Summer 2019 Project 4: Defeat Learners

From Quantitative Analysis Software Courses

Contents

Revisions

This assignment is subject to change up until 3 weeks prior to the due date. We do not anticipate changes; any changes will be logged in this section.

Overview

For this homework you will generate data that you believe will work better for one learner than another. This will test your understanding of the strengths and weaknesses of various learners. The two learners you should aim your datasets at are:

- A decision tree learner with leaf_size = 1 (DTLearner). Note that for testing purposes we will use our implementation of DTLearner
- The LinRegLearner provided as part of the repo.

Your data generation should use a random number generator as part of its data generation process. We will pass your generators a random number seed. Whenever the seed is the same you should return exactly the same data set. Different seeds should result in different data sets.

Template and Data

Instructions:

- Download the appropriate zip file File:19summer defeat learners.zip
- You should see the following files and directory
  - `defeat_learners/` the assignment directory
  - `defeat_learners/gen_data.py` An implementation of the code you are supposed to provide: It includes two functions that return a data set, and a third function that returns a user ID. Note that the data sets those functions return DO NOT satisfy the requirements for the homework. But they do

show you how you can generate a data set.

- `defeat_learners/LinRegLearner.py` Our friendly, working, correct, linear regression learner. It is used by the grading script. Do not rely on local changes you make to this file, as you may only submit `gen_data.py`.
- `defeat_learners/DTLearner.py` A working, but INCORRECT, Decision Tree learner. Replace it with your working, correct DTLearner.
- `defeat_learners/testbest4.py` Code that calls the two data set generating functions and tests them against the two learners. Useful for debugging.
- `defeat_learners/grade_best4.py` The grading script; for more details see here: ML4T_Software_Setup#Running_the_grading_scripts

Generate your own datasets

Create a Python program called gen_data.py that implements two functions. The two functions should be named as follows, and support the following API:

```
X1, Y1 = best4LinReg(seed = 5)
X2, Y2 = best4DT(seed = 5)
```

- **seed** Your data generation should use a random number generator as part of its data generation process. We will pass your generators a random number seed. Whenever the seed is the same you should return exactly the same data set. Different seeds should result in different data sets.

best4LinReg() should return data that performs significantly better (see rubric) with LinRegLearner than DTLearner. best4DT() should return data that performs significantly better with DTLearner than LinRegLearner.

Each data set should include from 2 to 10 columns in X, and one column in Y. The data should contain from 10 (minimum) to 1000 (maximum) rows.

Implement the author() function

Update the author() function to use your own user ID.

What to turn in

Be sure to follow these instructions diligently!

Via Canvas, submit as attachment (no zip files; refer to schedule for deadline):

- Your code as `gen_data.py`

We WILL NOT use your DTLearner, or LinRegLearner, so do not submit them.

Unlimited resubmissions are allowed up to the deadline for the project.

Rubric

Deductions:

- Does either dataset returned contain fewer or more than the allowed number of samples? -20 points each.
- Does either dataset returned contain fewer or more than the allowed number of dimensions in X? -20 points each.
- When the seed is the same does the best4LinReg dataset generator return the same data? -20 points otherwise.
- When the seed is the same does the best4DT dataset generator return the same data? -20 points otherwise.
- When the seed is different does the best4LinReg dataset generator return different data? -20 points otherwise.
- When the seed is different does the best4DT dataset generator return different data? -20 points otherwise.
- Is the author() method implemented? -10 points if not.
- Does the code attempt to import a learner? -10 points if so.

For best4LinReg (1 test case):

- We will call best4LinReg 15 times, and select the 10 best datasets. For each successful test +5 points (total of 50 points)
- For each test case we will randomly select 60% of the data for training and 40% for testing.
- Success for each case is defined as: RMSE LinReg < RMSE DT * 0.9

For best4DT (1 test case):

- We will call best4DT 15 times, and select the 10 best datasets. For each successful test +5 points (total of 50 points)
- For each test case we will randomly select 60% of the data for training and 40% for testing.
- Success for each case is defined as: RMSE DT < RMSE LinReg * 0.9

Required, Allowed & Prohibited

Required:

- No reading of data from files.
- Your project must be coded in Python 2.7.x.
- Your code must run on one of the university-provided computers (e.g. buffet01.cc.gatech.edu), or on one of the provided virtual images.
- Your code must run in less than 5 seconds on one of the university-provided computers.
- The code you submit should NOT include any data reading routines. You should generate all of your data within your functions.
- The code you submit should NOT generate any output: No prints, no charts, etc.

Allowed:

- You can develop your code on your personal machine, but it must also run successfully on one of the university provided machines or virtual images.
- Your code may use standard Python libraries.
- You may use the NumPy, SciPy, matplotlib and Pandas libraries. Be sure you are using the correct versions.
- Code provided by the instructor, or allowed by the instructor to be shared.
- Cheese.

Prohibited:

- Any reading of data files.

- Any libraries not listed in the "allowed" section above.
- Any code you did not write yourself.
- Any Classes (other than Random) that create their own instance variables for later use (e.g., learners like kdtree).
- Code that includes any data reading routines. The provided testlearner.py code reads data for you.
- Code that generates any output when verbose = False: No prints, no charts, etc.
- Ducks and wood.

Legacy

MC3-Homework-1-legacy

Retrieved from "http://quantsoftware.gatech.edu/index.php?title=Summer_2019_Project_4:_Defeat_Learners&oldid=3174"

- This page was last modified on 13 May 2019, at 23:55.
- This page has been accessed 3,292 times.