

# MC3-Project-2

From Quantitative Analysis Software Courses

## Contents

- 1 Updates / FAQs
- 2 Overview
- 3 Summary of Plots To Create
- 4 Template and Data
- 5 Choosing Technical Features -- Your X Values
- 6 Choosing Y
- 7 Contents of Report
- 8 Hints & resources
- 9 What to turn in
- 10 Extra credit up to 3%
- 11 Rubric
- 12 Required, Allowed & Prohibited

## Updates / FAQs

2015-11-21

- Updated list of charts to generate.
- Q: In a previous project there was a constraint of holding a single position until exit. Does that apply to this project? Yes, hold one position til exit.

2015-11-22

- Q: Is that 5 calendar days, or 5 trading days (i.e., days when SPY was traded)? A: Always use trading days.

2015-11-23

- Q: Are there constraints for Python modules allowed for this project? Can we experiment with modules for optimization or technical analysis and cite or are we expected to write everything from scratch for this project as well? A: You can use whatever modules you like as long as you cite them. You've already written most of what you need though.
- Q: Can we change our policy to work better for IBM vs ML4T-399? A: No, you must use the same indicators, policy, etc. for both. I suggest you optimize first for IBM, then go back to 399 because almost anything should work with 399.

# Overview

In this project you will transform your regression learner into a stock trading strategy. Overall, you should follow these steps:

- Train a regression learner (KNN or LinReg, or other of your choice with or without bagging) on data from 2008 to 2009. This is your in sample training data.
  - For your X values: Identify and implement at least 3 technical features that you believe may be predictive of future return. You should implement them so they output values typically ranging from -1.0 to 1.0. This will help avoid the situation where one feature overwhelms the results. See a few formulae below.
  - For your Y values: Don't use price, use future 5 day return. Remember you're trying to PREDICT the future.
- Create a plot that illustrates your training Y values in one color, price in another color and your model's PREDICTED Y in a third color. With this chart we should be able to see how well your learner performs and that your Y values are shifted back 5 days. You may find it convenient to zoom in on a particular time period so this is evident.
- Create a trading policy based on what your learner predicts for future return. As an example you might choose to buy when the forecaster predicts the price will go up more than 1%, then hold for 5 days.
- Create a plot that illustrates entry and exits as vertical lines on a price chart for the in sample period 2008-2009. Show long entries as green lines, short entries as red lines and exits as black lines. You may find it convenient to zoom in on a particular time period so this is evident.
- Now use your code to generate orders and run those orders through your market simulator. Create a chart of this backtest. It should do VERY well for the in sample period 2008-2009.
- Freeze your model based on the 2008-2009 data. Now test it for the year 2010 -- Plot that illustrates entry & exits, generate trades, run through simulator, chart the backtest.

Perform the above steps first using the data ML4T-399.csv, once you've validated success (it should work well), repeat using IBM data over the same dates. Remember 2008-2009 is training, 2010 is testing. You should have one set of charts for each symbol.

## Summary of Plots To Create

1. Training Y/Price/Predicted Y: Create a plot that illustrates your training Y values in one color, price in another color and PREDICTED Y in a third color. With this chart we should be able to see how well your learner performs and that your Y values are shifted back 5 days. You may find it convenient to zoom in on a particular time period so this is evident.
2. In Sample Entries/Exits: Create a plot that illustrates entry and exits as vertical lines on a price chart for the in sample period 2008-2009. Show long entries as green lines, short entries as red lines and exits as black lines. You may find it convenient to zoom in on a particular time period so this is evident.
3. In Sample Backtest
4. Out of Sample Entries/Exits: Freeze your model based on the 2008-2009 data. Now test it for the year 2010 -- Plot that illustrates entry & exits, generate trades,
5. Out of Sample Backtest

6. IBM Data In Sample Entries/Exits: Create a plot that illustrates entry and exits as vertical lines on a price chart for the in sample period 2008-2009. Show long entries as green lines, short entries as red lines and exits as black lines. You may find it convenient to zoom in on a particular time period so this is evident.
7. IBM Data In Sample Backtest
8. IBM Data Out of Sample Entries/Exits: Freeze your model based on the 2008-2009 data. Now test it for the year 2010 -- Plot that illustrates entry & exits, generate trades,
9. IBM Data Out of Sample Backtest

## Template and Data

You will use data in the ML4T/Data directory. In particular files named ML4T-399.csv, and IBM.csv.

## Choosing Technical Features -- Your X Values

You should have already successfully coded the Bollinger Band feature. Here's a suggestion of how to normalize that feature so that it will typically provide values between -1.0 and 1.0:

```
bb_value[t] = (price[t] - SMA[t]) / (2 * stdev[t])
```

Two other good features worth considering are momentum and volatility.

```
momentum[t] = (price[t] / price[t-N]) - 1
```

Volatility is just the stdev of daily returns.

## Choosing Y

Your code should predict 5 day change in price. You need to build a new Y that reflects the 5 day change and aligns with the current date. Here's pseudo code for the calculation of Y

```
Y[t] = (price[t+5] / price[t]) - 1.0
```

If you select Y in this manner and use it for training, your learner will predict 5 day returns.

## Contents of Report

- Your report should be no more than 8 pages long. Use 1" margins and no smaller than 10 point font. Your report should contain no more than 12 charts. Penalties will apply if you violate these constraints.
- Include the charts listed in the overview section above.
- Describe each of the indicators you have selected in enough detail that someone else could reproduce them in code.

- Describe your trading policy clearly.
- Discussion of results. Did it work well? Why? What would you do differently?

## Hints & resources

### What to turn in

Turn your project in via t-square.

- Your report as `report.pdf`
- Your code as `code.py`

### Extra credit up to 3%

Choose one or more of the following:

- Compare the performance of KNN and LinReg in this task. The instructor anticipates that LinReg might work well. If that turns out to be the case, how can that be? This is a non-linear task.
- Extend your code to create a "rolling" model that updates each day rolling forward.
- Extend your code to simultaneously forecast all the members of the S&P 500. Generate trades accordingly, and backtest the result.

Submit to the extra credit assignment on t-square. One single PDF file only, max 5 pages.

### Rubric

- Are all 9 plots present and correct? -5 points for each missing plot.
  - Note: Correct in the sense that they properly display the information requested. The result may not be the desired one.
- Are numerical results correct? (ML4T-399 in sample & out of sample, IBM in sample & out of sample) -10 points for each incorrect result.
- Indicators used: Are descriptions of factors used sufficiently clear that others could reproduce them? Up to -10 points for lack of clarity.
- Trading strategy: Is description sufficiently clear that others could reproduce it? Up to -10 points for lack of clarity.
- Is discussion of results concise, complete, correct? Up to -5 points for each of concise, complete, correct.

### Required, Allowed & Prohibited

[for 2016]

Required:

- Your project must be coded in Python 2.7.x.

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.
- You may reuse sections of code (up to 5 lines) that you collected from other students or the internet.
- Code provided by the instructor, or allowed by the instructor to be shared.

Prohibited:

- Any libraries not listed in the "allowed" section above.
- Any code you did not write yourself (except for the 5 line rule in the "allowed" section).
- Any Classes (other than Random) that create their own instance variables for later use (e.g., learners like kdtree).
- Holy hand grenades.

Retrieved from "<http://quantsoftware.gatech.edu/index.php?title=MC3-Project-2&oldid=744>"

- 
- This page was last modified on 9 December 2015, at 14:35.
  - This page has been accessed 5,189 times.