

Exercise 1: Tuning k -NN

In this exercise we will perform hyperparameter optimization (HPO) for the task of classifying the **credit risk** data with a k -NN classifier. The **kkn** implementation used by **mlr3** contains several hyperparameters, three of which are to be tuned for our prediction:

- k (number of neighbors)
 - **kernel**
 - **scale**
- a) Describe briefly the role of each hyperparameter in the learning algorithm – which effects can be expected by altering them?
- b) In **mlr3** (using the **mlr3tuning** library), define an appropriate search space to tune over. We want to explore a range between 1 and 100 for k and the kernel to be chosen from "rectangular", "epanechnikov", "gaussian", "optimal".
- c) Perform the tuning procedure using **TuningInstanceSingleCrit**. Set aside 200 test observations first. Use 5-fold cross validation and random search, and terminate the process after either 30 seconds or 200 evaluations.
- d) You realize that a high AUC is the performance measure you are actually interested in. Modify the HPO procedure such that performance is optimized w.r.t. AUC.
- e) Visualize the tuning result with a suitable command. What do you observe regarding the impact of different hyperparameters on predictive performance? What are limits of such a form of analysis?
- f) After analyzing the tuning results, you notice that changes in k are more influential for smaller neighborhoods. Re-run the HPO procedure with a log-transformation for k .
- g) With the hyperparameter configuration found via HPO, fit the model on all training observations and compute the AUC on your test data.