CS7646 ML4T
Machine Learning
for Trading

# PROJECT 4: DEFEAT LEARNERS

## 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

In this assignment, 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. You will submit the code for the project in Gradescope SUBMISSION. There is no report associated with this assignment.

## ABOUT THE PROJECT

In this project, you will evaluate the relative strengths of two types of supervised learners: a linear regression learner and a decision tree learner. Specifically, you will write a module that generate datasets that will be used by the learners. Your goal is to 1) produce datasets that enable a linear regression learner to outperform a decision tree learner and 2) produce datasets that enable a decision tree learner to outperform a linear regression learner.

Your data generation implementation 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. This project may require readings or additional research to ensure an understanding of the relative strengths and weaknesses of the different types of learners.

## YOUR IMPLEMENTATION

You will create a Python program called gen_data.py, which three functions according to this API specification. The DTLearner and LinRegLearner specification are also provided in the API specification as a reference. This gen_data.py file must implement the three functions given in the API specification:

- best_4_lin_reg
- best_4_dt
- author

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 (as shown below). Whenever the seed is the same you must return exactly the same data set. Different seeds must result in different data sets.

## TASK & REQUIREMENTS

**Construct the Best for Linear Regression Learner and Decision Tree Learner Datasets**

Implement the **best_4_lin_reg()** function such that when the generated dataset is passed to both learners, the LinRegLearner performs significantly better than the DTLearner (see rubric below for specific performance details).

Also implement the **best_4_dt()** function such that when the generated dataset is passed to both learners, the DTLearner performs significantly better than the LinRegLearner (see rubric below for specific performance details).

Each dataset must include no fewer than 2 and no more than 10 features (or "X") columns. The dataset must contain 1 target (or "Y") column. Each dataset must contain no fewer than 10 and no more than 1000 examples (I.e., rows). While you are free to determine these sizes, they may not vary between generated testsets.

Example

X1, Y1 = best_4_lin_reg( seed = 5 ) X1, Y1 = best_4_dt( seed = 5 )

### Implement the author() function (Up to 10 point penalty)

You must implement a function called author() that returns your Georgia Tech user ID as a string. This is the ID you use to log into Canvas. It is not your 9 digit student number. Here is an example of how you might implement author():

```
def author():
  return 'tb34' # replace tb34 with your Georgia Tech username.
```

Implementing this method correctly does not provide any points, but there will be a penalty of up to -10 points for not implementing the author function.

## TECHNICAL REQUIREMENTS

You will use your DTlearner from Project 3 and the provided LinRegLeaner during development, local testing, and any testing performed in the Gradescope TESTING environment.

- The decision tree learner (DTLearner) will be instantiated with leaf_size=1. (Note: We expect you to fix your DTLearner implementation from Project 3, if it was incorrect since this code is required to run/test on Gradescope TESTING.)
- You will use the LinRegLearner that is provided as part of the project setup .zip file.

You should set the seed every time to ensure dataset reproducibility. The grader will pass in a seed, and if your code does not produce the same result for the same seed, and different results for different seeds, you will receive a points deduction.

No part of this project implementation may read data from files. All data must be generated by the functions.

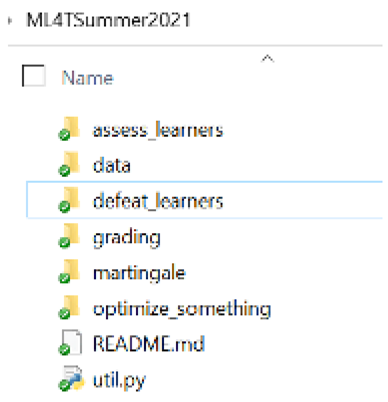
Your code must run in less than 5 seconds per test case.

The code you submit should NOT generate any output: No prints, no charts, etc.

## STARTER CODE

To make it easier to get started on the project and focus on the concepts involved, you will be given a starter framework. This framework assumes you have already set up the local environment and ML4T Software. The framework for Project 4 can be obtained from: defeat_learners2021Sum.zip.

Extract its contents into the base directory (e.g., ML4T_2021Summer). This will add a new folder called “defeat_learners” to the course directly structure.


ML4TSummer2021

Name

assess_learners
data
defeat_learners
grading
martingale
optimize_something
README.md
util.py

Within the defeat_learners folder are a folder and several files:

- defeat_learenrs/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 local grading / pre-validation script. This is the same script that will be executed in the Gradescope TESTING Environment

## CONTENTS OF REPORT

There is no report associated with this assignment.

## TESTING

To test your code, you can modify the provided testbest4.py file. You are encouraged to perform any unit tests necessary to instill confidence that the generated datasets will produce the desired learner results.

Additionally, we have provided the grade_best4.py file that can be used for your tests. This file is the same script that will be run when the code is submitted to Gradescope TESTING. This file is not a complete test suite and does not match the more stringent private grader that is used in Gradescope SUBMISSION. To run and test that the file will run from within the defeat_learners directory, use the command:

```
PYTHONPATH=../:. python grade_best4.py
```

You are encouraged to submit your files to Gradescope TESTING, where some basic pre-validation tests will be performed against the code. Gradescope TESTING does not grade your assignment. No credit will be given for coding assignment fails in Gradescope SUBMISSION that has also failed to pass this pre-validation in Gradescope TESTING.

NOTE: When submitting to Gradescope TESTING, you will also need to submit your DTLearner.py file in addition to the gen_data.py file.

You are allowed **unlimited** resubmissions to Gradescope **TESTING**. Please refer to the Gradescope Instructions for more information.

## SUBMISSION INSTRUCTIONS

**This is an individual assignment**. All work you submit should be your own. Make sure to cite any sources you reference and use quotes and in-line citations to mark any direct quotes.

Late work is not accepted without advanced agreement except in cases of medical or family emergencies. In the case of such an emergency, please contact the Dean of Students.

## Report Submission

There is no report submission associated with this assignment

## Code Submission

This class uses Gradescope, a server-side autograder, to evaluate your code submission. No credit will be given for code that does not run in this environment and students are encouraged to leverage Gradescope TESTING prior to submitting an assignment for grading.

Please submit the following file to Gradescope SUBMISSION:

- **gen_data.py**

Do not submit any other files.

You are allowed a MAXIMUM of three (3) code submissions to Gradescope SUBMISSION.

## GRADING INFORMATION

The submitted code is run as a batch job after the project deadline. The code represents 100% of the assignment grade will be graded using a rubric design to mirror the implantation details above. Deductions will be applied for unmet implementation requirements or code that fails to run.

We do not provide an explicit set timeline for returning grades, except that everything will be graded before the institute deadline (end of the term). As will be the case throughout the term, the grading team will work as quickly as possible to provide project feedback and grades. Once grades are released, any grade related matters must follow the Assignment Follow-Up guidelines and process.

## RUBRIC

### Auto-Grader [100 Points]

**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 best_4_lin_reg dataset generator return the same data? (-20 points otherwise)
- When the seed is the same does the best_4_dt dataset generator return the same data? (-20 points otherwise)
- When the seed is different does the best_4_lin_reg dataset generator return different data? (-20 points otherwise)
- When the seed is different does the best_4_dt 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 best_4_lin_reg (1 test case)**

- We will call best_4_lin_reg 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 best_4_dt (1 test case)**

- We will call best_4_dt 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

## ALLOWED

- You may use code written in other GT OMS courses if the following three conditions are met: 1) you are the 100% sole author of the code, 2) the code fully meets the requirements of this assignment, and 3) the code is properly cited and referenced.
- All libraries provided in the Local Environment setup page (be sure to use the correct package versions)
- All standard Python libraries (except the os library, which is prohibited)
- Code provided by the instructor or allowed by the instructor to be shared.

-

## PROHIBITED

- You may not read data files.
- You may not use any data reading routines or functions.
- You may not use any libraries not listed in the “allowed” section above.
- You may not use any code you did not write yourself.
- You may not use any classes (other than Random) that create their own instance variables for later use (e.g., learners like kdtree).
- Code may not send any output to the screen/console/terminal (other than run-time “warning” messages). Note: You may generate and save charts as .png files as indicated in the instructions.
- Code must not create extra directories.
- Code must not use of global variables.
- Code must not use absolute import statements, such as: *from folder_name import optimization*, where “folder_name” is the path/name of a folder or directory.
- Do not discuss solutions or approaches, since the assignment requires “[generating] data that you believe will work better for one learner than another”. However, it is okay to discuss the strengths/weaknesses and pros/cons of various learners.
- You must **not** share tables, charts, and statistics.

## RELEVANT RESOURCES

You may find the following resources useful in completing the project or in providing an in-depth discussion of the material.

- Mitchell, Machine Learning (Chapter 3)
- James, D. Witten, T. Hastie, R. Tibshirani (2017), An Introduction to Statistical Learning (Chapters 3 and 8)

Videos:

- Decision Tree Videos, Charles Isbell and Michael Littman, Georgia Tech ML 7641
- Decision Tree Video (Part 1, staring around 0:40 minutes), Tom Mitchell, CMU 601
- Decision Tree Video (Part 2), Tom Mitchell, CMU 601