Code ▾

Homework 4

PSTAT 131/231

Resampling

For this assignment, we will be working with **two** of our previously used data sets – one for classification and one for regression. For the classification problem, our goal is (once again) to predict which passengers would survive the Titanic shipwreck. For the regression problem, our goal is (also once again) to predict abalone age.

Load the data from `data/titanic.csv` and `data/abalone.csv` into *R* and refresh your memory about the variables they contain using their attached codebooks.

Make sure to change `survived` and `pclass` to factors, as before, and make sure to generate the `age` variable as `rings` + 1.5!

Remember that you'll need to set a seed at the beginning of the document to reproduce your results.

Section 1: Regression (abalone age)

Question 1

Follow the instructions from Homework 2 to split the data set, stratifying on the outcome variable, `age`. You can choose the proportions to split the data into. Use *k*-fold cross-validation to create 5 folds from the training set.

Set up the same recipe from Homework 2.

Question 2

In your own words, explain what we are doing when we perform *k*-fold cross-validation:

- What **is** *k*-fold cross-validation?
- Why should we use it, rather than simply comparing our model results on the entire training set?
- If we split the training set into two and used one of those two splits to evaluate/compare our models, what resampling method would we be using?

Question 3

Set up workflows for three models:

1. *k*-nearest neighbors with the `kknn` engine, tuning `neighbors`;
2. linear regression;
3. elastic net **linear** regression, tuning `penalty` and `mixture`.

Use `grid_regular` to set up grids of values for all of the parameters we're tuning. Use values of `neighbors` from 1 to 10, the default values of penalty, and values of mixture from 0 to 1. Set up 10 levels of each.

How many models total, **across all folds**, will we be fitting to the **abalone data**? To answer, think about how many folds there are, how many combinations of model parameters there are, and how many models you'll fit to each fold.

Question 4

Fit all the models you created in Question 3 to your folded data.

*Suggest using `tune_grid()`; see the documentation and examples included for help by running `?tune_grid`. You can also see the code in **Lab 4** for help with the tuning process.*

Question 5

Use `collect_metrics()` to print the mean and standard errors of the performance metric ***root mean squared error (RMSE)*** for each model across folds.

Decide which of the models has performed the best. Explain how/why you made this decision. Note that each value of the tuning parameter(s) is considered a different model; for instance, KNN with $k = 4$ is one model, KNN with $k = 2$ another.

Question 6

Use `finalize_workflow()` and `fit()` to fit your chosen model to the entire **training set**.

Lastly, use `augment()` to assess the performance of your chosen model on your **testing set**. Compare your model's **testing** RMSE to its average RMSE across folds.

Section 2: Classification (Titanic survival)

Question 7

Follow the instructions from Homework 3 to split the data set, stratifying on the outcome variable, `survived`. You can choose the proportions to split the data into. Use *k*-fold cross-validation to create 5 folds from the training set.

Question 8

Set up the same recipe from Homework 3 – but this time, add `step_upsample()` so that there are equal proportions of the `Yes` and `No` levels (you'll need to specify the appropriate function arguments). *Note: See Lab 5 for code/tips on handling imbalanced outcomes.*

Question 9

Set up workflows for three models:

1. *k*-nearest neighbors with the `kknn` engine, tuning `neighbors`;
2. logistic regression;
3. elastic net **logistic** regression, tuning `penalty` and `mixture`.

Set up the grids, etc. the same way you did in Question 3. Note that you can use the same grids of parameter values without having to recreate them.

Question 10

Fit all the models you created in Question 9 to your folded data.

Question 11

Use `collect_metrics()` to print the mean and standard errors of the performance metric ***area under the ROC curve*** for each model across folds.

Decide which of the models has performed the best. Explain how/why you made this decision.

Question 12

Use `finalize_workflow()` and `fit()` to fit your chosen model to the entire **training set**.

Lastly, use `augment()` to assess the performance of your chosen model on your **testing set**. Compare your model's **testing** ROC AUC to its average ROC AUC across folds.

Required for 231 Students

Consider the following intercept-only model, with $\epsilon \sim N(0, \sigma^2)$:

$$Y = \beta + \epsilon$$

where β is the parameter that we want to estimate. Suppose that we have n observations of the response, i.e. $y_1, \ldots, y_n$, with uncorrelated errors.

Question 13

Derive the least-squares estimate of β.

Question 14

Suppose that we perform leave-one-out cross-validation (LOOCV). Recall that, in LOOCV, we divide the data into n folds.

Derive the covariance between $\hat{\beta}^{(1)}$, or the least-squares estimator of β that we obtain by taking the first fold as a training set, and $\hat{\beta}^{(2)}$, the least-squares estimator of β that we obtain by taking the second fold as a training set?