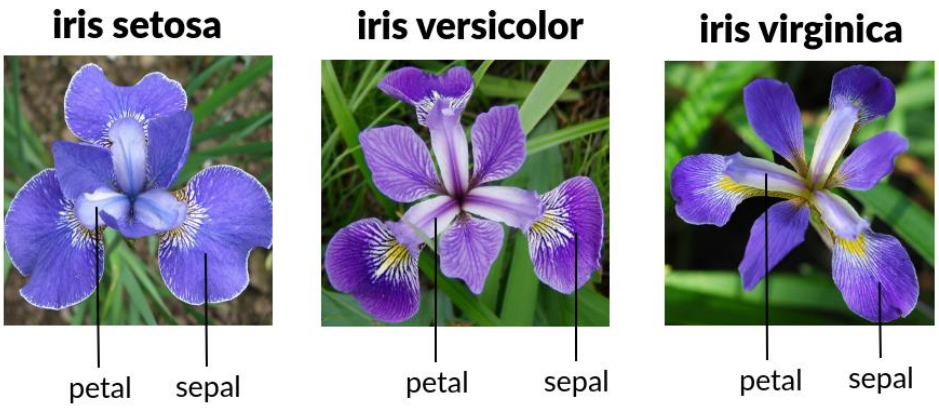

Figure 1. Three flowers in the Iris dataset: Iris Versicolor, Iris Setosa, and Iris Virginica

1.2 Training and Test Sets

There are 150 samples in the iris dataset, 3 classes and 50 samples in each class. The dataset was split into two as 30 sample training data from each plant and the remaining (20 samples) test data. As a result, 90 sample training data was used.

2. Classification

2.1 Preprocess

Within the scope of the project, only sepal length and petal width features were used as a distinguishing feature in order to accelerate the code.

2.2 Decision boundaries

At this stage, we created a decision boundaries map in line with our training data. If the training data in the data are not changed, any estimation is requested, it will respond according to these limits. Decision boundaries created for different k values can be observed on Chars 1-2-3.

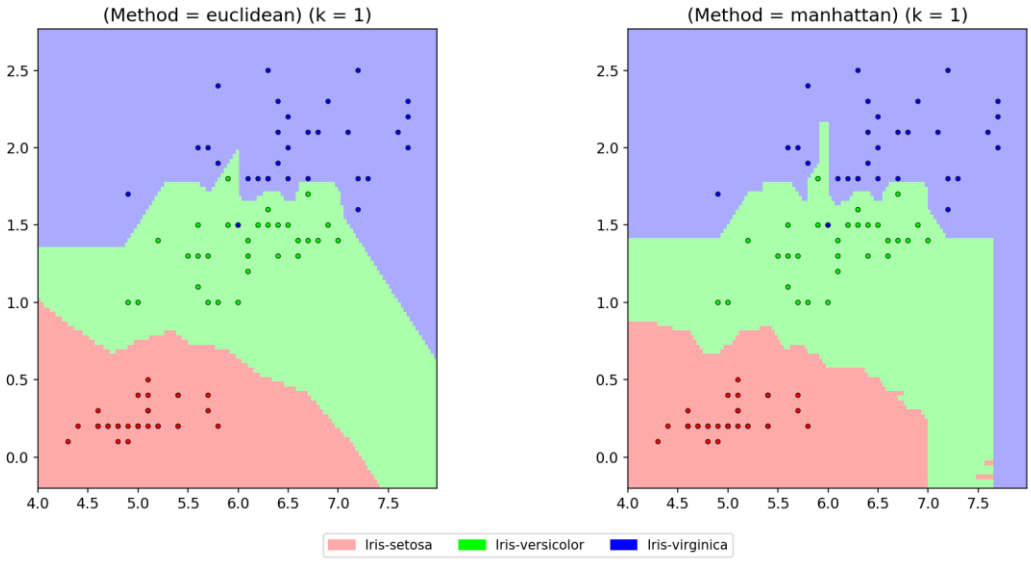

Chart 1. Decision boundaries for k=1

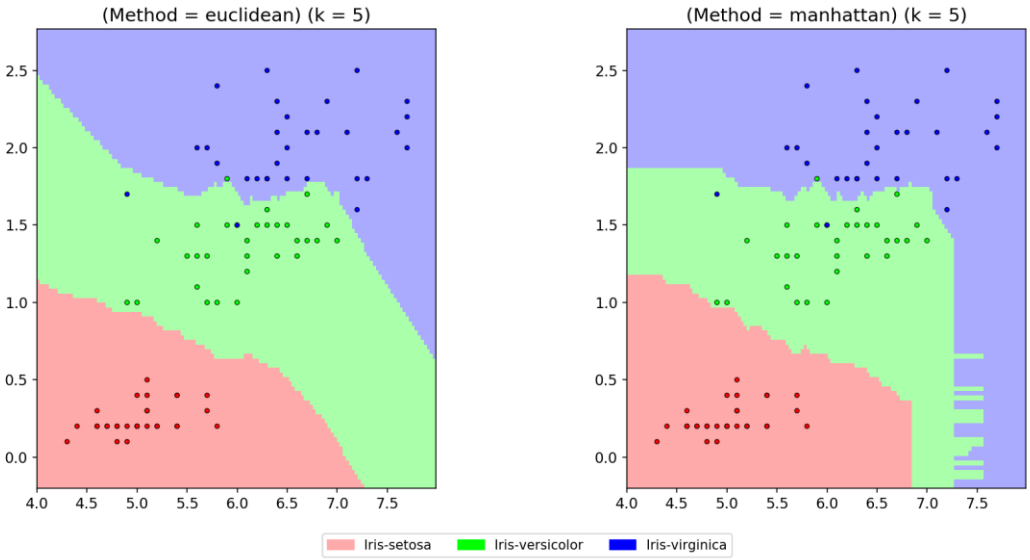

Chart 2. Decision boundaries for k=5

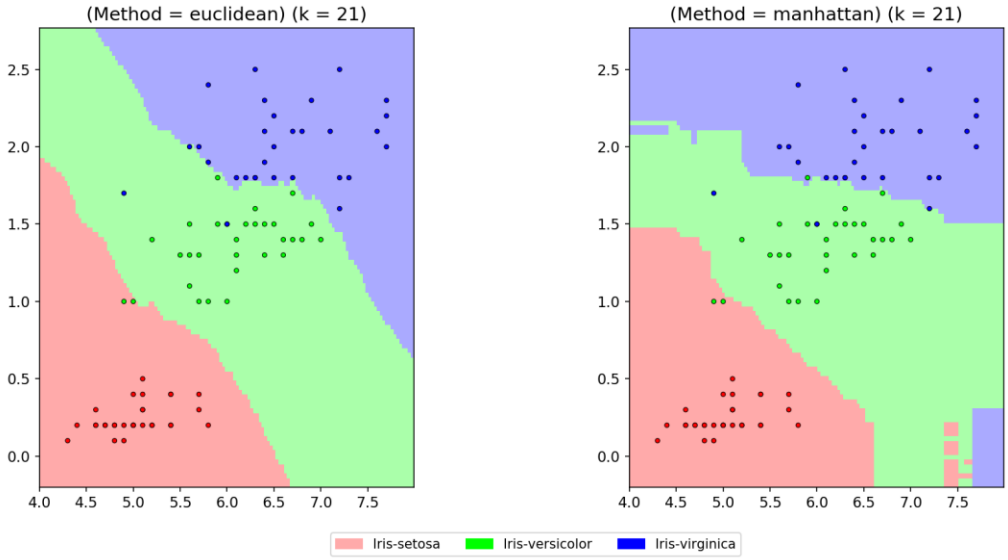

Chart 3. Decision boundaries for k=21

2.3 Result

Euclidean and Manhattan distance metrics were used to evaluate the test samples and were compared to the k-NN library in the sklearn library. Besides, accuracy rates were calculated for the different neighbor (k / n_neighbor) values of these algorithms. You can see the accuracy rates and incorrect estimation numbers for different k values of algorithms in Table 1.

	Euclidean Distance		Manhattan Distance		Sklearn Knn dictionary	
	Accuracy (%)	Error Count	Accuracy (%)	Error Count	Accuracy (%)	Error Count
k=1	93,33%	4/60	93,33%	4/60	93,33%	4/60
k=3	96,67%	2/60	96,67%	2/60	96,67%	2/60
k=5	96,67%	2/60	96,67%	2/60	96,67%	2/60
k=7	96,67%	2/60	96,67%	2/60	96,67%	2/60
k=9	96,67%	2/60	96,67%	2/60	96,67%	2/60
k=11	96,67%	2/60	96,67%	2/60	96,67%	2/60
k=13	96,67%	2/60	96,67%	2/60	96,67%	2/60
k=15	96,67%	2/60	95,00%	3/60	95,00%	3/60
k=17	96,67%	2/60	95,00%	3/60	96,67%	2/60
k=19	95,00%	3/60	95,00%	3/60	95,00%	3/60
k=21	95,00%	3/60	95,00%	3/60	93,33%	4/60
k=30	90,00%	6/60	90,00%	6/60	90,00%	6/60
k=50	83,34%	10/60	86,67%	8/60	81,67%	11/60

Table 1. Accuracy rates and a number of errors in algorithms for different distance metrics and k values.

When we compare distance metrics and k values, it is observed that a value of k between 3 and 13 gives a higher accuracy rate. No difference was observed when the K value was 1. In addition, it was observed that the accuracy rate decreased faster than the euclidean application in Manhattan between 13 and 21 k values. Based on these data, it can be said that the average k = 5 or 7 value and the euclidean metric are more suitable for the iris dataset. You can see the change graph of the metrics with different values of k in Chart 1.
However, after the k value exceeds 31, the decrease in manhattan is slower than the others. Therefore, although it is inefficient to use a high k value, manhattan distance metric can be used in high k values.

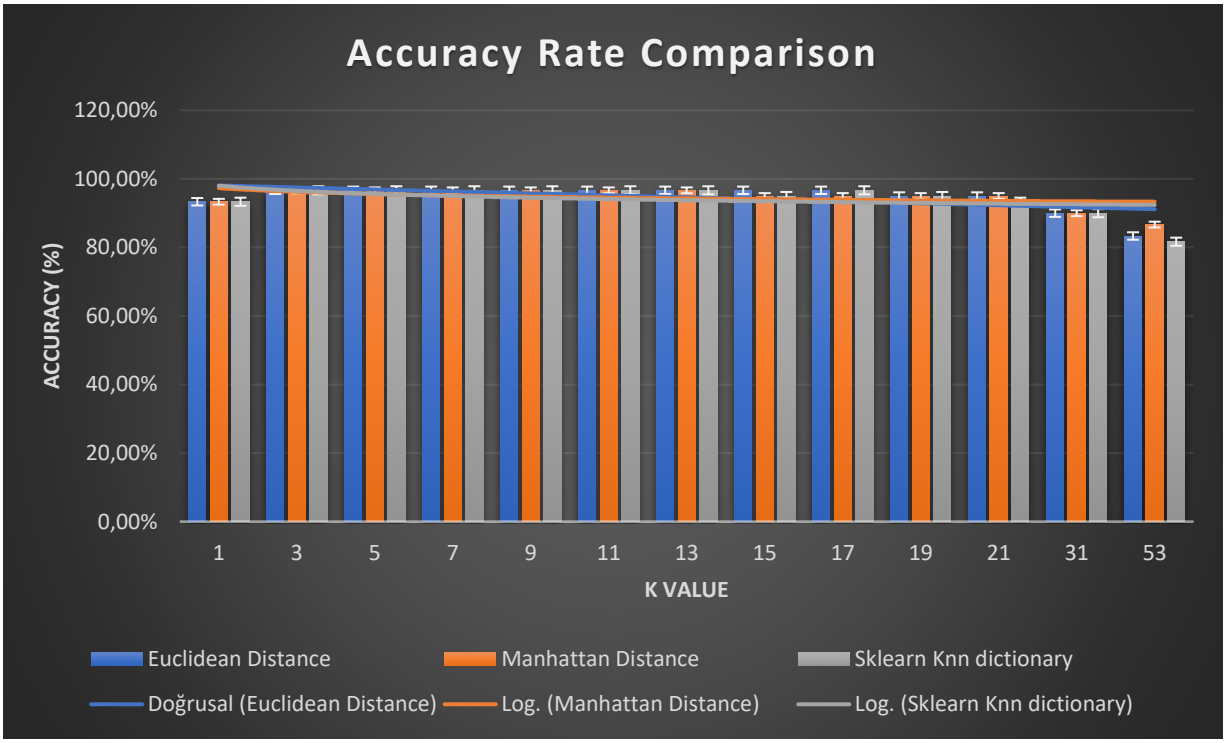

Chart 4. Accuracy rate comparison