The Elements of Statistical Learning, WS 2019/2020
Jilles Vreeken and Tobias Marschall
Exercise Sheet #1: *The Basics*

Deadline: Wednesday, October 31. 2019, 10:00 a.m.
Please read and follow the following requirements to generate a valid submission. This problem set is worth 50 points. You may submit your solutions in groups of two students. The solutions to the theoretical problems should be submitted either digitally (in PDF format) to esl-ta@mpi-inf.mpg.de or as a hard copy before the lecture. **Label your hard copy submissions with your name(s), Matriculation number, and Exercise group.**
Solutions to programming problems, including resulting plots and answers to the questions, need to be submitted in digital format (PDF). For the programming problems you have to submit an R script (that means it needs to run with `Rscript your_file.R`). Write comments within the script to explain what your code is supposed to do.
For digital submissions the subject line of your email should have the following format:

[SL][problem set 1][*day of ex. group*]_lastname1_firstname1_lastname2_firstname2

where *day of ex. group* is either "Mo" or "We" depending on your assigned time slot. Please include the numbers of the problems you submitted solutions to (both digitally and analogously) in the email's body. **Please make sure that all the files are attached to the email.** The attached files should only include an R script (.R suffix) and one .pdf file with all other solutions.

Problem 1 (T, 10 Points). Describe the main principles of statistical learning in short and concise words using each of these terms:

- unsupervised learning
- supervised learning
- inputs/features/predictors/independent variables
- outcome/response/dependent variables
- quantitative variable
- qualitative/categorical variable
- classification
- regression
- training data
- test data
- prediction
- inference
- parametric models
- non-parametric models

You are allowed to use a maximum of **300 words** for this exercise, every **ten** more words will lead to losing **one** point.

Problem 2 (T, 8 Points). Let Y be a random variable. Show that

$$\mathbb{E}(Y) = \text{argmin}_c \mathbb{E}\left[(Y-c)^2\right].$$

Problem 3 (T, 12 Points). Prove the bias-variance tradeoff with irreducible error. Please note that you should prove both equalities.

$$\begin{aligned}\mathbb{E}\left[(y_0 - \hat{f}(x_0))^2\right] &= \mathbb{E}\left[(\hat{f}(x_0) - \mathbb{E}(\hat{f}(x_0)))^2\right] + \left[\mathbb{E}(\hat{f}(x_0) - f(x_0))\right]^2 + \text{Var}(\epsilon) \\ &= \text{Var}(\hat{f}(x_0)) + \left[\text{Bias}(\hat{f}(x_0))\right]^2 + \text{Var}(\epsilon).\end{aligned}$$

Note, that this is Equation (2.7) from the book, which also contains Equation (2.3).

Problem 4 (P, 20 Points). In this exercise you will predict the ozone concentration in New York from wind speed, daily maximum temperature and solar radiation.

1. Familiarize yourself with the R programming language. Go through **2.3 Lab: Introduction to R** (ISLR p. 42–51).
2. Download the dataset `ozone.RData` from the course website. (*hint:* use the `load()` command). This file contains 3 objects: `ozone` (the data table), `trainset` (the row indices for the training set) and `testset` (the row indices of the test set). Inspect the structure of the objects using `ls()`, `str()`, `summary()`, `dim()`, `length()`,`range()`, `colnames()`. Identify the column names corresponding to each of the data types mentioned in the introduction. How many observations do you have (in total, in the training set, in the testset)?
3. What is the range of each input variable? What is the mean and standard deviation of each variable? *useful functions:* `range()`, `apply()`, `mean()`, `sd()`
4. Create scatterplots for every pair of features in the dataset. Calculate the Pearson correlation coefficients for each pair of datatypes. In general, what is the range of the Pearson correlation coefficient? What does a correlation coefficient of 0 tell you about the relationship between two variables? What trends do you observe in the data according to the correlation coefficient? Can you see them directly from the plot (visually)? *useful functions:* `plot()`, `pairs()`, `cor()`
5. Implement a function `rss` that computes the Residual Sum of Squares (RSS) between a vector of predicted values and a vector of true values. *See 3.6. Lab, section 3.6.7 (ISLR p.119)* for how to write R functions.
6. Predict the ozone level based on radiation, temperature and wind speed using a linear regression model. Use the training set to train the model and the test set to test the model. Report the RSS as well as the correlation (Pearson) with the true responses. Create a scatterplot for the predicted and true values of ozone for the test set. *useful functions:* `lm()`, `predict.lm()`
7. Perform k nearest neighbor (kNN) regression to predict the ozone level from the other features. Use $k = 1, 2, \ldots, 30$. Plot the RSS for training and test set respectively for every k. On which side of the graph do you have the most complex models? Argue with the bias-variance tradeoff. Which value of k would you choose for this data? In general, does the kNN method make any assumptions on the underlying data distribution? *useful functions:* `knn.reg()` from the `FNN` package, `which()`, `order()`
8. Compare the RSS and correlation of the linear model and your chosen nearest neighbor model. Which one would you prefer over the other for this example? Consider model complexity (degrees of freedom), model assumptions and prediction quality.