Coding Assignment 4

Due Date: *Monday, November 10 (11.59pm)*

Submission: on Gradescope

Instructions:

The coding assignments may be completed in groups up to 4 students. If you do so, please make sure that you include everyone's full name, and that you *also select everyone's name when submitting the assignment on Gradescope*. This ensures that each group member will get a grade assigned and have access to the comments from the graders.

For this assignment, the following items need to be submitted:
(1) the **code** in R or *Python* (a Markdown file is ok; a Jupyter notebook is ok)
(2) a **pdf** file with your code and results with all necessary plots and comments.

Problem 1 (30 points)

In this problem we are going to work with the `Wine` data set found at the UCI repository here: https://archive.ics.uci.edu/dataset/109/wine. This data set contains the results of a chemical analysis of 178 wine samples grown in the same region of Italy but derived from three different grape cultivars (classes). Each sample is described by 13 continuous variables, including alcohol content, malic acid, ash, magnesium, flavonoids, color intensity, and other chemical properties.

You can download the data directly from the UCI repository, or load it as a part of the `sklearn` library (`wine`) in python or the `mlbench` library (`Wine`) in R.

The goal of this exercise is to compare Discriminant Analysis methods (linear and quadratic) vs. a Logistic/Multinomial approach to correctly classify the *three* different wine cultivars. For the questions below, split the data into training and testing (70% - 30%) using 598 as a seed.

(a) Using the training data fit *three* models: **LDA**, **QDA** and **Multinomial** (Logistic) **regression**.
Note that here the response has three levels, so a binomial logistic is not applicable.

(b) Report the accuracy for all methods on both training and testing data sets, and prepare confusion matrices.

(c) Comment on the results. For example, discuss the following: Is there a model that seems to perform better overall? Which model misclassified more observations? Is there a class that performed better or worse (in terms of classification)?

Problem 2 (70 points)

In this exercise, we are going to re-visit a familiar data set from Week 1: the UCI digits recognition data set found here: https://doi.org/10.24432/C5MG6K.

In this iteration, we are going to work with all the digits (no need to filter out specific ones) and the goal is to compare the performance of SVM, Decision Trees and Boosting methods, such as AdaBoost, Gradient Boosting, XGBoost for correctly classifying all 10 digits. For the analysis below, please split the data into training and testing (70% - 30%) using 598 as a seed.

(a) Fit a **SVM** classifier with a **Gaussian kernel** (also known as radial basis function) on the training data set.

(b) Fit a **Decision Tree** classifier on the training data set.

(c) Choose two boosting algorithms among **AdaBoost**, **Gradient Boosting**, or **XGBoost** and fit a classifier on the training data set.

(d) Report the accuracy for all classifiers fitted in (a), (b), (c) in both training and testing data.

(e) Report the confusion matrix for the test predictions.

(f) Comment on the results. Which are the digits that seem to be most commonly confused? Did you have any overfitting issues with any of the approaches?

Remark: The results might be different between Python and R. That is expected and ok.