Department of Information Systems
Dennis Assenmacher
Dr. Pascal Kerschke
Moritz Seiler

Summer Term 2020
April 30, 2020

Data Analytics 2

– Tutorial 1 –

Exercise 1 (Performance Measures for Binary Classification)

Ten vacationers are randomly selected to be tested on influenza before boarding an airplane back to Europe. A binary classifier $\hat{f}_\theta(x_1, .., x_p) = \hat{y}$ is used to predict cases of influenza. If $\hat{y} = 1$, we have a case of influenca and if $\hat{y} = 0$ we do not. The predictions of the classifier can be found in the following table:

y_{true}	$\hat{y}$
0	0.15
0	0.29
1	0.76
0	0.39
1	0.41
0	0.18
0	0.44
0	0.09
1	0.49
0	0.31

1. Use a threshold parameter $\tau = 0.5$ to assign positive and negative cases and create the corresponding *confusion-matrix*.
2. Calculate the *accuracy*, *recall* and *precision*. Use these values to further calculate the F_1-*Score*. Why does the F_1-*Score* differ from the *accuracy*?
3. How do the F_1-*Score* and *accuracy* change if $\tau = 0.4$? Which classifier is more desirable in this scenario?

Exercise 2 (k-NN)

One frequently applied supervised learning method is called the **k-nearest neighbors algorithm** (k-NN) which can be used on quantitative as well as on categorial data. The basic idea of the algorithm is to classify an unknown observation by incorporating the information of the training data that are close to the new observation. Here the parameter k specifies how many nearest neighbors should be considered. For categorial data an unlabeled observation is classified by assigning the *most frequent* label among the k nearest neighbors. For quantitative data, the average of the neighbors is taken.

We are going to implement a naive version of the k-NN algorithm. Therefore load the dataset `NNData` provided in the learnweb (Use `load("NNData")`) and do the following:

1 Implement a function that takes two vectors and returns the Euclidean distance betweem them.

2 Implement a function `identifyNearestNeighbourIndices` that takes a set of labelled data, an unlabeled point (`p`) and the amount of nearest neighbors and returns the indices of the k-nearest neighbors of `p` (use your distance function from (1)).

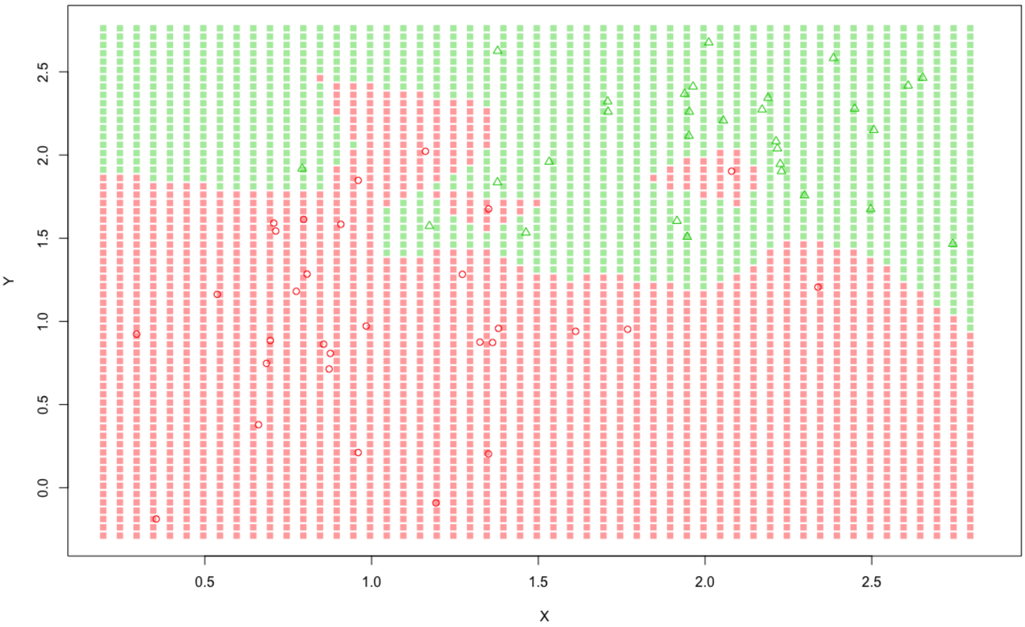
3 Implement a function called `NearestNeighbourClassification` that returns the class of the most frequent label among the k nearest neighbors of a point `p`. Utilize the function from (2).

Now we will visualize the classification areas by spanning a grid over the objective space.

4 Use the `expand.grid()` function to create a grid of points over the observation space in 0.05 steps. (If you do not know how `expand.grid` works, take a look at the documentation `help("expand.grid")`)

5 Plot the `NNData` dataset and use different symbols for different classes.

6 For each point in your grid use your k-NN function (k=1) to dertermine its class belonging. Add all points to the plot and color them appropriately (Use pch =15). The result should look similar to the Figure.



7 Use different values for the k parameter. How does it influence the classification result? Is k=*1* a good setting?

Exercise 3 (ROC & Parameter Tuning)

When it comes to binary classification problems, the ROC curve is a sophisticated performance measurement.

1. Please explain how the ROC curve is constructed. How can the algorithm's performance be assessed from the curve?
2. Load the *heart*-dataset (provided in the learnweb) and use `mlr` to create a task. Next, create a k-NN learner with k*=3* and use the resampling strategy *"Crossvalidation"* with `iters`*=10* to measure its performance. Use *accuracy* and F_1-score as measures.
3. Draw the *ROC*-curve using `plotROCCurves`. Briefly describe the graph and highlight where τ is optimal. (Note, you need to install and load `ggplot2`)
4. Familiarize yourselves with `tuneParams` (`https://mlr.mlr-org.com/articles/tutorial/tune.html`) and tune k by using *"Crossvalidation"*. How does the new model perform compared to $k = 3$?
5. Draw the *ROC*-curve for the tuned model and compare it to the *ROC*-curve of the original model with k*=3*. What do you find?