Model Tuning and Pipelines - Introduction

[(https://github.com/learn-co-curriculum/dsc-tuning-pipelines-intro)](https://github.com/learn-co-curriculum/dsc-tuning-pipelines-intro) [(https://github.com/learn-co-curriculum/dsc-tuning-pipelines-intro/issues/new/choose)](https://github.com/learn-co-curriculum/dsc-tuning-pipelines-intro/issues/new/choose)

Introduction

Now that you have learned the basics of a supervised learning workflow, it's time to get into some more-advanced techniques! In this section you'll learn about tools for tuning model hyperparameters, building pipelines, and persisting your trained model on disk.

Tuning Model Hyperparameters with GridSearchCV

With non-parametric models such as decision trees and k-nearest neighbors, you have seen that there are various hyperparameters that you can specify when you instantiate the model. For example, the maximum depth of the tree, or the number of neighbors. Often these hyperparameters help to balance the bias-variance trade-off between underfitting and overfitting and are important for finding the optimal model.

With so many different hyperparameter combinations to try out, it can be difficult to write clean, readable code. Fortunately there is a tool from scikit-learn called `GridSearchCV` that is specifically designed to search through a "grid" of hyperparameters! In this section we'll introduce how to use this tool.

Machine Learning Pipelines

Pipelines are extremely useful for allowing data scientists to quickly and consistently transform data, train machine learning models, and make predictions.

By now, you know that the data science process is a flow of activities, from inspecting the data to cleaning it, transforming it, running a model, and discussing the results. Wouldn't it be nice if there was a streamlined process to create nice machine learning workflows? Enter the `Pipeline` class in scikit-learn!

In this section, you'll learn how you can use a pipeline to integrate several steps of the machine learning workflow. Additionally, you'll compare several classification techniques with each other, and integrate grid search in your pipeline so you can tune several hyperparameters in each of the machine learning models while also avoiding data leakage.

Pickle and Model Deployment

So far, as soon as you shut down your notebook kernel, your model ceases to exist. If you wanted to use the model to make predictions again, you would need to re-train the model. This is time-consuming and makes your model a lot less useful.

Luckily there are techniques to *pickle* your model -- basically, to store the model for later, so that it can be loaded and can make predictions without being trained again. Pickled models are also typically used in the context of model deployment, where your model can be used as the backend of an API!

Summary

This section only scratches the surface of the advanced modeling tools you might use as a data scientist. Get ready to optimize your workflow and get beyond the basics!

How do you feel about this lesson?

👍 👎

Have specific feedback?

[Tell us here! (https://github.com/learn-co-curriculum/dsc-tuning-pipelines-intro/issues/new/choose)](https://github.com/learn-co-curriculum/dsc-tuning-pipelines-intro/issues/new/choose)