2.1 Nearest neighbor

In this task a dataset with medical records for Pima Indians was used. The goal of the task was to implement the K-NN algorithm to predict if a Pima Indian has diabetes based on their medical record. Different K should be evaluated.

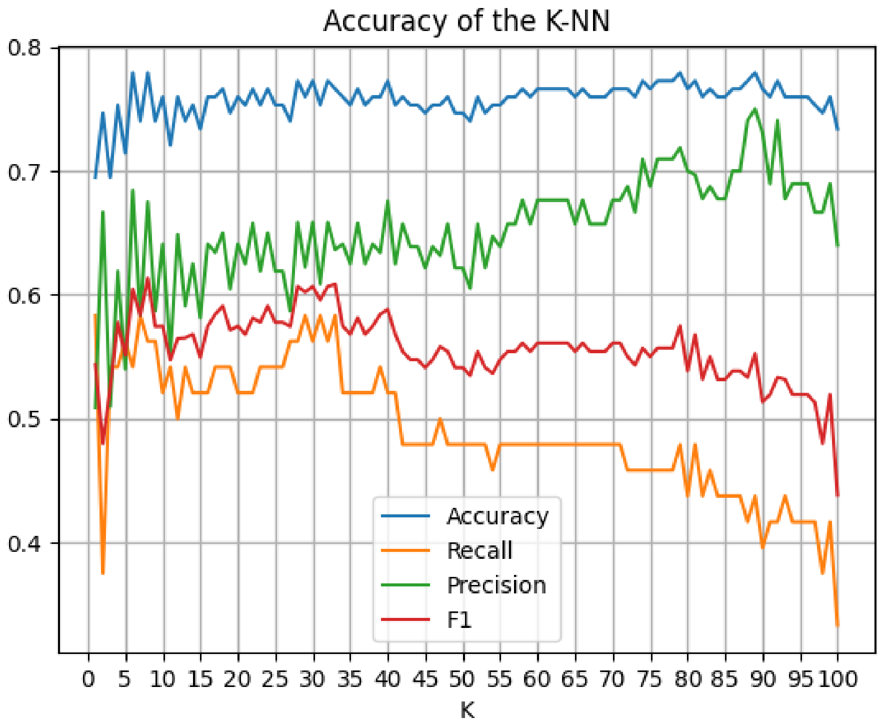
We were shown an example of a 1-NN algorithm on the same data before implementing K-NN ourselves. This made the task much easier as we were allowed to modify the code from 1-NN to be K-NN instead of implementing from scratch.

1-NN loops through the data searching for the element with the shortest distance to the one we try to predict to find a prediction and remember the nearest element it found. In K-NN we want to remember the K nearest elements, so we need a data structure. In my implementation I used a binary heap.

The heap was sorted with the element with the biggest distance at the top. We add all the elements to the heap, but if the size of the heap reaches K, we remove the most distant element after adding an element. This way after iterating over the data we are left with the K nearest neighbors.

The most frequent value among the K nearest neighbors is our prediction.

Results:



From the result it looks like the accuracy doesn’t get affected much by K, but with a big K we gained some precision at the cost of recall. This means we increased false negatives but decreased false positives. If we look at F1 we see that it’s probably a good idea to keep K somewhere between 5 and 35.