



CLASSIFICATION TRADER HINTS

OVERVIEW

You will utilize your Random Tree learner to train and test a learning trading algorithm. Here are some ideas (gathered from a previous project) that you might find helpful if you are going to use a classification or regression learner for your trader.

ML TRADER

Convert your decision tree **regression** learner into a **classification** learner. The classifications should be:

- +1: LONG
- 0: CASH
- -1: SHORT

The X data for each sample (day) are simply the values of your indicators for the stock — you should have 3 to 5 of them. The Y data (or classifications) will be based on N day return (your choice for N). You should classify the example as a +1 or “LONG” if the N day return exceeds a certain value, let’s call it YBUY for the moment. You should classify the example as a -1 or “SHORT” if the N day return is below a certain value we’ll call YSELL. In all other cases the sample should be classified as a 0 or “CASH.” Note that it is very important that you train your learner with these classification values (not the N day returns).

Note that your X values are calculated each day from the current day’s (and earlier) data, but the Y value (classification) is calculated using data from the future. You may tweak various parameters of your learner to maximize return (more on that below). Train and test your learning strategy over the in sample period.

Important note: You must set the leaf_size parameter of your decision tree learner to 5 or larger. This requirement is intended to avoid a degenerate overfit solution to this problem.

You should tweak the parameters of your learner to maximize performance during the in sample period. Here is a partial list of things you can tweak:

- Adjust YSELL and YBUY.
- Adjust leaf_size.
- Utilize bagging and adjust the number of bags.

OVERVIEW

- Indicator design hints:
 - For your X values: Identify and implement at least 3 technical features that you believe may be predictive of future return.
- Train a classification learner on in sample training data:
 - For your Y values: Use future N day return (not future price). Then classify that return as LONG, SHORT or CASH. You're trying to predict a relative change that you can use to invest with.
 - For debugging purposes, you may find it helpful to plot the value of the training classification data (-1, 0, 1) versus the stock price in one color.
 - For debugging purposes, you may find it helpful to plot the value of the training classification output (-1, 0, 1) versus the stock price in another color. Ideally, these two lines should be very similar.

Your code should classify based on N day change in price. You need to build a new Y that reflects the N day change and aligns with the current date. Here's pseudo code for the calculation of Y

Code

```
ret = (price[t+N]/price[t]) - 1.0
if ret > YBUY:
    Y[t] = +1 # LONG
else if ret < YSELL:
    Y[t] = -1 # SHORT
else:
    Y[t] = 0 # CASH
```

If you select Y in this manner and use it for training, your learner will classify N day returns.

