Assignment 1 – k-Nearest Neighbor Classification

Elif Erden
Student ID: 041503038
Date: 24.02.2019

COMP 462
Introduction to Machine Learning

1. Algorithm Explanation

In this assignment, k nearest neighbor classification algorithm for the iris dataset was implemented and tested for predicting iris names according to given features. The details of the algorithm are explained below.

Dataset

Iris dataset contains three flowers: iris-setosa, iris-versicolor and iris-virginica. Each flower is represented by four features: sepal length, sepal width, petal length, and petal width. In this assignment, the first and fourth features were used to predict the labels. Complete iris dataset is provided as a text file.

Training and Test Sets

In the iris dataset, each flower has 50 samples. First 30 samples from each flower class were put into the train set and the last 20 samples from each flower class were put into the test set. One column was added to the training set to be filled with the distances and one column was added to the test set to be filled with the predicted labels of irises in the test set.

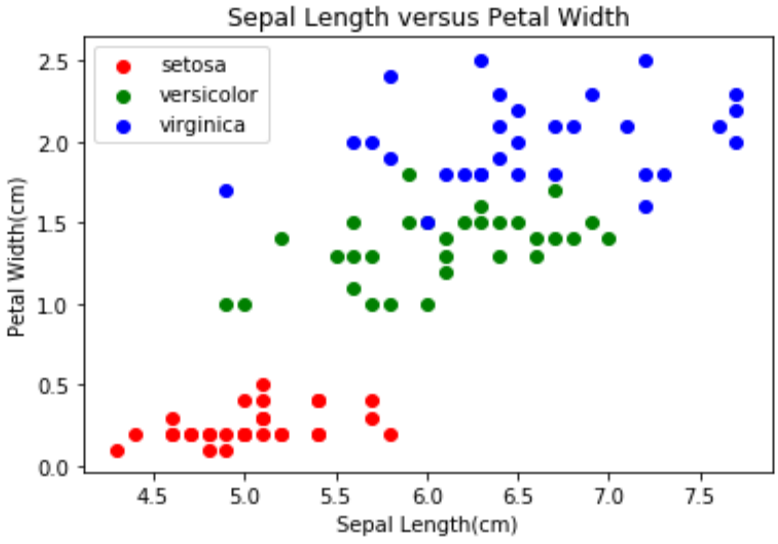

Figure 1. Scatter plot of irises in the training set according to sepal lengths and petal widths.

After training and test sets were created separately, the scatter plot of irises in the training set according to sepal lengths and petal widths is plotted to visually understand the general behavior of the irises by their sepal length and petal width features. As it can be seen in Figure 1, while iris-versicolor and iris-virginica appear close to each other, iris-setosa is more reserved than other classes.

Before applying the classification algorithm, firstly distance metric is received from the user. There are three options for the distance metric: 0= Euclidean distance, 1= Manhattan distance and 2= Cosine distance. For implementing the classification algorithm, kNNclassify function was created and it takes train set, test set and the distance metric option as input parameters. In the kNNclassify function, firstly the distance metric is checked so that if the user enters a wrong input for the distance, the algorithm does not proceed. After checking the distance metric, k value for the number of neighbors is received from the user and distances between the point in the test set and every point in the training set are calculated according to the intended distance metric. These distances are put to the distance column in the training set and then the training set is sorted by the distance column in ascending order. From the beginning to the k lines in the training set, the numbers of each class are counted and the class with the highest value is taken as predicted label and put into the predicted label column for the related row in the test set. This process is continued while every point in the test set is predicted. When every iris was predicted, the test set that contains both predicted and real iris names is printed. For understanding the classification accuracies, getAccuracy function was created which takes test set as an input parameter, counts errors in the test set and calculates accuracies according to error counter and the length of the test set.

2. Results

Classification Results

The classification algorithm was tested for different k values and distance metrics on the iris dataset. In general, it was observed that misclassifications occur between iris-vericolor and iris-virginica classes and predictions for iris-setosa class are consistent since these type of flowers are more distinguishable than other classes. Classification results of k nearest neighbor algorithm are shown in Table 1 for different k values and distance metrics. As it can be seen in the table, Euclidean distance has the highest average accuracy and Cosine distance has the lowest average accuracy for varying k values. So in this iris dataset, the most useful distance metric to be used is Euclidean distance for more accurate predictions. On the k values side, when the k value takes 3 and 9, the average accuracy rate is the highest according to varying distance metrics and when k takes the value of 1, the average accuracy gets the lowest. Therefore, it can be inferenced that 1 should not be given as a value of k for this dataset.

Table 1. k-NN classification accuracies for different k values and distance metrics.

	Euclidean Distance (L2-norm)		Manhattan Distance (L1-norm)		Cosine Distance		Avg (%)
	Accuracy (%)	Error Count	Accuracy (%)	Error Count	Accuracy (%)	Error Count	
k=1	93,33	4/60	90	6/60	86,67	8/60	90
k=3	96,67	2/60	96,67	2/60	91,67	5/60	95
k=5	96,67	2/60	96,67	2/60	88,33	7/60	93,89
k=7	96,67	2/60	96,67	2/60	88,33	7/60	93,89
k=9	96,67	2/60	96,67	2/60	91,67	5/60	95
k=11	96,67	2/60	96,67	2/60	88,33	7/60	93,89
k=15	96,67	2/60	96,67	2/60	88,33	7/60	93,89
Avg (%)	96,19		95,72		89,05		

Decision Boundaries

In this section, according to different k values and distance metrics, scatter plots of irises in respect to sepal lengths and petal widths with decision boundaries are displayed.

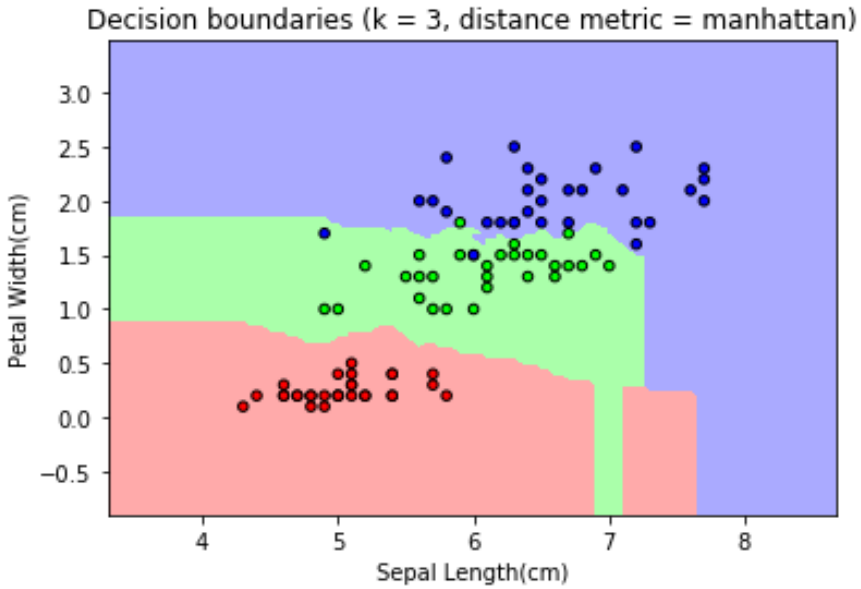

Figure 2. Scatter plot of irises according to sepal lengths and petal widths with decision boundaries (k = 3, distance metric = manhattan).

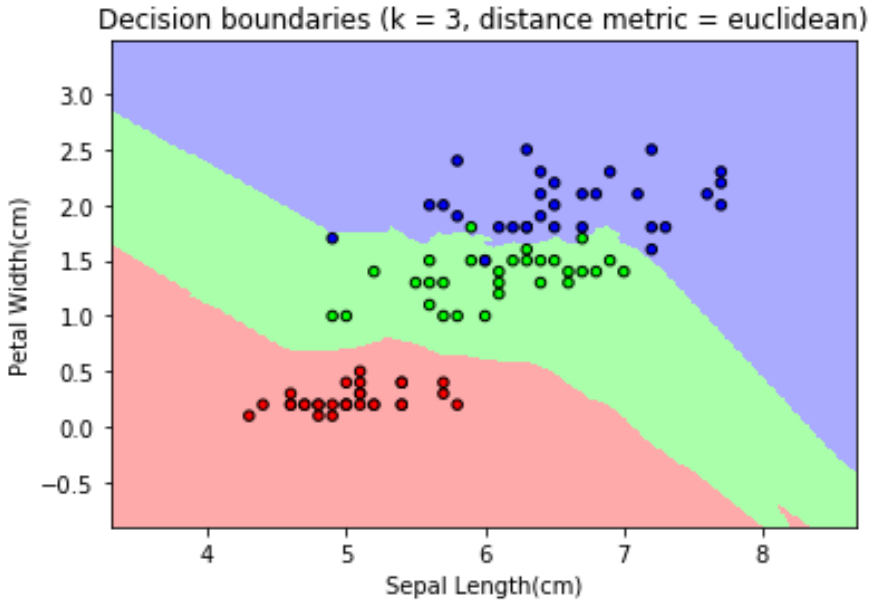

Figure 3. Scatter plot of irises according to sepal lengths and petal widths with decision boundaries (k = 3, distance metric = euclidean).

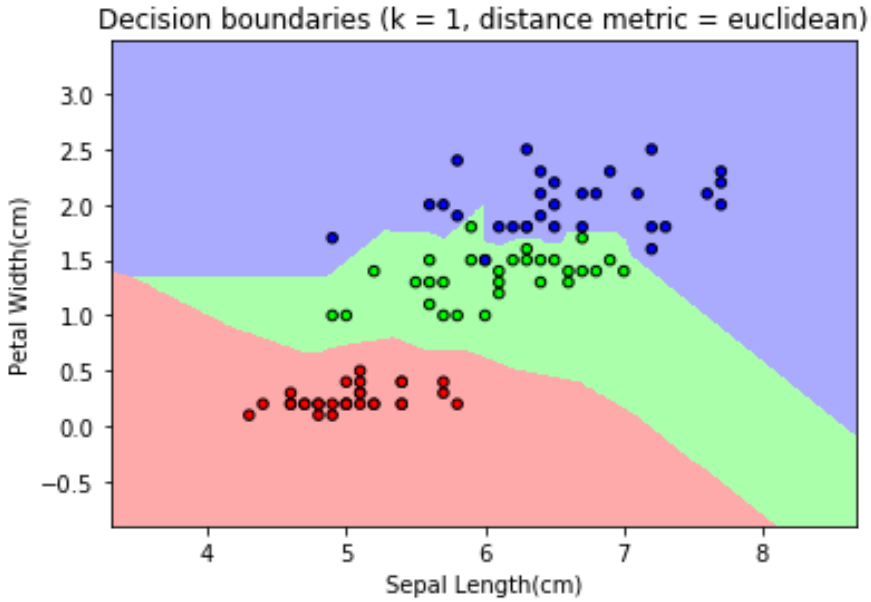

Figure 4. Scatter plot of irises according to sepal lengths and petal widths with decision boundaries (k = 1, distance metric = euclidean).

When the distance metric is Euclidean, although decision boundaries for the red class remain almost same in Figure 3 and Figure 4, boundaries get wider for the green class if the k value is equal to 3 in Figure 3. When the k value is 3, decision boundaries are more disorganized with the Manhattan distance in Figure 2 compared to decision boundaries with the Euclidean distance in Figure 3.