Microsoft: DAT210x Programming with Python for Data Science

Bookmarks

- Start Here
- 1. The Big Picture
- 2. Data And Features
- 3. Exploring Data
- 4. Transforming Data
- 5. Data Modeling
- **6. Data Modeling II**

Lecture: SVC
Quiz

Lab: SVC
Lab

Lecture: Decision Trees
Quiz

Lab: Decision Trees
Lab

6. Data Modeling II > Lecture: Random Forest > Knowledge Checks

Bookmark

Review Question 1

(1 point possible)

How would you explain out-of-bag samples to someone who's studied decision trees but not yet random forest?

○ The out-of-bag samples are those samples withheld from the forest ensemble while training.

○ The out-of-bag samples are the bootstrapped samples used for training your decision trees

○ The out-of-bag samples are the bootstrapped samples used for training and evaluating the accuracy score of your random forest

● The out-of-bag samples are those training samples withheld from a particular decision tree while training ✓

○ The out-of-bag samples are those testing samples withheld from the ensemble so they can be used to test its accuracy

Lecture: Random Forest
Quiz

Dive Deeper

You have used 1 of 2 submissions

Review Question 2

(1/1 point)
After telling your cousins you're taking a data science course that deals with machine learning, they get super excited and put together a fictitious dataset for you to run classification on to prove you know what you're doing.

You decide to model their dataset using random forest, since that's the chapter you just studied. What's unsettling is that they didn't give you a lot of samples, but the samples they did provide have a **lot** of features. After visualizing the forest's decision boundary, you fear it might be overfit. There are many instances where long, erroneous looking spikes shoot out that only correctly classify a single sample!

Until now, you had thought the mere use of random forests was enough to inhibit overfitting; but it looks like that's not enough. Your cousins are coming over soon and can't wait to see the results of your modeling, and you don't want to let them down!

What parameter might you increase to stop the overfitting?

● min_samples_split ✓

○ criterion

○ max_depth

○ max_leaf_nodes

○ n_estimators

EXPLANATION

The best choice from the list above is to increase min_samples_split. By increasing that parameter, you effectively require more samples be contained in a node before your decision tree is allowed to split it.

In the question we mentioned your DTree algorithm was targeting single samples. By forcing internal nodes to have a higher sample count, you promote any non-split node that fails to meet the threshold from an internal node to a leaf. As such, it never has the chance to get split and chase after individual samples.

N_Estimators is also a good guess since increasing the number of trees typically offers better or more averaged results. But you don't have that many observations to begin with, so you'd only end up with more erroneous trees, each finely trained to correctly classify outliers!

You have used 1 of 2 submissions