Udacity MLND Capstone Project Proposal

Ruoyu Lin

Jan. 10

1 Domain Background

Quantitative finance/financial engineering is the practice of approaching financial analytics through extensively technical methods, such as dynamical stochastic models and machine learning approaches. In this project, we attempt to use financial data and machine learning techniques to conduct technical analysis on large market-cap, tech-sector stocks.

As machine learning is a fast growing field, the method utilized in this study is relatively new. The relevant timeline is as follows:

- In 1995, Tin Kam Ho first introduced the idea of random decision forest as an ensemble model of decision trees to solve stochastic classification problems[3].
- In 2001, Leo Breiman formally formulated the random forest classifier as we know it, by introducing the layer that bags a random selection of features in each estimator.
- In 2016, Tianqi Chen and Carlos Guestrin published the gradient boosting system called XGBoost based on the research project of Chen. XGBoost was built upon the traditional decision tree ensemble, but instead of treating each decision tree estimator equally like the random forest does, it instead compute gradient and hessian in each learning epoch, and update the weight put on each estimator. The XGBoost model generally vastly better than the traditional random forest appraoch.
- The earliest utilization of XGBoost in financial forecasting can be traced to 2016, the same year the XGBoost library was formally published, and was implemented by the same group of researchers that created the paper that inspired my previous study. Their study utilized gradient boosted tree ensembles to predict the movement direction of the stock market.

To sum up, as it can be observed, both the theoretic principles and the industrial utilization of this method are relatively novel, and more studies should be invested into the field.

2 Problem Statement

A Random Walk Down Wall Street[1] by former Vanguard director Burton Malkiel introduces the idea that although hedge funds and investment companies hire experienced portfolio managers to design dynamic strategies to maximize portfolio value, the vast majority of portfolio managers on Wall Street **do not outperform the stock market**. In other words, if one chooses to invest in high market-cap value stocks, designating the entire portfolio into market index funds such as SPY and holding for a long time on average yields a higher return than dynamically managing the portfolio.

Hence, following Malkiel's thesis without in-depth examination, it's somewhat meaningless to predict **exactly change of the stock price**; however, it might be conducive to identify **if/when would the stock price fall**. Should a model be able to somewhat accurately predict a fall of price, one can maintain a market-correlated portfolio with downside risks controlled, yielding higher returns. Trusting in Malkiel's thesis, I aim to train a ML classifier on stock data (volume, high, low, adjusted close, other engineered features) in technology sector (due to their high correlation with the market index), attempting to predict trading days with significant loss.

3 Data

The raw data will be obtained from yahoo finance, which is a trusted source for stock market data, including daily open price, close price, adjusted close price, high price, low price for stocks with ticker GOOG, FB, AAPL, NVDA, SPY, MSFT (tentative list). This is easily achievable using yahoo finance API or Python library yfinance. Further feature engineering, such as creation of volatility index, returns, etc., will be conducted to increase model performance.

The dataset will be a pandas dataframe indexed by (1) timestamps with unit of one trading day, ranging from Apr. 1st 2020 (post 2020 crash) to Dec. 31 2021 and (2) stock tickers. Other than the basic open, close, high, low, the following indicators would also be included (tentative list):

- *OHLC Volatility.* OHLC volatility is a aggregate function on open, close, high and low:

  Figure 1: Garman and Klass OHLC Volatility

  $$\sigma_{gk}^2 = \frac{1}{T}\sum_{t=1}^{T}\left(0.511(\ln(H_t/L_t))^2 - 0.019\ln(C_t/O_t)\ln(H_tL_t/O_t^2) - 2\ln(H_t/O_t)\ln(L_t/O_t)\right).$$

- *Open hour return.* The return in the first trading hour of the day.

- *Boolean: return rolling bollinger band out of bound.* A bollinger band (BB) of a time series is generated by

$$UB = \mu(d) + k\sigma(d)$$
$$LB = \mu(d) - k\sigma(d)$$

  where $\mu(d)$ is rolling mean with window d and $\sigma(d)$ is rolling standard deviation with window d. If a time series is mean reverting, one would expect the series to have negative change should it exceed the upper band, and to have positive change should it exceed the lower band.

- *TBD.* More will be added after EDA.

Hence the dataset is expected to have around (#tickers)×450 ≈ 2500 rows, and approximately 10 columns.

4 Solution Statement

To see whether my project can predict stock market down days, I will design a classifier trained in past time series, validated on out-of-sample future observation with respect to the training sample. If the model succeeds in predict some significant downside movements in the stock market, then it would grant an investor some advantage to reduce downward risk while adjust her/his position only minimally.

The project will utilize a XGBoost model trained *from scratch*, as compared to more computationally complex models such as various deep learning models, the computational power required to train an XGBoost ensemble is much lower; on the other hand, there does not really exist a well-performing canonical model for this particular industrial usage – due to the high efficient of financial markets, the "benchmark" predictive model usually lose their lead relatively quickly, as other developers/investors can employ a similar strategy as the benchmark and trade away the advantage.

After obtaining a tentative implementation of the training program, the workflow will be relocated to AWS. Firstly, unlike some of the projects completed on Udacity, the data used in this project does not require upload to s3, as compared to large datasets such as serialized images, the size of financial dataset is small and hence they can be pulled directly from yahoo finance before training. After configuring training procedure, the program will interact with AWS Sagemaker to tune hyperparameters such as booster learning rate, max tree depth, mininimum child tree weight, alpha and gamma of XGBoost model. Finally, after obtaining tuned model, I will deploy the model to an Sagemaker endpoint. After deployment, I will invoke the endpoint and pass in the same raw market data (since financial data is scarce) and let the endpoint make its predictions for the future movement of the selected stock. As a curiousity-oriented bonus, I will use the predicted movement to simulate a portfolio, and check its return after the trading period of the input data.

5 Benchmark Model

Using tree-based ensemble model to predict stock price is not uncommon. Personally, in a computational finance undergraduate class, I used random forest to read technical indicators such as RSI, OBV, MACD, stochastic oscillator, and William's R to forecast the daily movement of certain stocks, and created a strategy that yielded 14% return from Jan. 1st to Dec. 31st 2020. The study I conducted before was inspired by a applied mathematics paper by Khaidem, Saha and Dey in 2016, who used as well random forest on a series of technical indicators to predict the movement of the stock returns. Although not planning on using pre-existed technical indicators in this studies, nor the actual random forest model, what I do in this project is extremely similar to that of this paper: on one hand, the features I will be generating are homomorphic to established financial technical indicators – they are all just aggregate functions that take volume, open, high, low and close as arguments; on the other hand, the Xgboost model I will be using is a improved version of random forest – instead of treating each decision tree equally, the Xgboost assigns weights to different trees to achieve a better performance.

6 Evaluation Metrics

The evaluation metrics are as follows:

- *Accuracy*: As a classic evaluation metric for classifiers, accuracy measures how many trading days to we accurately predict the movement of the stock market. (86% in the paper).

- *Precision*: Since we want to manipulate the portfolio minimally, we put more weight on correctly predicting the down days of the stock. Hence, precision that measures the proportion of only one class is as well a metric to be considered. (88% in the paper).

- *Portfolio returns*: After all, the purpose of this exercise can be only employed in financial investment, where the most important measurement is the final return of the portfolio. After our model makes classifications, we can as well evaluate the model based on how much the portfolio grew. (14% in previous study which operated on technical indicators instead of raw market data.)

7 Project Design

After ingesting data, I will conduct exploratory data analysis on the raw data, and generate some preliminary features based on economic intuition, such as daily return and rolling volatility. As the project progresses into modeling section, I will as well try to add additional features if needed. For modeling, I will chose some tree-based ensemble classifier, such as random forest or Xgboost as the baseline model, and try to ameliorate the performance through hyperparameter tuning. For the evaluation metric, I will chose the classification precision of predicting negative return day (threshold pending) as one metric, and as well a customized loss function that especially punishes erroneous classifications. To sum up the project design, the workflow is as follows:

- *Data ingestion*: ingest data source; write functions/scripts to pull data from web directly, so that s3 is not needed; build and organize pandas dataframe; cleaning the data; save data.
- *EDA*: conduct exploratory data analysis for own understanding, such as visualization, displaying summary statistics, etc.
- *Feature engineering*: through insights gained in EDA, construct additional features to input into the model; potentially use aggregation methods to conduct dimensionality reduction (PCA).
- *Incorporating AWS*: relocate the benchmark model unto AWS Sagemaker; adapt the data ingesting / model training / hyperparameter tuning pipeline to sagemaker scripts such that the model can be deployed to sagemaker endpoint.
- *Training*: construct functions/scripts to fit desired features into the model; conduct hyperparameter tuning/grid search to obtain best performing model.
- *Deployment*: deploy the model to sagemaker endpoint.
- *Forecasting*: feed the raw stock data into the deployed endpoint to obtain movement predictions. Save predictions into dataframe.
- *Evaluation*: output prediction precision, and additional insights such as backtest simulation to show model's performance as a robot-trader deployed in the market.

References

[1] Malkiel, B. G. (2003). *A random walk down wall street.* W.W. Norton & Company.

[2] Khaidem, L., Saha, S., &; Roy Dey, S. (2016, April 29). *Predicting the direction of stock market prices using random forest.* https://arxiv.org/abs/1605.00003. Retrieved January 10, 2022, from https://arxiv.org/abs/1605.00003

[3] Tin Kam Ho. (1995). Random decision forests. *Proceedings of 3rd International Conference on Document Analysis and Recognition.* https://doi.org/10.1109/icdar.1995.598994

[4] Breiman, L. (2001). *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/a:1010933404324

[5] Chen, T., &; Guestrin, C. (2016). XGBoost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* https://doi.org/10.1145/2939672.2939785

[6] Dey, S., Kumar, Y., Saha, S., &; Basak, S. (2016). Forecasting to Classification: Predicting the direction of stock market price using Xtreme Gradient Boosting. researchgate . Retrieved January 15, 2022, from https://www.researchgate.net/profile/Snehanshu-Saha/publication/309492895_Forecasting_to_Classification_Predicting_the_direction_of_stock_market_pric
to-Classification-Predicting-the-direction-of-stock-market-price-using-Xtreme-Gradient-Boosting.pdf