Presenting groups: 5 and 6, Date: 2.6.2021

Exercise 1:

In this exercise, you will continue working with the `Auto` data set which can be downloaded at `https://archive.ics.uci.edu/ml/machine-learning-databases/auto-mpg/`. As in the lectures, we will study the relationship between car's fuel efficiency and its characteristics. However, this time we will develop a model to predict whether a given car gets "high" or "low" gas mileage.

a) Create a binary variable, say, `mpg1`, that is equal to 1 if `mpg` contains a value above its median, and to 0 if `mpg` contains a value below its median.

b) Explore the data. Which of the variables seem most likely to be useful in predicting `mpg1`? Justify your findings (you can do it either verbally or graphically).

c) Split the data into a training set and a test set that have the same size.

d) Perform logistic regression on the training data in order to predict `mpg1` using different sets of variables. What are the training errors and the test errors of the model obtained?

e) Perform k-nearest neighbours on the training data, with several values of k, in order to predict `mpg1` using only the variables that you selected in (b). What are the training errors and the test errors of the model obtained?

f) Investigate the fit of the model in d) and e) using LOOCV and 5-fold cross-validation. Plot the results together with the training errors and the test errors from d) and e) on one plot. Based on the obtained results, which set of the variables would you choose for the logistic regression and which k would you pick in the kNN approach?

Exercise 2 (Simulation Study)**:**

In this exercise, you need to design a simulation study that will illustrate the bias-variance trade-off between LOOCV and k-fold cross-validation. You can use the binary model that we studied in class as a starting point:

$$\mathbb{P}(Y_i = y_i | \, \mathbf{X} = \mathbf{x}_i) = p(\mathbf{x}_i)_i^{y_i}(1 - p(\mathbf{x}_i))^{1-y_i}$$

where

$$p(\mathbf{x}_i) = \frac{exp\{\mathbf{x}_i'\boldsymbol{\beta}\}}{1 + exp\{\mathbf{x}_i'\boldsymbol{\beta}\}}.$$

The DGP has the following parameters: $n = 1000$, $\beta_1 = -2$, $\beta_2 = 0.1$, $\beta_3 = 1$, X_1 is a constant, $X_2 \sim \mathcal{U}(18, 60)$ and $X_3 \sim \mathcal{B}(0.5)$. In the lecture we estimated $\beta_1, \beta_2, \beta_3$ via maximum likelihood. Remember that in the simulation study you *know* the true test error rate and you can use it as a benchmark.

Hint: varying the sample size can be a good idea.

You do not have to program the `glm` and `kNN` functions yourself (although you may, of course). Some helpful packages, libraries and commands:

```
library(class) ###required to use knn command below
library(boot) ###required to use cv.glm command below

glm(..., family = binomial) ###fits generalized linear models including logistic regression
cv.glm() ###automatically computes the cross-validation estimates for glm

knn(train, test, train.classifiers, k, ...) ###performs kNN, see help-file
```