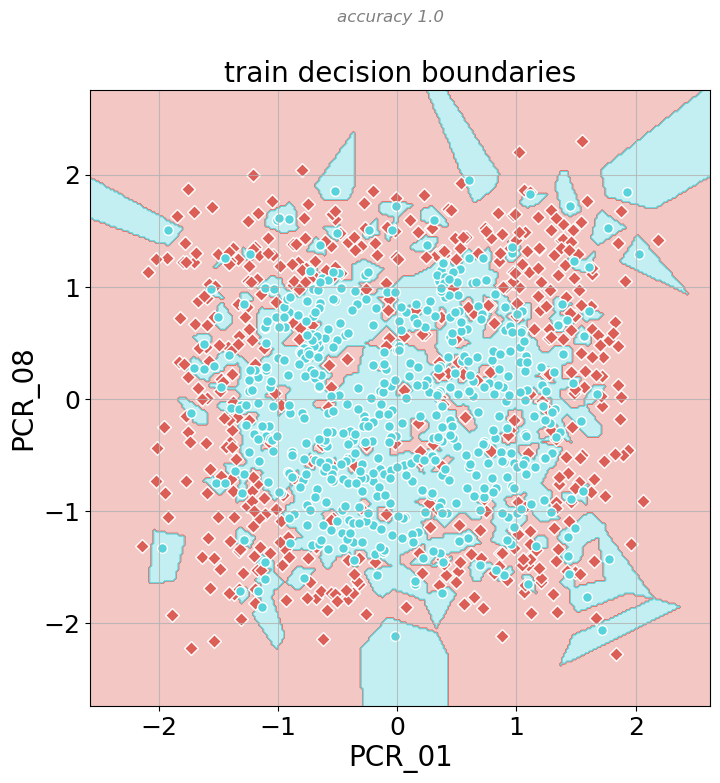
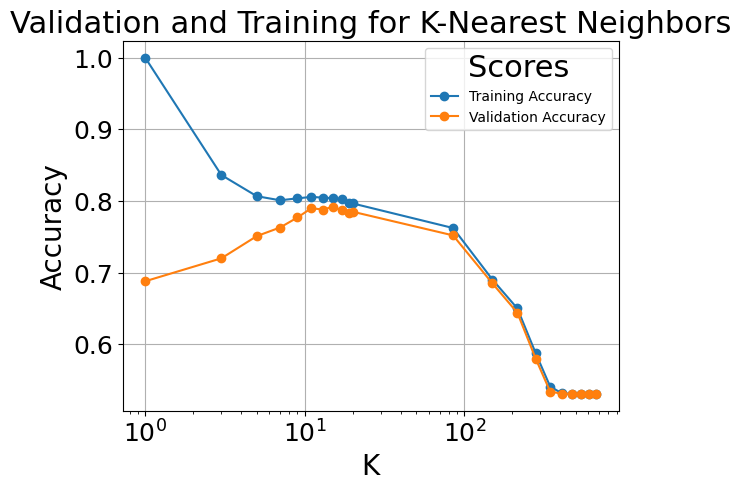
Part 1: Basic model selection with k-Nearest Neighbors

A diagram of a test

Description automatically generated1.



2.

The best score is:

The model is overfitting when the model scores significantly better on the training data compared to validation (unable to generalize). It happens when k ~ 1,3,5 probably because the small neighbor size mandates more sensitivity to noise and thus it’s harder for the model to capture the true distribution.

The model is underfitting when it seems it’s not able to learn from the data leading to low train and validation scores. It happens when we have large K’s such as 150 + probably because when considering these many points the predictions show low variance and are not very useful to predict a single point.

3.

A diagram of a train decision

Description automatically generatedA diagram of a test decision boundaries

Description automatically generated

4. Compared to K=1 we can see that the decision boundary is less dotted. This means that when K=15 the model is less sensitive to noise and seem to better predict the true distribution better.

Part 2: Decision trees

5. A diagram of a diagram

Description automatically generated with medium confidence

6.

A screenshot of a table

Description automatically generated

c. (one among) the best combination is:



d. a combination that causes under-fitting is max\_depth= 1 and min\_sample\_split = 2

e. a combination that causes over-fitting is max\_depth= 19 and min\_sample\_split = 2

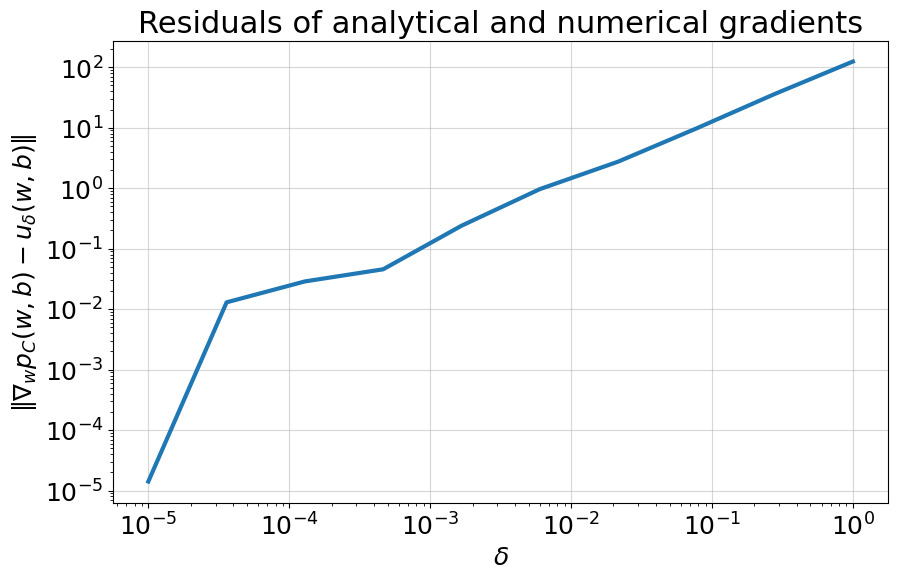
f. the first combination causes underfitting because a tree of depth 1 doesn’t have enough expressive power to actually learn something and make case specific predictions. The second combination has too much power and can create exact cases to fit the training data perfectly but in the process losses generalization.

7. we tried every combination 5 times (for each fold as validation set) so .

In CV we need to try all the possible combination so number of times will be the cross product of the sets time the folds. If we were to add a third parameter we’ll get a lot more combinations to check.

8. the test score with the best parameters is 0.76

9.



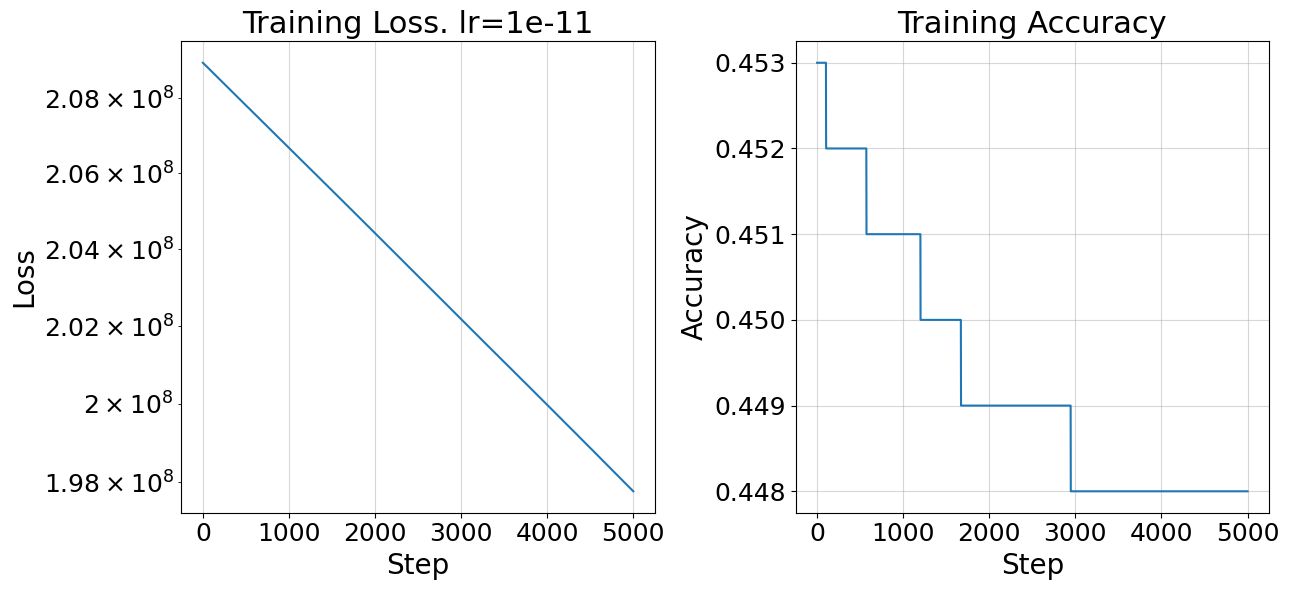
The graph makes sense because as the numerical gradient becomes the analytical gradient (as known from calculus) so we see the values getting closer to each other.

10. the train loss goes down as expected, meaning our optimization method works.

For the accuracy we will notice that the learning rate is very low and C is very high, meaning the model put emphasis on minimizing hinge loss which we can see doesn’t mean accuracy (overfitting). All these fluctuations are caused by the emphasis on hinge-loss while the learning rate ensures we converge (and don’t miss) to the minima in the loss plain.

11.

Plots:



A graph of a step and a step

Description automatically generated with medium confidence

A graph of a step and a step

Description automatically generated

A graph of blue lines

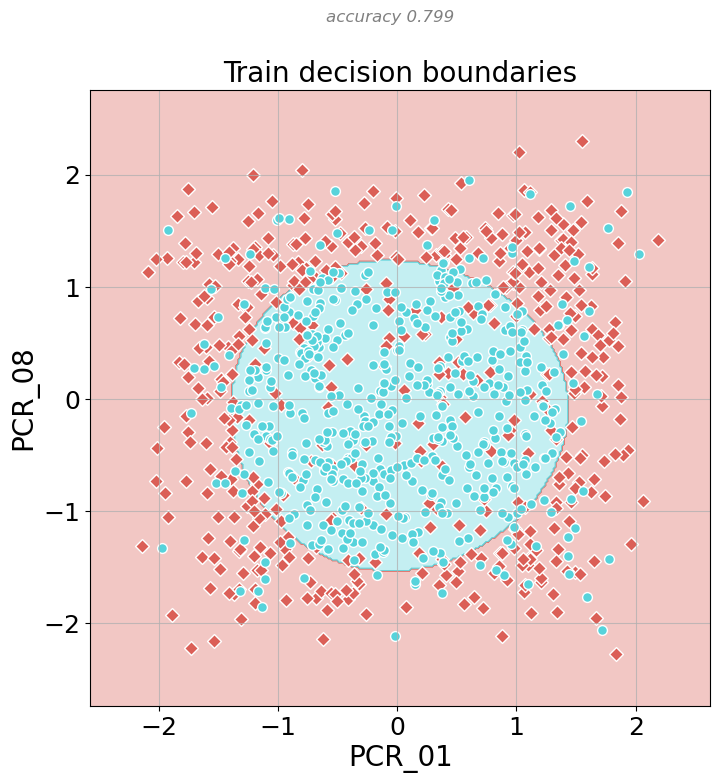
Description automatically generated with medium confidence

A graph of a performance

Description automatically generated with medium confidence

We would choose 1e-9 as the learning rate because as it converges to the minima in a fairly stable way (in contrast to 1e-7) and maybe with more iterations would even surpass 1e-7.

12.



A diagram of a test decision

Description automatically generated

13.

a. is a vector of length where is 1 iff has the word . When performing the dot product of two vectors we are multiplying each location in the two vectors, which result of that is 1 iff both vectors had 1 in the location k, and then summing over the ones that we got. The result of that will be the total number of word both sets have in common.

this kernel is defined because upholding-

b.

so it’s defined from kernel rules 1 and the previous result.

c. let denote then

Valid kernel from rule 2 and previous result.

d. same as in the dry hw.

if is a valid kernel and then

Now we will define so

Showing that k is a valid kernel.

e.

Is a valid kernel from d.

f. we saw in the blogpost that in RBF the numerator can be thought of as an equivalent to similarity. If we are trying to find a good measure for similarity of sets a good start would look at the number of words that can be found in both sets, but a better way would be also considering the portions those words make from all the words in the sets.

g. we can throw in more measures for similarity like length checking.

Say

And

14.

**A diagram of a train decision

Description automatically generated**

**A graph with red and blue dots

Description automatically generated**

The model is under-fitting as it seems it doesn’t learn much hence the low train and test scores. If we’re thinking of the Knn analogy than the classification of a point takes into consideration alot of train point thus we always get the same classification (the most prominent classification in the training set).

15.

A diagram of a train decision

Description automatically generated

A diagram of a test

Description automatically generated

seems a close simulation to Knn with k=1. At Q3 we found that the optimal k is 15 which created smoother decision lines.

Considering the hint maybe the difference can be caused by the numerical instability when the exponent of is a minus big number. The result will be very close to 0 and thus the farther points with won’t be considered properly when calculating the weighted sum of the dual problem, resulting in only the closer points have an effect.

16.

A diagram of a train decision

Description automatically generated

A diagram of a test decision

Description automatically generated

The model is overfitting. The decision boundaries depend solely on the training data like in Knn when k=1.