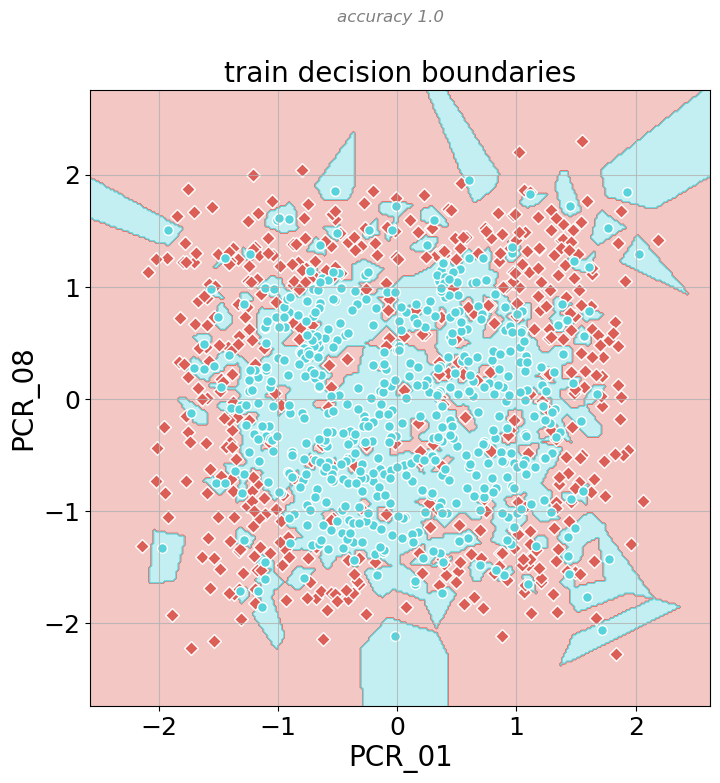
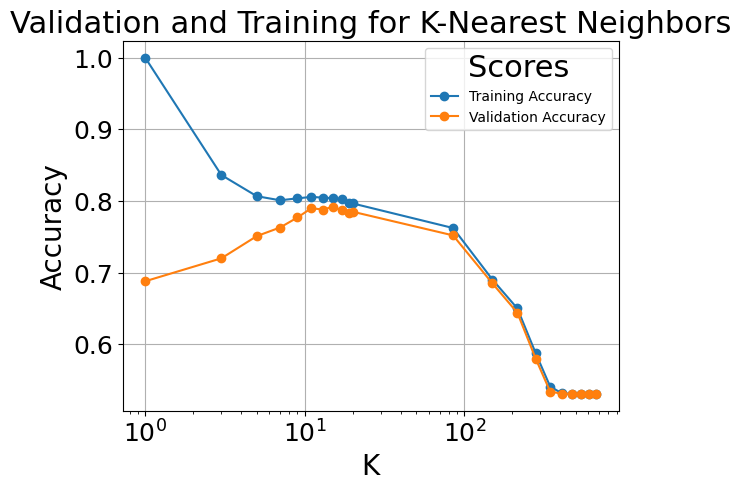
Part 1: Basic model selection with k-Nearest Neighbors

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.

A graph with a line

Description automatically generated

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 to the minima in the loss plain.

11.

Plots:

A graph of a loss

Description automatically generated with medium confidence

A graph of a loss

Description automatically generated with medium confidence

A graph of a performance

Description automatically generated with medium confidence

A graph of a performance

Description automatically generated with medium confidence

A graph of performance and performance

Description automatically generated with medium confidence

We would choose 1e-11 as the learning rate because it’s the only rate in which the model is learning, the other models seem to miss the minima.

12.

A diagram of a train decision

Description automatically generated

