Financial Data Analytics
Professor Huang

Financial Data Analytics Project

Due Date: Friday, April 28 at 11:59 pm ET

Deliverable: A Project Report

A. Objective: We will backtest a portfolio strategy of the students' choice over a 30-40 year period using the Python language and a variety of data sources.[1]

B: Deliverable: A project report. The report should not exceed 10 pages, including summary and conclusion, but excluding all tables and appendices, and should be typed double-spaced with a font size of 12 points. Technical details should be delegated to tables and appendices. It is important that the report be easy to follow and understand. The grade of the report will be based on:

- Qualitative and quantitative analysis
- Quality of write-up
- Overall impression of the commitment to the project

Please deposit your report, and the associated computer codes, to dropbox submission folder "Final Project" at the course Learn website. Each group in the class will be named with group combination, for example, "G1". Accordingly, please name your report file as Group Name + Report1, for example, "G1 Project Report". Pdf file is preferred; however, word is acceptable.

The project accounts for 30% of the course mark. No late report will be accepted (as I have a definite University deadline to report your course mark).

C. Guidelines on Quantitative Trading Strategy:

In-Sample "model fitting period" or "model training period" (I suggest using first 2/3 or 3/4 of data time period for return predictions and model validation—for example, use the first 30 years of sample as the model training period for a 40-year sample period). Depending on your algorithm, you may also need to split your sample into training, validation, and testing subsamples.

Estimation Rules:

1. Universe of test assets. You have two choices of the universe of stocks. First choice: NYSE/Nasdaq stocks, only common stocks (not preferred, ETFs, etc.). Second choice: S&P 1500 stocks. For the second choice, you can use the data posted for the machine learning slide sets. Aside from these two, you may have your own universe of securities, for example, crypto currencies; in this case, please articulate clearly where your data comes

[1] If you prefer, you can also use Matlab, R, or SAS to code your project. If you wish to use another language, e.g., Eviews, C++, or Stata, please check with the instructor to approve this prior to working on your code.

from and your data filtering process. In the last case, you also need to be able to mimic similarly the guideline herein (for example, use a reasonably long period of data, construct your own factors and cite literature supporting your choice of factors).

2. Each stock must have a price that exceeds $5 per share and a market capitalization of equity of at least $100 million at the beginning of a given forecast year (January of each year), to avoid trading illiquid stocks. Similar data screening should be carried out if you use alternative data.

3. You should start with a factor model having 10 to 20 chosen factors, as some factors may not work out for you during this backtest step. It is normal to have to drop several factors because they don't work. The factors can be chosen from the textbook, research papers, or outside sources, but you should have some source that identifies each of these as "likely candidates" for picking stocks, and what to expect in terms of returns when you implement them as well as the proper way to construct the factors. One important consideration is to think about "categories" of factors when you decide on the set of factors. For one review of "categories," see Hou, Xue and Zhang (2018). You can focus on, say, just one category of factors.

4. Keep in mind that you will need to collect historical data for each factor from WRDS (or elsewhere) back to (ideally) year 1985, or even earlier, which will limit the types of factors you can use. Feel free to use Bloomberg or CapitalIQ. **As a starting point, we have created a clean dataset of 50 monthly factors for S&P 1500 stocks from 1980 to 2019, alongside their monthly one-month- to 12-month-ahead returns, and they are posted on Learn as a part of the machine learning exercises. As a general rule, I recommend that you start your backtest no earlier than 1974 (depending on the data availability for your factors)—due to the lack of data on Nasdaq stocks prior to 1974—and stop with data in 2020.** You do not have to use any of these factors, but they are available for you. **Also, you can create some other factors, beyond these 50. Alternatively, you can extend your chosen factors to the universe of all NYSE/Nasdaq stocks. The more original factors you create (outside of the 50 provided) or the larger sample that you use the better chance you have a higher grade. Of course, the quality and difficulty of forming each factor also weighs heavily into the grade for your project, so just a few very well-chosen and difficult to obtain factors can be a good approach.**

5. **Uniformization and data filtering.** As you know, you should work with standardized values of each of your factors. For example, z-scores of each of your factors, relative to the industry average and standard deviation of each month. You should **not** standardize the predictive variable, which is usually the following-month return. Engage in other data filtering procedures as you deem appropriate.

6. Estimate your training set trading model with at least two methods: (1) Fama-MacBeth cross-sectional regressions (as you did in WRDS data assignment 2); (2) one of the machine learning methods covered in class. In your training set, the left-hand-side (or the output variable) is future returns, and the right-hand-side (or input variables) are the set of factors chosen.

7. In Fama-MacBeth regressions, your objective is to estimate the factor premia on each factor during each estimation window, then compute the average factor premia across all

windows and the *t*-statistic of this average (i.e., the Fama-MacBeth average and the Fama-MacBeth *t*-statistic). These represent the average profitability and the consistency of profitability of each factor, respectively. In machine learning, however, depending on the model used, the line is not entirely clear on factor selection in the training step. To make things simple, I advise to use the results from Fama-MacBeth regressions as your selection tool.

8. If you're working in a group, you should then meet with your group to have a serious discussion on which factors to drop from your final model, which would be appropriate if they end up with a backtest *t*-statistic that is low (e.g., below 1.5) or the wrong sign (e.g., negative for the E/P ratio, which should positively predict returns). **However, you may choose to keep a signal with a low *t*-statistic or even a wrong sign if there seems to be a strong economic story or a compelling research article that indicates it should work well when tried over many years. This is where your judgment comes in. (This is somewhat like using a Bayesian approach, where your strong priors outweigh the data result.)** You should avoid dropping too many factors, or you will have a very weak model in the out-of-sample tests. **A rough "rule-of-thumb" is to keep at least 5-10 factors.**

9. Repeat Steps 6-7 and finish training your model. By the end of this step, you have chosen the factors as the input for your model, and have trained your model for testing.

Out-of-sample test/testing sample forecasting of the model [your testing sample]:

10. Now it's time to use the model to score stocks during the out-of-sample period. In linear models such as Fama-MacBeth, the task is relatively straightforward. Predicted returns are:

$$\tilde{r}_{i,t} = \hat{a}_0 + \hat{b}_1 F1_{i,t} + \hat{b}_2 F2_{i,t} + \ldots + \hat{b}_K Fk_{i,t} \quad \text{for stocks } i \text{ at } t$$

where the RHS is the predicted return, *F1* to *Fk* are the factor values at time *t*, and the coefficients (including the intercept—which is alpha) are estimated factor coefficients (factor premia) from Step 9. This equation will provide you a mechanism to rank the stocks in each out-of-sample period.

Alternatively, you can use a weighted exposure of each stock to factors to sort stocks, for example, the score of each stock at each time period for each stock is given by:

Score = z-score(Factor 1) × t-stat(Factor 1) + z-score(2) × t-stat(2) + …

where *t*-stat is again from Step 9's Fama-MacBeth. (The t-stat should already have the proper sign if the backtest worked in the right direction for that factor.)

It's less straightforward in machine learning how prediction is determined; depending on the method employed, you might not even have a clue how certain prediction is made. In your ranking of stocks in the testing sample, I expect that you should be able to articulate the intuition behind the stock scoring for your chosen method of machine learning.

11. Portfolio weights going forward: It is recommended that you simply go long an equal-weighted portfolio of the top 10% of ranked stocks, and equal-weight short the bottom 10%. If you want to be fancier, you can try to optimize weights with a mean-variance optimization program, but this is not required. If you choose to do this, you can use any software or data source for choosing portfolios (positions in stocks each year). You should rebalance your portfolio at the end of each time period (in the next step below).

12. Repeat 10 & 11 above to compute scores for stocks for every month, until you end with the second to the last month the final sample period which is used to pick stocks for the last month.

13. Next, compute the return difference for the hedge portfolio between the top score equal-weighted 10% portfolio and the bottom score equal-weighted 10% portfolio **for each month** during the testing sample. (Or, if you used another weighting scheme, such as portfolio optimized, compute the difference between the long and short portfolios for that weighting).

14. Finally, you will conduct performance analytics on your resulting out-of-sample returns over all months in the out-of-sample period. You should compute (1) Raw return, (2) Sharpe ratio, (3) CAPM alpha, (4) 4-Factor alpha, and (5) Information Ratio using 4-factor model. Code for doing this is largely included in Python-lab #5.

15. To increase the power of your trading strategy, you may use a rolling window approach for the above steps. For example, building your trading strategy in any given month (i.e., the given month is used as the testing sample) by relying on the previous certain amounts of months as your training sample. This will increase the work load exponentially; in particular, in each month you need to update your choice of factors. This step is not required but offered as a mere suggestion as an option.

16. Your report should cover not only which factors you chose, but also mention the research articles that support them, and the predictions of the research (in terms of exact return expected and risk). You should state clearly in your report covering the results of your in-sample estimation, your selection of your final model (after backtesting), and the results of your out-of-sample test. You might end with a conclusion about why your model did/did not work very well during the out-of-sample period. And, what you might try differently in the future.

As a last caveat, if you wish to attempt some non-linear machine learning methods such as random forest or neural networks, you should keep running time in mind. Should you choose to go this route, I strongly recommend you to go with a portfolio approach as in Gu, Kelly, and Xiu (2020). For example, you can form 10 decile size portfolios at the beginning of your sample period, and try to predict the returns of the 10 decile size portfolios for your testing period. In other words, you can try to predict 10 asset returns instead of those of your universe of individual stocks—which apparently improves the running time dramatically.