Model Comparison

For this homework you will create a github repo, clone the repo to your computer as an R project, create a `.qmd` file, and practice working using a proper github workflow. You'll submit a pdf to Gradescope.

Your submission should include both the code and corresponding output/text that answers the question.

Commit and push your changes (at a minimum) after each task you complete.

Note: There is a 24 hour late window, in which 10% will be deducted. We understand that you are busy/life happens. Please take advantage of this window if needed.

Step 1

- Head to github and create a new repo.
  - Be sure to make the repo puble and **do not** choose a `.gitignore`

Step 2

- Create a new R project from version control (as we did in the notes/videos) that clones this repository locally.
  - Recall you can click on the green button on the github.com repo website to copy the repo link.
  - A `.gitignore` file may be created in this process. That isn't a worry!

Step 3

- Create a new `.qmd` document that outputs to PDF. You can give this a title about programming in Base R. Save the file in the main repo folder.
- In this document, answer the questions below.

Outside of updating your YAML, please follow the instructions below for proper formatting.

Please recreate the Task section headers in your .qmd by using two `#`, followed by the header text (ex. Tast 1: Conceptual Questions).

For each question within the task, please put three `#`, followed by the question number (ex. Question 1).

Task 1: Conceptual Questions

On the exam, you'll be asked to explain some topics. How about some practice?! Create a markdown list with the following questions. That is, type each question as part of a bulleted list and answer each question underneath each bullet.

1. What is the purpose of using cross-validation when fitting a random forest model?
2. Describe the bagged tree algorithm.
3. What is meant by a general linear model?
4. When fitting a multiple linear regression model, what does adding an interaction term do? That is, what does it allow the model to do differently as compared to when it is not included in the model?
5. Why do we split our data into a training and test set?

Task 2: Data Prep

First, create a sub-header titled `packages and data` and library the `tidyverse`, `tidymodels`, `caret`, and the `yardstick` package. We will need these our homework. In the code chunk setting, suppress all messages associated with calling these packages.

We'll use the data set called `heart.csv` available here. 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. In the same code chunk, read in your data set as a tibble.

1. Run and report `summary()` on your data set. Then, answer the following questions:

    a. What type of variable (in R) is Heart Disease? Categorical or Quantitative?
    b. Does this make sense? Why or why not.

2. Change `HeartDisease` to be the appropriate data type, and name it something different. In the same tidyverse pipeline, remove the `ST_Slope` variable and the original `HeartDisease` variable. Save your new data set as `new_heart`. We will use this new data set for the remainder of the assignment.

Task 3: EDA

1. We are going to model someone's age (our response variable) as a function of heart disease and their max heart rate. First, create the appropriate scatterplot to visualize this relationship. Add a line to the scatterplot that represents if someone has or does not have heart disease. Remove the standard error bars from the lines and add appropriate labels. Also, change the color pallet to be more colorblind friendly.

2. Based on visual evidence, do you think an interaction model or an additive model is more appropriate? Justify your answer.

Task 4 Testing and Training

Split your data into a training and test set. (Ideally you'd do this prior to the EDA so that info from the EDA doesn't bias what you do modeling-wise, but that isn't usually done.)

Set your seed to be 101 before performing your random process. Name your data sets `test` and `train`. Perform an 80-20 split. You do not have to do this "by hand" like in the code along (but you are more than welcome to do so).

Task 5: OLS and LASSO

1. Regardless of your answer in Task 3, we are going to fit an interaction model. First fit an interaction model (named ols_mlr) with age as your response, and max heart rate + heart disease as your explanatory variables using the training data set using ordinary least squares regression. Report the summary output.

2. We are going to use RMSE to evaluate this model's predictive performance on new data. Test your model on the testing data set. Calculate the residual mean square error (RMSE) and report it below.

3. Now, we are going to see if a model fit using LASSO has better predictive performance than with OLS. We are going to use cross validation to select the best tuning parameter, and then evaluate our LASSO model on the testing data set and compare RMSEs.

Let's use a 10 fold VC to choose our tuning parameter. Set up your correct "LASSO recipe", and be sure to standardize your predictors. Name this set up `LASSO_recipe` below. Note, fitting an interaction model using LASSO regression has slightly different syntax than when just using `lm`. We are going to learn how to do this together in this assignment. In your LASSO recipe…

a) in the `recipe()` function, use a + to separate your explanatory variables, regardless of the fact that this will be an interaction model
b) standardize your variables as normal
c) after you standardize your variables, add an additional pipe into the function `step_interact()`.
d) within `step_interact()`, start with a `~`, and then put variable_name_1:starts_with("variable_name_2_

We have to do this because of the way we standardize our new Heart Disease variable. R transforms our original column into dummy named columns with our variable name followed by a _level.

Print your LASSO_recipe by adding LASSO_recipe at the end of your code chunk.

4. Now, set up your appropriate spec, and grid. Next, select your final model, and report the results using the `tidy()` function around your model name.

5. Without looking at the RMSE calculations, would you expect the RMSE calculations to be roughly the same or different? Justify your answer using output from your LASSO model.

6. Now compare the RMSE between your OLS and LASSO model and show that the RMSE calculations were roughly the same.

7. Why are the RMSE calculations roughly the same if the coefficients for each model are different?

Task 6: Logistic Regression

Propose two different logistic regression models with heart disease as our response. Note: You don't have to use the dummy columns you made here as the `glm()` function (and the `caret` implementation of it) can handle factor/character variables as predictors.

Fit those models on the training set, using repeated CV.

Identify your best performing model. Justify why this is your best performing model and provide a basic summary of it.

2. Lastly, check how well your chosen model does on the test set using the `confusionMatrix()` function.

3. Next, identify the values of sensitivity and specificity, and interpret them in the context of the problem.

Great work! Make sure to check over your PDF before submitting it to Gradescope!