Programming Assignment 5

Write a program that implements a 2-class kNN classifier with an IB2 created case base with the following weighting scheme[1]:

$$w_i = \begin{cases} \frac{d_k^{NN} - d_i^{NN}}{d_k^{NN} - d_1^{NN}} & d_k^{NN} \neq d_1^{NN} \\ 1 & d_k^{NN} = d_1^{NN} \end{cases}$$

where d_k^{NN} is the farthest nearest neighbor and d_1^{NN} the nearest. Then w_i represents the weight of nearest neighbor i with its distance d_i^{NN}. The assignment for the point is then determined by the sum of the weights per class, whereas the maximal sum is the "winner".

Given are the two data sets[2] named *Example* and *Gauss* as tsv files from the last two assignments. The points have been shuffled around and are to be processed in that order. For that purpose, the Example data set has two different shuffles, which result in two different case bases when performing IB2.

Your task is to create the case base using IB2 and leave the other points for classification. IB2 is being adapted for each k, meaning that the case base is tested against a kNN classifier instead of just the nearest neighbor like on the slides. You are then required to classify the left over points using the case base and calculate the absolute number of misclassifications for different k (namely 2, 4, 6, 8 and 10). The output of your program should then be a tsv file, that contains the following content:

1. The first line contains the tab-separated error values for $k = 2, 4, 6, 8, 10$
2. The following lines are the data points from the input file, that contributed to the 4-NN case base. They should be printed in the same order they have been added to the case base (i.e. the original order given in the data set).

If the program fails, the data format is incorrect or I have to change source code, in order to make it work, you will get zero points. Machine learning libraries are not allowed. You can use libraries for handling the CSV/TSV format and the input parameters.

Your program must accept *at least* the following parameters:

1. **data** - The location of the data file (e.g. /media/data/Example.tsv).
2. **output** - Where the output tsv should be written to.

Please prepare example statements on how to use your program. E.g. for a python program:

[1] Jianping Gou, Taisong Xiong, Yin Kuang, "A Novel Weighted Voting for K-Nearest Neighbor Rule", Journal of Computers vol. 6, no. 5, pp. 833-840, 2011

[2] http://wwwiti.cs.uni-magdeburg.de/iti_dke/Lehre/Materialien/WS2018_2019/ML/res/kNN.zip

```
python3 kNN.py --data Example.tsv --output Example_4NN_Solution.tsv
```

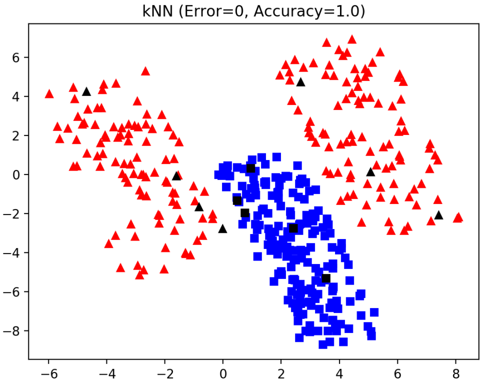
The final program code must be sent via email until Sunday, 13th of January 2019, 23:59 to marcus.thiel@ovgu.de. Please format your e-mail header as follows:

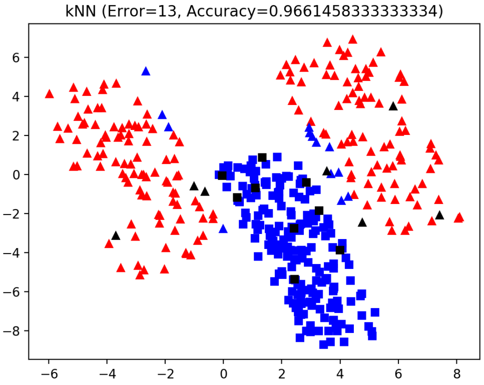
[Exercise Group] ML Programming Assignment 5

Replace *Exercise Group* with the day and time of your exercise group. E.g for Monday from 13:00 to 15:00 it would be:

[Monday 13-15] ML Programming Assignment 5

The figures below show the 4-NN solutions for the two different shuffles of the *Example* data set. The black points are inside the case base. You can only gain one point, if both shuffles of the Example data set are treated correctly. The other point is given for the Gauss data set.


kNN (Error=0, Accuracy=1.0)


kNN (Error=13, Accuracy=0.9661458333333334)

2 points