CS7646 ML4T
Machine Learning
for Trading

PROJECT 6: INDICATOR EVALUATION

DUE DATE

10/18/2020 11:59PM Anywhere on Earth time

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

This assigment counts towards 7% of your overall grade.

In this project you will develop technical indicators and a Theoretically Optimal Strategy that will be the ground layer of a later project. The technical indicators you develop will be utilized in your later project to devise an intuition-based trading strategy and a Machine Learning based trading strategy. Theoretically Optimal Strategy will give a baseline to gauge your later project's performance against. We hope Machine Learning will do better than your intuition, but who knows?

Please keep in mind that completion of this project is pivotal to Project 8 completion. Spending time to find and research indicators will help you complete the later project.

TEMPLATE

There is no distributed template for this project. You should create a directory for your code in ml4t/indicator_evaluation. You will have access to the data in the ML4T/Data directory but you should use ONLY the API functions in util.py to read it.

You should create the following code files for submission. They should comprise ALL code from you that is necessary to run your evaluations.

- `indicators.py` Your code that implements your indicators as functions that operate on DataFrames. The "main" code in indicators.py should generate the charts that illustrate your indicators in the report.
- `marketsimcode.py` An improved version of your marketsim code that accepts a "trades" DataFrame (instead of a file). More info on the trades data frame below. It is OK not to submit this file **if** you have subsumed its functionality into one of your other required code files.
- `TheoreticallyOptimalStrategy.py` Code implementing a TheoreticallyOptimalStrategy object (details below). It should implement testPolicy() which returns a trades data frame (see below). The "main" code in TheoreticallyOptimalStrategy.py should call marketsimcode as necessary to generate the plot and statistics for your Theoretically Optimal Strategy in the report.

DATA DETAILS, DATES & RULES

- Use only the data provided for this course. You are not allowed to import external data.
- Please add in an author function to each file.
- For your report, use only the symbol JPM. This will enable us to more easily compare results.
- Use the time period January 1, 2008 to December 31 2009.
- Starting cash is $100,000.
- Allowable positions are: 1000 shares long, 1000 shares short, 0 shares.
- Benchmark: The performance of a portfolio starting with $100,000 cash, investing in 1000 shares of JPM and holding that position.
- There is no limit on leverage.
- Transaction costs for TheoreticallyOptimalStrategy: Commission: $0.00, Impact: 0.00.
- Correct trades df format used.

TASKS

Part 1: Technical Indicators

Develop and describe 5 technical indicators. You may find our lecture on time series processing,the Technical Analysis video and vectorize_me PowerPoint to be helpful. For

each indicator you should create a single, compelling chart that illustrates the indicator (you can use sub-plots to showcase different aspects of the indicator).

As an example, you might create a chart that shows the price history of the stock, along with "helper data" (such as upper and lower Bollinger Bands) and the value of the indicator itself. Another example: If you were using price/SMA as an indicator you would want to create a chart with 3 lines: Price, SMA, Price/SMA. In order to facilitate visualization of the indicator you might normalize the data to 1.0 at the start of the date range (i.e. divide price[t] by price[0]).

Your report description of each indicator should enable someone to reproduce it just by reading the description. We want a written detailed description here, not code, however, it is OK to augment your written description with a **pseudocode** figure. Do NOT copy/paste code parts here as a description.

You should have already successfully coded the Bollinger Band feature:

```
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 standard Deviation of daily returns.

It is usually worthwhile to standardize the resulting values (see https://en.wikipedia.org/wiki/Standard_score).

You are allowed to use up to two indicators presented and coded in the lectures (SMA, Bollinger Bands, RSI) but the other three will need to come from outside the class material (momentum and volatility are allowed to be used).

NOTE: In Project-8, you will need to use the same indicators you will choose in in this project, we recom-mend you check Project-8 before choosing your indicators.When you decide on your indicators, be sure to describe how they create buy and sell signals. While it's not necessary in Project 6 to code the indicators this way, it is required for Project 8. For example, Bollinger Bands alone does not give an actionable signal to buy/sell that is easily framed for a learner but BBP/%B does.

Part 2: Theoretically Optimal Strategy

Assume that you can see the future, but that you are constrained by the portfolio size and order limits as specified above. Create a set of trades that represents the best a strategy could possibly do during the in-sample period.The reason we're having you do this is so that you will have an idea of an upper bound on performance, which can be referenced in Project 8. Note: Theoretically Optimal Strategy does not use the indicators developed in the previous section.

Use adjusted close prices with the market simulator that you wrote earlier in the course. For this activity, use $0.00 and 0.0 for commissions and impact respectively.

Provide a chart that reports:

- Benchmark (see definition above) normalized to 1.0 at the start: Green line
- Value of the theoretically optimal portfolio (normalized to 1.0 at the start): Red line

You should also report in your report:

- Cumulative return of the benchmark and portfolio
- Stdev of daily returns of benchmark and portfolio
- Mean of daily returns of benchmark and portfolio

Your code should implement testPolicy() as follows:

```
import TheoreticallyOptimalStrategy as tos
df_trades = tos.testPolicy(symbol = "AAPL", sd=dt.datetime(2010, 1, 1), ed=
```

The input parameters are:

- symbol: the stock symbol to act on
- sd: A datetime object that represents the start date
- ed: A datetime object that represents the end date
- sv: Start value of the portfolio

The output result is:

- df_trades: A data frame whose values represent trades for each day. Legal values are +1000.0 indicating a BUY of 1000 shares, -1000.0 indicating a SELL of 1000 shares, and 0.0 indicating NOTHING. Values of +2000 and -2000 for trades are also legal so long as net holdings are constrained to -1000, 0, and 1000.

Implement author() function (deduction if not implemented)

You should implement a function called `author()` that returns your Georgia Tech user ID as a stringin all python classes created. 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 for not implementing it.

Implement Test Project

Not included in template. You will have to create this code file.

Create testproject.py and implement the necessary calls (following each respective API) to **indicators.py**,**TheoreticallyOptimalStrategy.py**, with the appropriate parameters to run everything needed for the report in a single Python call:

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

This is to have a single entry point to test your code against the report. Calling testproject.py should run all assigned tasks and output all necessary charts and statistics for your report.

README.txt

Create a README.txt file that has:

- Description of what each file is for/does
- explicit instructions on how to properly run your code.

Report

Answer the following prompt in a maximum of 10 pages (excluding references) in JDF format. Any content beyond 10 pages will not be considered for a grade. Ten pages is a maximum, not a target; our recommended per-section lengths intentionally add to less than 10 pages to leave you room to decide where to delve into more detail. This length is intentionally set expecting that your submission will include diagrams, drawings, pictures,

etc. These should be incorporated into the body of the paper unless specifically required to be included in an appendix.

The JDF format specifies font sizes and margins, which should not be altered. Include charts to support each of your answers. Charts should be generated by the code and saved to files. Charts should be properly annotated with legible and appropriately named labels, titles, and legends.

Please address each of these points / questions, the questions asked in the Project 6 wiki, and the items stated in the Project 6 rubric in your report. The report is to be submitted as **report.pdf**.

Part 1: Indicators: ~ 5 pages

At minimum, address each of the following for each indicator:

- Introduce and describe each indicator you use in sufficient detail that someone else could reproduce it. (The indicator can be described as a mathematical equation or as pseudo code).
- Provide a compelling description regarding why that indicator might work and how it could be used. (Explain how the indicator could be used alone and/or in conjunction with other indicators go generate buy/sell signals).
- Provide one or more charts that convey how each indicator works in a compelling way. (up to 3 charts per indicator).

The total number of charts for Part 1 must not exceed 10 charts.

Part 2: Theoretically Optimal Strategy (TOS) ~ 1.5 pages

For the Theoretically Optimal Strategy, at a minimum, address each of the following:

- Describe how you created the strategy and any assumptions you had to make to make it work.
- Describe the strategy in a way that someone else could evaluate and/or implement it.
- Provide a chart that illustrates the TOS performance versus the benchmark.

Note: An abstract (as specified in the JDF format) is not required for this report.

References: ~0.25 pages

References should be placed at the end of the paper in a dedicated section. Reference lists should be numbered and organized alphabetically by first author's last name. If multiple papers have the same author(s) and year, you may append a letter to the end of the year to allow differentiated in-line text (e.g. Joyner, 2018a and Joyner, 2018b in the section above). If multiple papers have the same author(s), list them in chronological order starting

with the older paper. Only works that are cited in-line should be included in the reference list. The reference list does not count against the length requirements.

Submission Instructions

Complete your assignment using JDF, then save your submission as a PDF. Assignments should be submitted to the corresponding assignment submission page in Canvas. You should submit a single PDF for this assignment. Be sure to following the instructions in Canvas for the report and code submissions.

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 immediately contact the Dean of Students followed by contacting the "Instructors" through a Piazza private post.

WHAT TO TURN IN

Be sure to follow these instructions diligently! No zip files.

Canvas:

Submit the following files (only) via Canvas before the deadline:

- Project 6: Indicator Evaluation (Report)
  - Your report as report.pdf.

Gradescope:

- (SUBMISSION) Project 6: Indicator Evaluation
  - Your code as testproject.py, indicators.py, TheoreticallyOptimalStrategy.py and marketsimcode.py (optional if needed)
  - README.txt document

Do not submit any other files.

You are only allowed 3 submissions to **(SUBMISSION) Project 6: Indicator Evaluation** but unlimited resubmissions are allowed on **(TESTING) Project 6: Indicator Evaluation.**

Note that Gradescope does **not** grade your assignment live; instead, it pre-validates that it will run against our batch autograder that we will run after the deadline. There will be **no** credit given for coding assignments that do not pass this pre-validation.

Refer to the Gradescope Instructions for more information.

RUBRIC

Report

General

- Neatness (up to 5 points deduction if not).
- Bonus for exceptionally well written reports (up to 2 points)
- Is the required report provided (-100 if not)

Indicators

- Are at least three indicators different from those provided/coded by the instructor's code or lectures (i.e., another indicator that is not SMA, Bollinger Bands or RSI) (-15 points each if not)
- Does the submitted code indicators.py properly reflect the indicators provided in the report (up to -75 points if not)

Individual Indicators (up to 15 points potential deductions per indicator):

- Is there a compelling description why the indicator might work (-5 if not)
- Is the indicator described in sufficient detail that someone else could reproduce it? (-5 points if not)
- Is there a chart for the indicator that properly illustrates its operation including properly labeled axis and legend? (up to -5 points if not)

Theoretically optimal (up to 20 points potential deductions):

- Is the methodology described correct and convincing? (-10 points if not)
- Is the chart correct (dates and equity curve) including properly labeled axis and legend (up to -10 points if not)
- Historic value of benchmark normalized to 1.0 with green line (-5 if not)
- Historic value of portfolio normalized to 1.0 with red line (-5 if not)
- Are the reported performance criteria correct ? See appropriate section for required stats. (-2 points for each item if not)

Code

Is the required code provided, including code to recreate the charts and usage of correct trades DataFrame. **DO NOT use plt.show() and/or manually save your charts. The**

charts should be created and saved using Python code (up to -100 if not)

Auto-Grader

No auto-grader

REQUIRED, ALLOWED & PROHIBITED

Required:

- Your project must be coded in Python 3.6.x.
- Your code must be submitted to Gradescope in the appropriate Gradescope assignment.
- Use only the API functions in util.py to read data.
- All charts must be generated in Python, and you must provide the code you used.

Allowed:

- You can develop your code on your personal machine, but it must also run successfully on Gradescope.
- 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.
- A herring.

Prohibited:

- Generating charts using a method other than Python.
- Any other method of reading data besides util.py
- Modifying (or depending on modifications to) util.py.
- Any use of plot.show()
- Absolute import statements of the **current** project folder such as `from indicator_evaluation import XXXX or import indicator_evaluation.XXXX`
- Extra directories (manually or code created)
- Extra files not listed in "WHAT TO TURN IN"
- Any libraries not listed in the "allowed" section above.
- Any code you did not write yourself.

FAQ

- **Q:** I want to read some other values from the data besides just adjusted close, how can I do that?
  A: Look carefully at util.py and you will see that you can query for other values.

- **Q:** Are we only allowed one position at a time?
  A: You can be in one of three states: -1000 shares, +1000 shares, 0 shares.

- **Q:** Are we required to trade in only 1000 share blocks? (and have no more than 1000 shares long or short at a time?
  A: You can trade up to 2000 shares at a time as long as you maintain the requirement of holding 1000, 0 or -1000 shares.

- **Q:** Are we limited to leverage of 2.0 on the portfolio?
  A: There is no limit on leverage.