University of Copenhagen

Faculty of Science

Introduction to Data Science

Assignment 2

Dimitrios Galinos (`bst265`)

February 23, 2020

1 Nearest neighbor classification

The code is included inside the dimitrios_galinos.zip and is named "assignment2.py". Please also read the README file.

I have applied sklearn's nearest neighbor algorithm and have set it up to work with 1-NN (1 nearest neighbor). For our train data the accuracy of 1-NN is 1.0 (100% accuracy) and for our test data the accuracy of 1-NN is: 0.945993031358885. Since we are only checking 1 nearest neighbor I was expecting that the accuracy of our train data would be 100% because of the fact that each point the K-NN algorithm tries to predict has as nearest neighbor himself (because the K-NN algorithm was also trained on the train set) and therefore it is always correct. I was also expecting that the accuracy on our test data would not be 100% and since the exercise states that the classification results for these data are pretty accurate the 95% accuracy we got seems reasonable.

2 Cross-validation

For this part of the exercise I set up K-NN Classifiers for k=[1,3,5,7,9,11] and used the cross_val_score() function with our train data and cv=5 (Cross validation folds=5) from sklearn.model_selection to quickly get the average accuracy of each K-NN classifier for the 5-fold cross validation on my XTrain data. I found that the best accuracy is achieved by $k_{best} = 3$ because the 3-NN classifier had the biggest average accuracy in our 5-fold cross validation out of all the other classifiers.

3 Evaluation of classification performance

I've set up and fitted the k_{best}-NN classifier which was found to be 3-NN using the complete train data and got the below results:

The performance of the 3-NN Classifier for our train data is: 0.971
The performance of the 3-NN Classifier for our test data is: 0.9494773519163763

I was expecting the accuracy if the train data to not be 100% anymore because if $k > 2$ then there are possibilities for wrong classification and I also expected the increase in accuracy for our test data (even if it is a small one) since our cross validation found that 3-NN is better than 1-NN.

4 Data normalization

In this part of the exercise I have first normalized the data myself by subtracting from the Train and Test data their means respectively and then dividing the results by their respective standard deviations. That produced the normalized datasets of XTrain and XTest with zero mean and unit variance. When it comes to the versions provided by the exercise the first version is flawed because the Test dataset is getting normalized using the mean and standard deviation of the train dataset. The second version is correct because each dataset is normalized correctly by their own fit. Finally the third version is flawed because Test and Train datasets get normalized by the mean and variance of their concat() instead of using their own respective mean and standard deviation. Moreover my own normalization results are the same as the ones from version 2.

I then repeated the process from exercise 2 and 3 with the difference of using the new normalized datasets. My k_{best} parameter for the normalized data was found to be 3 because the 3-NN classifier once again had the biggest average accuracy in the new 5-fold cross validation I applied with the normalized Train dataset.

Then I set up and fitted the 3-NN classifier with the whole normalized training set and got the results below:

The performance of the 3-NN Classifier with normalization for our train data is: 0.972
The performance of the 3-NN Classifier with normalization for our test data is: 0.9599303135888502

The results with normalization are better. Algorithms like the K-NN are influenced by the measurement units of the variables. This means that if we want better and more accurate results we need to normalize the data before training and use of the algorithms. This is done in order to eliminate miss classifications that are caused because some features are being counted in really high numbers and others really low.