Artificial Intelligence – HW3 Report

Submitters:

Sahar Cohen – 206824088

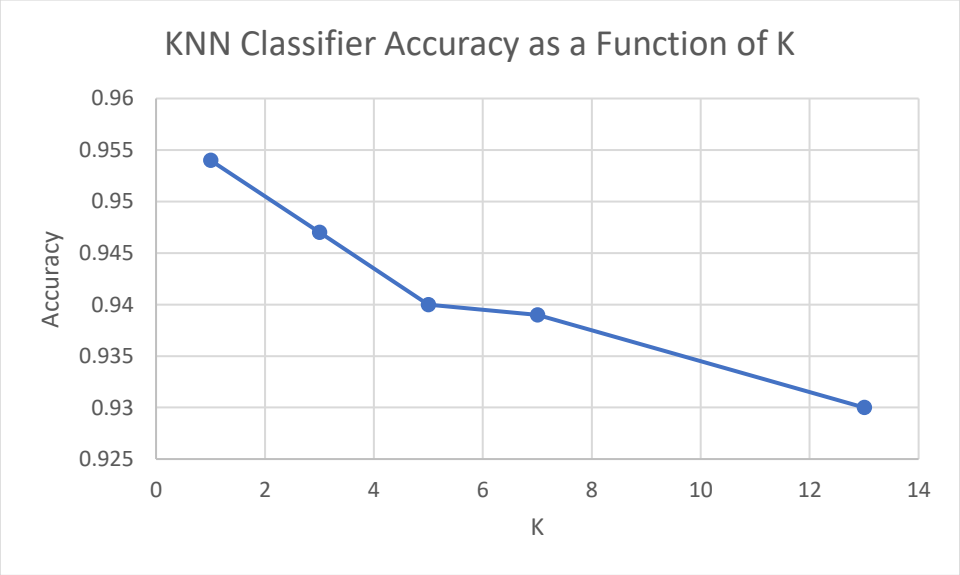
Yuval Nahon – 206866832

Part B:

3) It's important to stay consistent with the data folds (that is, not run the *split_crosscheck_groups* method again) so we don't mix up the data that we train with and the data that we test on. Later, we'll want to perform k-fold cross validation on our KNN classifier. It is important to keep the subsets divided as they are for this purpose, because in each iteration of this evaluation process: we choose a different subset of the data (=fold) to train on; and it is critical **not** to test the classifier on data that it has been previously trained with: it basically invalidates our classifier's results.

For that reason, we keep the original data folds as they are. We will also refrain from changing them in between evaluations because the classifier will obtain different results on different subsets: attached to this assignment is the subsets split that we evaluated our classifiers with. It is important to have it stay as it is for the purpose of validating the results we achieved.

5) The following graph showcases the accuracy of the KNN classifier on the given dataset (folded to the two subsets we've seen earlier):



We can see that the KNN classifier reaches decent results. The K values that were tested for the sake of this graph are: 1, 3, 5, 7, 13.

The graph shows that for higher values of k - we get lower accuracy. So, the optimal accuracy is reached for k = 1, and minimal accuracy for k = 13. Of course, this is not the general case as we've seen in class.

We suspect that the reason for this behavior lies in the structure of the dataset: most labels are categorized *True* whereas only a few are *False*. There is some pattern for the *False* labels, however: there are 187 features in total for every ECG test of every person in the dataset. We know from class that the KNN classifier is very sensitive to the "curse of dimensionality", which means it gets worse results the more irrelevant features we have in our dataset. In other words, the Euclidean distance of a *False* labeled object is not necessarily close another *False* labeled object that the classifier has been trained on previously – even if the two objects share virtually the same values in many of their features. Thus, we can conclude that if we were to take away all the irrelevant features: the graph's behavior would've been different. We suspect that the maximum value wouldn't have been for k=1 in this situation, again, following what we've seen in class and the explanation above.

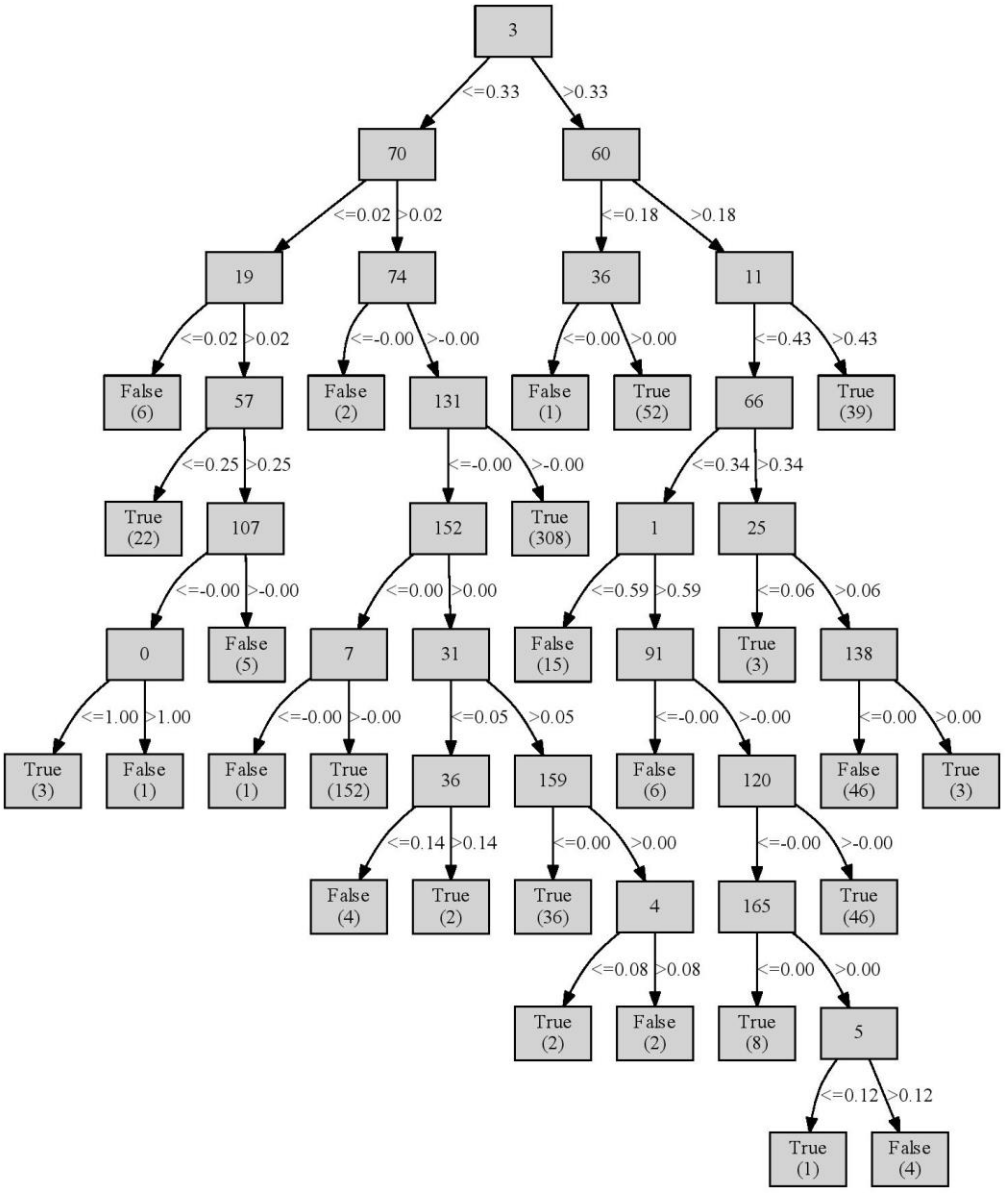
7) The best result came from the KNN classifier with k = 1: its accuracy is 0.954. Second best result was from the perceptron algorithm, with an accuracy of 0.905.

It is worth noting that, on average (after performing 100 experiments), the ID3 classifier reaches an accuracy rate of 0.898, but its best performance (algorithm has a random element) was better than the perceptron.

Competition Classifier:

The classifier we created is a committee that consists of three decision trees: CART (information gain heuristic), CART (gini heuristic) and ID3. These decision trees prune leaves with less than 4 classifications in them. Reason for this is that after our observation of the 187 features - we found many splits that were likely "noise". An illustration is added on the next page to support our statement. The illustration represents the ID3's resulted tree.

Each inner node in the illustration represents a feature, the edges represent the values that the feature was split by and the leaves represent the classifications and their amounts. In feature #7, for example, we can a split with 1 *False* classification and **152** *True* classifications. This is likely noise considering the magnitudes of this difference and served as our legitimacy in pruning leaves.


3
<=0.33
>0.33
70
60
<=0.02
>0.02
<=0.18
>0.18
19
74
36
11
<=0.02
>0.02
<=-0.00
>-0.00
<=0.00
>0.00
<=0.43
>0.43
False (6)
57
False (2)
131
False (1)
True (52)
66
True (39)
<=0.25
>0.25
<=-0.00
>-0.00
<=0.34
>0.34
True (22)
107
152
True (308)
1
25
<=-0.00
>-0.00
<=0.00
>0.00
<=0.59
>0.59
<=0.06
>0.06
0
False (5)
7
31
False (15)
91
True (3)
138
<=1.00
>1.00
<=-0.00
>-0.00
<=0.05
>0.05
<=-0.00
>-0.00
<=0.00
>0.00
True (3)
False (1)
False (1)
True (152)
36
159
False (6)
120
False (46)
True (3)
<=0.14
>0.14
<=0.00
>0.00
<=-0.00
>-0.00
False (4)
True (2)
True (36)
4
165
True (46)
<=0.08
>0.08
<=0.00
>0.00
True (2)
False (2)
True (8)
5
<=0.12
>0.12
True (1)
False (4)

Project Structure:

The project contains the following files:

classifier.py – **all** our classifiers and classifier factories were implemented in this file, as well as the Euclidean distance utility function, utilized by the KNN classifier.
experiments.py – contains all the methods in part B for question 3-7, as well as some methods for testing said methods, and several other methods for running the desired experiments and creating the .csv files.
hw3_tests.py – unit tests for the KNN classifier.
ecg_fold_0.data, **ecg_fold_1**.**data** – pickle files that were the result of running the *split_crosscheck_groups* on num_folds=2.
experiments6.csv – generated by the *create_experiments6* method in experiments.py. This file showcases the performance of the KNN classifier on different values of K. The graph from this report was extracted from this file.
experiments12.csv – generated by the *compare_classifiers* method in experiments.py. This file showcases the difference in performance of ID3 and perceptron.
experiments12_avg.csv – same as **experiments12.csv**, but the ID3 algorithm was run 100 times and the average performance was extracted.
experiments6_copy.csv, ecg_fold_test_i.data – these files were created for testing purposes and are not included in this report.

External Code:

We utilized the following in part B question #7 and for part C:

sklearn (linear model and DecisionTreeClassifier), ID3-decision-tree.

Our requirements.txt contains many modules, most of them not in use but removing one breaks the others, so we kept all of them as they were.