

# Lesson Summary

Congratulations! You have completed this lesson. At this point in the course, you know:

- How to split your data using the `train_test_split()` method into training and test sets. You use the training set to train a model, discover possible predictive relationships, and then use the test set to test your model to evaluate its performance.
- How to use the generalization error to measure how well your data does at predicting previously unseen data.
- How to use cross-validation by splitting the data into folds where you use some of the folds as a training set, which we use to train the model, and the remaining parts are used as a test set, which we use to test the model. You iterate through the folds until you use each partition for training and testing. At the end, you average results as the estimate of out-of-sample error.
- How to pick the best polynomial order and problems that arise when selecting the wrong order polynomial by analyzing models that underfit and overfit your data.
- Select the best order of a polynomial to fit your data by minimizing the test error using a graph comparing the mean square error to the order of the fitted polynomials.
- You should use ridge regression when there is a strong relationship among the independent variables.
- That ridge regression prevents overfitting.
- Ridge regression controls the magnitude of polynomial coefficients by introducing a hyperparameter,  $\alpha$ .
- To determine  $\alpha$ , you divide your data into training and validation data. Starting with a small value for  $\alpha$ , you train the model, make a prediction using the validation data, then calculate the R-squared and store the values. You repeat the value for a larger value of  $\alpha$ . You repeat the process for different  $\alpha$  values, training the model, and making a prediction. You select the value of  $\alpha$  that maximizes R-squared.
- That grid search allows you to scan through multiple hyperparameters using the Scikit-learn library, which iterates over these parameters using cross-validation. Based on the results of the grid search method, you select optimum hyperparameter values.
- The `GridSearchCV()` method takes in a dictionary as its argument where the key is the name of the hyperparameter, and the values are the hyperparameter values you wish to iterate over.

