Elements of Machine Learning, WS 2022/2023
Jilles Vreeken and Aleksandar Bojchevski
Exercise Sheet #6: *Trees and Forests and Supports etc.*

Deadline: Thursday, February 2, 2023, 16:00
Before solving the exercises, read the instructions on the course website.

- For each theoretical problem, submit a single `pdf` file that contains your answer to the respective problem. This file may be a scan of your (legible) handwriting.
- For each practical problem, submit a single `zip` file that contains
  - the completed jupyter notebook (`.ipynb`) file,
  - any necessary files required to reproduce your results, and
  - a `pdf` report generated from the jupyter notebook that shows all your results.
- For the bonus question, submit a single `zip file` that contains
  - a `pdf` file that includes your answers to the theoretical part,
  - the completed jupyter notebook (`.ipynb`) file for the practical component,
  - any necessary files required to reproduce your results, and
  - a `pdf` report generated from the jupyter notebook that shows your results.
- Every team member has to submit a signed Code of Conduct.

Problem 1 (T, 5 Points). **Interpretability**
In the lecture, you have seen the term "interpretability" come up to describe certain models.

(a) (1 Point) Describe what we mean by interpretability.

(b) (3 Points) Rank the following methods by their interpretability and explain your reasoning.

- Ridge Regression
- LASSO
- Generalized Linear Models
- Neural Networks
- Decision Trees
- Random Forests

(c) (1 Point) Let M be some model which you would have rated as highly *non*-interpretable in the previous exercise. You learn that every such model M can be equivalently represented by a decision tree. Does this change your opinion of the interpretability of M? If so, why? If not, why not?

Solution.

(a) An interpretable model is one in which it is possible for a *human* to *understand* how the model makes its predictions, and can therefore check whether the model makes sense or not. In contrast, an uninterpretable model is one in which it is almost impossible for a human to make any sense of its outputs.

(b) From most to least interpretable: LASSO $\geq$ Decision Trees $\geq$ Ridge Regression $>$ GLMs $\geq$ Random Forests $>$ Neural Networks.

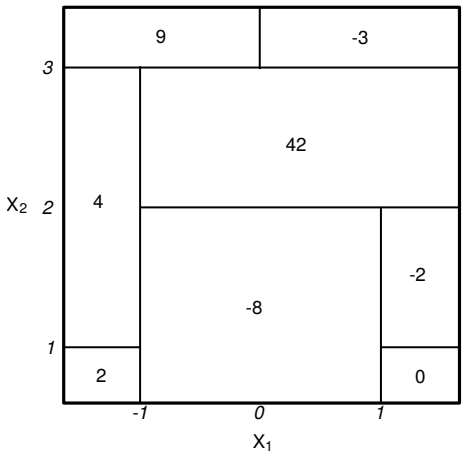
- LASSO is highly interpretable due to its additivity and sparsity, meaning that most variables won't have any influence on the output.

- Decision Trees too are highly interpretable because to understand a prediction, one can simply go down the correct branch of the tree.
- Ridge Regression is additive like LASSO but has the problem that the effects of correlated variables cannot be easily disentangled.
- GLMs generally have the same issues as Ridge Regression, but also add a nonlinearity to the model, making it even more difficult to interpret the model.
- Random Forests are difficult to interpret because each tree uses a different subset of the weights and then all models are averaged. To understand what the model is doing for a given input, one would have to first go through *every* tree and then interpret the average in some way.
- Neural Networks have so many interactions between all variables that understanding the effect of any one variable is enormously difficult and depends on all other variables.

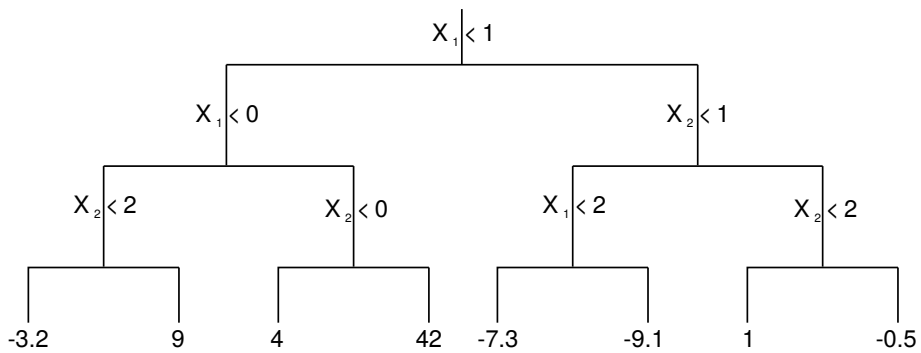
(c) If M is highly non-interpretable, then even if it can be equivalently represented by some decision tree, the issue is that this decision tree will generally be so large that it is still impossible for a human to interpret it.

Problem 2 (T, 10 Points). **Trees and Splits**

(a) (4 Points) Sketch a tree corresponding to the partition of the predictor space indicated in the figure below. The numbers inside the boxes indicate the mean of Y within each region.



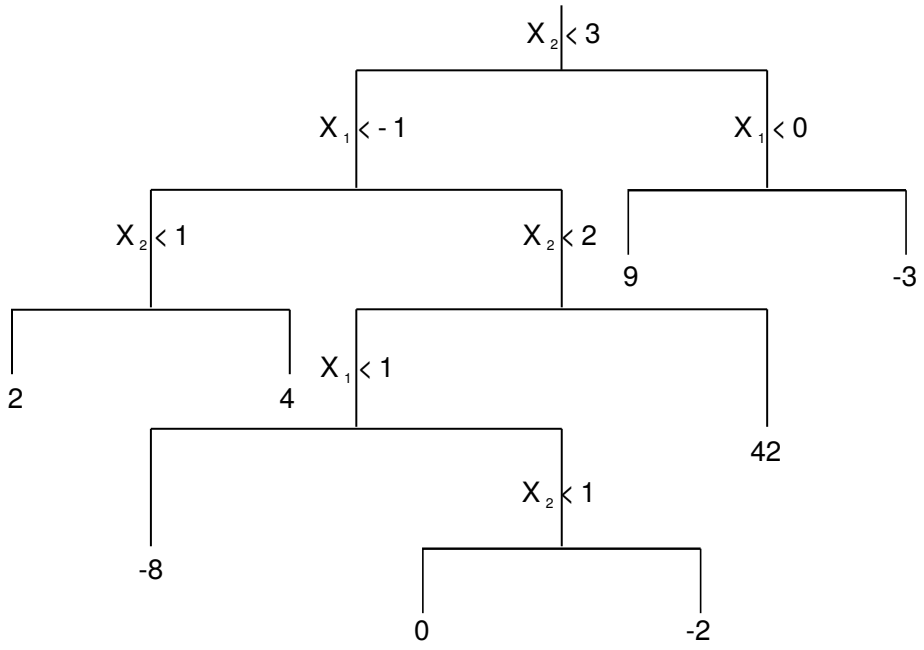
(b) (3 points) Create a diagram similar to the one provided in a), using the tree illustrated below. You should divide up the predictor space into the correct regions, and indicate the mean for each region.



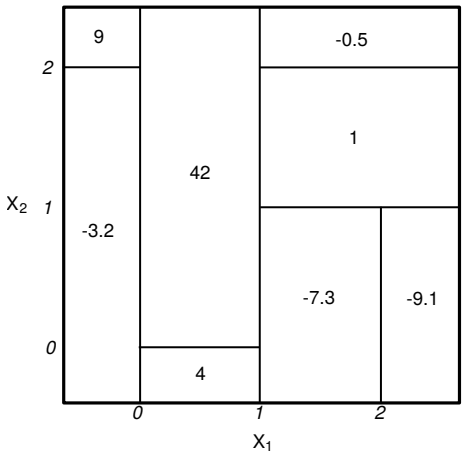
(c) (3 Points) Create another equivalent tree representing exact the same partition of the predictor space as the one discussed in b), but with *at least* a different split at the root node.

Solution.

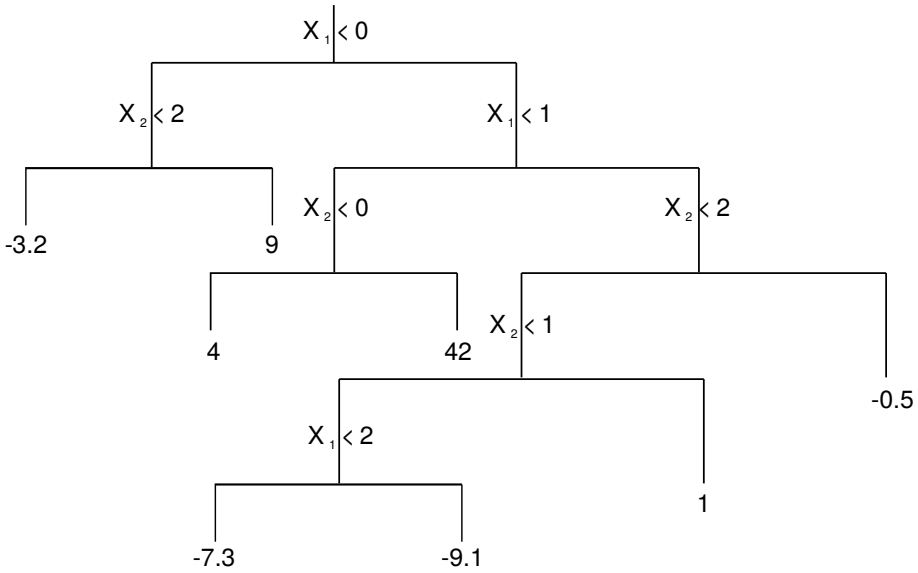
(a) Tree (note that the solution tree is **not** unique):



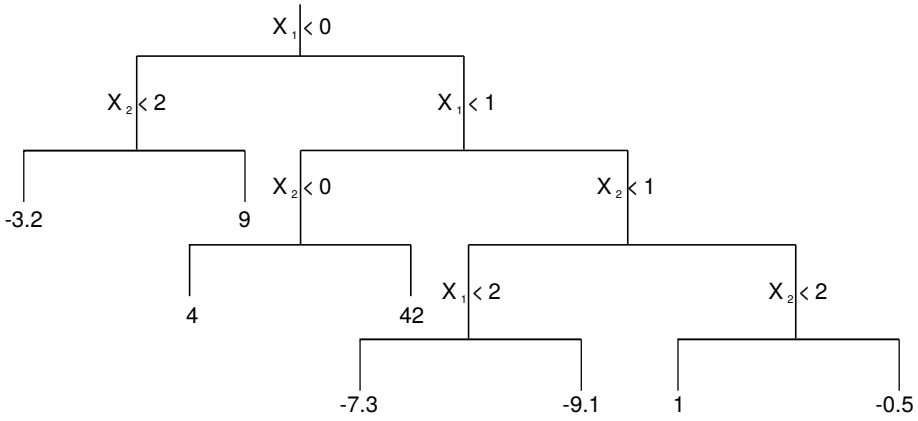
(b) Predictor Region:



(c) Tree 1:



or Tree 2:



Problem 3 (T, 10 Points). **Linear and Support Vector Regression**

(a) (4 Points) Show that the Ridge Regression problem is equivalent to

$$\min_{\beta_0,\beta,\xi_1,\ldots,\xi_N} \frac{1}{2}\|\beta\|_2^2 + C\sum_{i=1}^{N} \xi_i^2 + \tilde{\xi}_i^2$$
$$\text{subject to} \quad \xi_i, \tilde{\xi}_i \geq 0$$
$$-\tilde{\xi}_i \leq y_i - \beta_0 - \beta^\top x_i \leq \xi_i \quad \text{for } i = 1,\ldots,N$$

(b) (4 Points) Based on the above, derive the explicit loss function L minimized in Support Vector Regression for which the equivalent optimization objective is given by

$$\min_{\beta_0,\beta,\xi_1,\ldots,\xi_N} \frac{1}{2}\|\beta\|_2^2 + C\sum_{i=1}^{N} \xi_i + \tilde{\xi}_i$$
$$\text{subject to} \quad \xi_i, \tilde{\xi}_i \geq 0$$
$$-\tilde{\xi}_i - \epsilon \leq y_i - \beta_0 - \beta^\top x_i \leq \xi_i + \epsilon \quad \text{for } i = 1,\ldots,N$$

(c) (2 Point) Explain why the optimization objective involving the $2N$ additional $\xi_i, \tilde{\xi}_i$ may be preferred for SVRs.

Solution.

(a) Let $\hat{y}_i = \beta_0 + \beta^\top x_i$. Then if we have $y_i - \hat{y}_i \leq 0$, we can set $\xi_i = 0$ and $\tilde{\xi}_i = \hat{y}_i - y_i$. Conversely, for $y_i - \hat{y}_i \geq 0$ we can set $\tilde{\xi}_i = 0$ and $\xi_i = y_i - \hat{y}_i$. Overall, the sum $\xi_i^2 + \tilde{\xi}_i^2 = (y_i - \hat{y}_i)^2$ for all i so that the second sum is simply the normal sum of squared errors in ridge regression. By setting $\lambda = \frac{1}{2C}$ we obtain the standard formulation.

(b) Letting $\hat{y}_i$ be as above, we can write $L_\epsilon(y_i, \hat{y}_i) = \max(|y_i - \hat{y}_i| - \epsilon, 0)$.

Then as above, by setting $\xi_i = y_i - \hat{y}_i - \epsilon$ when this is larger than zero and zero otherwise, or $\tilde{\xi}_i = \hat{y}_i - y_i - \epsilon$ when this is larger than zero or zero otherwise, we have that $\xi_i + \tilde{\xi}_i = L_\epsilon(y_i, \hat{y}_i)$. The full loss function optimized is therefore

$$\min_{\beta_0, \beta} \sum_{i=1}^{N} L_\epsilon(y_i, \hat{y}_i) + \lambda \|\beta\|_2^2$$

where $\lambda = \frac{1}{2C}$ as before.

(c) The function $L_\epsilon(y_i, \cdot)$ is not differentiable at $y_i \pm \epsilon$. In contrast, the objective using all $\xi_i, \tilde{\xi}_i$ is differentiable everywhere in its domain.

Problem 4 (P, 15 Points). **Trees and Forests and Bags and Correlations**
In this exercise, we will study how correlated the individual trees in Bagging and in Random Forests are.

(a) (1 Point) Load the data in `train.csv` and compute the correlation of each predictor variable X_i with the target variable y.

(b) (2 Points) Use bagging to train $B = 100$ Regression Trees T^b on the data X.

(c) (3 Points) Load the data `test.csv` and compute the average correlation between the predictions $y^b_{\text{pred}} = T^b(X_{\text{test}})$ for different Trees $T^b, b = 1, \ldots, 100$.

(d) (2 Points) Similarly compute the average correlation between the residuals $y_{\text{test}} - y^b_{\text{pred}}$. Contrast the result with that derived from part (c) and explain which measure of correlation is more useful.

(e) (2 Points) For each $q \in \{0.2, 0.4, \ldots, 1\}$, train a Random Forest Regression with 100 trees on the training data, in which each tree uses only a fraction q of all available predictors.

(f) (2 Points) Recompute the correlations from (d) for each of the random forests, and plot the results in a suitable manner. Explain what you see.

(g) (3 Points) Compute the variable importances for each variable for each forest trained in (e). Plot them against the correlations computed in (a). What do you see? Why?

Note: All relevant models can be fit using `sklearn.ensemble.RandomForestRegressor`.

Problem 5 (Bonus). ¬ **Heaping**
In the lectures, you have seen model aggregation in terms of Bagging, Boosting and Random Forests. In this exercise, we try something a little different.
Let X, y be our available data, and let $f_1, \ldots, f_K$ be functions (not necessarily trees) trained to predict y from X.

(a) We want to use all models f_j by predicting y as $\hat{y} = \sum_{j=1}^{K} w_j f_j(x)$, where $\sum_{j=1}^{K} w_j = 1$ and all $w_j \geq 0$. Write down the optimization problem for learning the weights w_j.

(b) Explain what issues might arise in optimizing this objective.

(c) Unlike our above approach, a much simpler aggregation mechanism is often used, simply setting $\hat{y} = \frac{1}{K} \sum_{i=1}^{K} f_j(x)$, i.e. all $w_j = 1/K$. Explain under which conditions this would be a good choice, and under which conditions it would be a bad choice.

(d) What kind of regularizer could we add to the optimization problem from (a) to make all w_i be more similar to each other?

(e) In the above, we implicitly assumed that the same weights w_i are suitable for all x. Explain under which conditions this would not be suitable.

(f) How would you learn appropriate weights $w_j(x)$ depending on x? What would you need to know about the functions f_j to be able to do this?

(g) Explain how this approach differs from Boosting.