Homework 9

For this homework you will create an R Markdown file and output (HTML file) and upload both to wolfware. Be sure to include text explaining your thought process/what you are doing with your questions.

The purpose of this homework is to get practice fitting kNN and tree based methods using the `caret` package.

We'll use the data set in the assignment link called `heart.csv`. This data set gives information about whether or not someone has heart disease (`HeartDisease = 1 or = 0`) along with different measurements about that person's health. The data comes from here if you'd like to read a bit more about it.

Part 1: kNN

The article here gives a great example of selecting the number of neighbors to use with the `caret` package. They use repeated 10 fold cross-validation. Although computationally intensive, doing repeated CV helps to give a more stable prediction of CV error. This is similar to how a mean is less variable than a single value. Since there is some inherent randomness in doing a CV computation, we can get an overall more stable result by averaging!

Tasks

Please do the following:

1. Read in the `heart.csv` data file.
2. Create a new variable that is a factor version of the `HeartDisease` variable (if needed, this depends on how you read in your data). Remove the `ST_Slope` variable and the original `HeartDisease` variable (if applicable).
3. We want to use kNN to predict whether or not someone has heart disease. To use kNN we generally want to have all numeric predictors (although we could try to create our own loss function as an alternative). In this case we have some categorical predictors still in our data set: `Sex`, `ChestPainType`, and `RestingECG`

Create dummy columns corresponding to the values of these three variables for use in our kNN fit. The caret vignette has a function to help us out here. You should use `dummyVars()` and `predict()` to create new columns. Then add these columns to our data frame and remove the original columns from which these variables were created.

4. Now split the data set you've created into a training and testing set. Use `p = 0.8`.
5. Finally, train the kNN model. Use repeated 10 fold cross-validation, with the number of repeats being 3. You should also preprocess the data by centering and scaling. Lastly, set the `tuneGrid` so that you are considering values of `k` of 1, 2, 3, ..., 40. (Note: From the help for the `train()` function it says: `tuneGrid` A data frame with possible tuning values. The columns are named the same as the tuning parameters. The name of the tuning parameter here is `k`.)
6. Check how well your model does on the test set using the `confusionMatrix()` function.

Part 2: Ensemble

We'll look at predicting the same heart disease variable in this section as well, just instead of using kNN we'll use the following methods:

- a classification tree (use `method = rpart`: tuning parameter is `cp`, use values 0, 0.001, 0.002, ..., 0.1)
- a bagged tree (use `method = treebag`: no tuning parameter)
- a random forest (use `method = rf`: tuning parameter is `mtry`, use values of 1, 2, ..., 15
- a boosted tree (use `method = gbm`: tuning parameters are `n.trees`, `interaction.depth`, `shrinkage`, and `n.minobsinnode`, use all combinations of `n.trees` of 25, 50, 100, 150, and 200, `interaction.depth` of 1, 2, 3, 4, `shrinkage = 0.1`, and `nminobsinnode = 10`; Hint: use `expand.grid()` to create your data frame for `tuneGrid`)

Tasks

Using the training data you created above to fit each model (using repeated CV as above but just 5 fold for computational ease). Test the model by finding the confusion matrix on the test data.