



# Project 4: Defeat Learners

## DTLearner.py

**`class DTLearner.DTLearner(leaf_size=1, verbose=False)`**

This is a decision tree learner object that is implemented incorrectly. You should replace this DTLearner with your own correct DTLearner from Project 3.

### Parameters

- **leaf\_size** (*int*) – The maximum number of samples to be aggregated at a leaf, defaults to 1.
- **verbose** (*bool*) – If “verbose” is True, your code can print out information for debugging.

If verbose = False your code should not generate ANY output. When we test your code, verbose will be False.

**`add_evidence(data_x, data_y)`**

Add training data to learner

### Parameters

- **data\_x** (*numpy.ndarray*) – A set of feature values used to train the learner
- **data\_y** (*numpy.ndarray*) – The value we are attempting to predict given the X data

**`author()`**

### Returns

The GT username of the student

### Return type

str

**`query(points)`**

Estimate a set of test points given the model we built.

**Parameters**

**points** (*numpy.ndarray*) – A numpy array with each row corresponding to a specific query.

**Returns**

The predicted result of the input data according to the trained model

**Return type**

numpy.ndarray

## gen\_data.py

**author()****Returns**

The GT username of the student

**Return type**

str

**best\_4\_dt(*seed=1489683273*)**

Returns data that performs significantly better with DTLearner than LinRegLearner.

The 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.

**Parameters**

**seed** (*int*) – The random seed for your data generation.

**Returns**

Returns data that performs significantly better with DTLearner than LinRegLearner.

**Return type**

numpy.ndarray

**best\_4\_lin\_reg(*seed=1489683273*)**

Returns data that performs significantly better with LinRegLearner than DTLearner.

The 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.

**Parameters**

**seed** (*int*) – The random seed for your data generation.

**Returns**

Returns data that performs significantly better with LinRegLearner than DTLearner.

**Return type**

numpy.ndarray

## LinRegLearner.py

**`class LinRegLearner.LinRegLearner(verbose=False)`**

This is a Linear Regression Learner. It is implemented correctly.

### **Parameters**

**`verbose`** (*bool*) – If “verbose” is True, your code can print out information for debugging.

If `verbose = False` your code should not generate ANY output. When we test your code, `verbose` will be False.

**`add_evidence(data_x, data_y)`**

Add training data to learner

### **Parameters**

- **`data_x`** (*numpy.ndarray*) – A set of feature values used to train the learner
- **`data_y`** (*numpy.ndarray*) – The value we are attempting to predict given the X data

**`author()`**

### **Returns**

The GT username of the student

### **Return type**

str

**`query(points)`**

Estimate a set of test points given the model we built.

### **Parameters**

**`points`** (*numpy.ndarray*) – A numpy array with each row corresponding to a specific query.

### **Returns**

The predicted result of the input data according to the trained model

### **Return type**

numpy.ndarray

## testbest4.py

**`compare_os_rmse(learner1, learner2, x, y)`**

Compares the out-of-sample root mean squared error of your LinRegLearner and DTLearner.

### **Parameters**

- **`learner1`** (*class:'LinRegLearner.LinRegLearner'*) – An instance of LinRegLearner
- **`learner2`** (*class:'DTLearner.DTLearner'*) – An instance of DTLearner

- **x** (*numpy.ndarray*) – X data generated from either `gen_data.best_4_dt` or `gen_data.best_4_lin_reg`
- **y** (*numpy.ndarray*) – Y data generated from either `gen_data.best_4_dt` or `gen_data.best_4_lin_reg`

**Returns**

The root mean squared error of each learner

**Return type**

tuple

**test\_code()**

Performs a test of your code and prints the results

©2020, ML4T Staff | Powered by [Sphinx 2.2.0](#) & [Alabaster 0.7.12](#)