Stefano Pellegrini - Assignment 2

February 24, 2020

Exercise 1

1.1. Training and test results

1-Nearest Neighbour Training Accuracy (own implementation) = 1 = 100%
1-Nearest Neighbour Test Accuracy (own implementation) = 0.945993031358885 ≈ 94.6%

1.2. Discussion of the results

If we use the training set as a target for the prediction, the nearest close point of a target point will be the point itself. So with a 1-nearest-neighbor algorithm, I expected to have zero in-sample error and a training accuracy of 100%. We can observe that the model performs quite well on the test set with a prediction accuracy of 94,6%.

Exercise 2

2.1. Implementation

In our case, the k_{best} parameter is the k that minimizes the loss function. I found it by performing cross-validation on the training dataset with different values of k. For each iteration I stored the average loss, the accuracy (not requested) and the value of k in a tuple, then I appended each tuple to a list. At this point, it was sufficient to sort the list by classification error (the first element of the tuple) and select the k (the third element of the tuple) in the first tuple of the list.

2.2. Found parameter k_{best}

$k_{best} = 3$

Exercise 3

3.1. Training and test accuracy of k_{best}

k_{best} *(3-NN)* Training Accuracy = 0.971 = 97.1%
k_{best} *(3-NN)* Test Accuracy = 0.9494773519163763 ≈ 94.9%

Exercise 4

4.1. Discussion of the three normalization variants

The first version is the correct one. It obtains the scaler from the training data and then applies the scaler to both training and test datasets to center and normalize them. The second version is flawed

because it applies two different scalers to the two datasets, this is not a good approach because they should be normalized in the same way. The third version is also wrong because it obtains the scaler by computing the mean and the standard deviation of the training and test set combined, in this way the training input will not be transformed such that the mean and the variance of every feature are zero and one, respectively.

4.2. Parameter k_{best} on normalized data

k_{best} (normalized data) = 3

4.3. Training and test accuracy of k_{best} on normalized data

k_{best} *(3-NN)* Training Accuracy (normalized data) = $0.972 = 97.2\%$
k_{best} *(3-NN)* Test Accuracy (normalized data) = $0.9599303135888502 \approx 96.0\%$

4.4. A short discussion comparing results with and without normalization

Comparing the accuracy of the model before and after normalization of the datasets, is possible to observe that, after normalization the training accuracy is only slightly larger, in fact, there is an increase of 0.1%. Instead, the increase of the test accuracy after normalization is of 1%, which is ten-fold larger than the increase observed in the training accuracy. We center the data to removes any bias in the inputs and then we normalize it to ensure that all the input variables have the same scale, after input centering and normalization our model shows a quite better generalization to out-of-sample performance.