

BACS - HW (Week 17)

This week, we will look at a dataset of US health insurance premium charges. We will build models that can *predict* what someone's insurance charges might be, given several factors about them. You download the dataset, and find more information about it, at the Kaggle platform where machine learning people like to host challenges and share datasets: <https://www.kaggle.com/datasets/teertha/ushealthinsurancedataset>

Setup: Download the data, load it in your script, and omit any rows with missing values (NAs)

Note: The values of charges are large, so MSE values will be very large. This week let's use [RMSE](#), or the Root-Mean-Square Error, so we have smaller numbers. Simply compute it as the square-root of MSE.

Question 1) Create some explanatory models to learn more about charges:

- a. Create an OLS regression model and report which factors are significantly related to charges
- b. Create a decision (regression) tree with default parameters
 - i. Plot a visual representation of the tree
 - ii. How *deep* is the tree (see nodes with "decisions" – ignore the leaves at the bottom)
 - iii. How many leaf groups does it suggest to bin the data into?
 - iv. What is the average charges of each leaf group?
 - v. What conditions (decisions) describe each group?

Question 2) Let's use LOOCV to see how our models perform predictively

- a. What is the $RMSE_{\text{OOS}}$ for the OLS regression model?
- b. What is the $RMSE_{\text{OOS}}$ for the decision tree model?

For bagging and boosting, we will only use split-sample testing to save time: partition the data to create training and test sets using an 80:20 split. Use the regression model and decision tree you created in Question 1 for bagging and boosting.

Question 3) Let's see if bagging helps our models

- a. Write `bagged_learn(...)` and `bagged_predict(...)` functions using the hints in the class notes and help from your classmates on Teams. Feel free to share your code generously on Teams to get feedback, or ask others for help.
- b. What is the $RMSE_{\text{OOS}}$ for the bagged OLS regression?
- c. What is the $RMSE_{\text{OOS}}$ for the bagged decision tree?

(see next page)

Question 3) Let's see if boosting helps our models. You can use a learning rate of 0.1 and adjust it if you find a better rate.

- a. Write `boosted_learn(...)` and `boosted_predict(...)` functions using the hints in the class notes and help from your classmates on Teams. Feel free to share your code generously on Teams to get feedback, or ask others for help.
- b. What is the $RMSE_{\text{oos}}$ for the boosted OLS regression?
- c. What is the $RMSE_{\text{oos}}$ for the boosted decision tree?

Question 4) Let's engineer the best predictive decision trees. Let's repeat the bagging and boosting decision tree several times to see what kind of base tree helps us learn the fastest. Report the $RMSE_{\text{oos}}$ at each step.

- a. Repeat the bagging of the decision tree, using a base tree of maximum depth 1, 2, ... n while the $RMSE_{\text{oos}}$ keeps dropping; stop when the $RMSE_{\text{oos}}$ has started increasing again.
- b. Repeat the boosting of the decision tree, using a base tree of maximum depth 1, 2, ... n while the $RMSE_{\text{oos}}$ keeps dropping; stop when the $RMSE_{\text{oos}}$ has started increasing again.