This assignment makes you more familiar with decision trees and random forests. You will apply the concepts from the lecture, implement and experiment with them to see decision and regression trees *in action*. The assignment consists of two IPython notebooks and some questions to be answered with pen and paper. Those questions at the end are supposed to prepare you for the final exam as well as encourage you to think about some properties of trees and forests.

1. **Simple Regression Trees**
   In this exercise, you implement a regression tree and a random forest for a one dimensional problem. With that implementation, you explore the impact of some of the important hyperparameters.

   (a) Work through the 'regression_trees.ipynb' notebook found on ILIAS.

   (b) What is the computational complexity of the decision tree you implemented? To ease the notation, assume that N, the number of data points, is a power of 2, every split halves the observations exactly, and that $min\_leaf = 1$. In the exercise, you didn't use all the tricks one can use. How does the complexity change if the best loss for N data points could be computed in $\mathcal{O}(N)$?

2. **Decision Trees and Ensembles**
   Here, you explore ensembling via bagging for different types of trees and study the out-of-bag error, a useful estimate of the validation error that comes at low cost when bagging is used.

   (a) Work through the 'tree_ensembles.ipynb' notebook found on ILIAS.

3. **The CART Algorithm**
   This question aims to give you a better understanding of the algorithm used to *grow* decision trees.

   (a) Explain *in words* the CART algorithm for growing a classification tree as presented in the lecture. Give the definition of Entropy, expressed in terms of $p(v_k)$ for $k = 1, \ldots, K$ (see also the slides).

   (b) How would you handle the following three distinct cases that can happen when deciding on a splitpoint in a node:
   - All instances have the same class label, but various attribute values
   - All instances have the same value for all attributes, but various class labels
   - The instances have various class labels and various attribute values

   (c) The table below shows information from previous computer sales. We want to to build a decision tree that for any new computer determines whether we should buy it. Execute the decision tree algorithm by hand. Use the definition of Entropy and Information Gain.

nr	memory	processor	rest	buy
1	much	fast	bad	yes
2	much	slow	bad	no
3	few	fast	good	yes
4	few	fast	bad	no

   (d) Optimizing Entropy or Information Gain for every split does not always lead to the *best* tree. Can you find a perfectly balanced tree that has the same accuracy as the one in part (c)?

4. **Majority Voting and Class Probabilities**
   Show that for a binary classification problem with n_0 (n_1) observations of class 0 (1) in a given leaf:

   (a) majority voting minimizes the absolute error over the set of examples in that leaf

   (b) the class probabilities $p_0 = n_0/(n_0 + n_1)$ and $p_1 = n_1/(n_0 + n_1)$ minimize the sum of squared error.

5. **Number of Out-of-Bag Samples**
   In the lecture, we mentioned that almost 37% of the data points do not occur in a given bootstrap sample. This question will show you why. Consider a bootstrap sample of size N where N is also the number of data points, i.e. we draw a sample the same size as the original data, but we pick data points randomly with replacement.

(a) What is the probability p_i that any given data point is NOT in a bootstrap sample?

(b) As this probability is the same for all data points, and the random sampling is independent, the expected number of unique data points in the bootstrap sample will be $\mathbb{E}[N_{unique}] = N(1 - p_i)$. What is the limit $N \to \infty$ of the fraction N_{unique}/N? Hint: You can use

$$\mathrm{e}^x = \lim_{N\to\infty} \left(1 + \frac{x}{N}\right)^N$$