JPX Tokyo Stock Exchange Prediction

Project of *Introduction to Statistical Learning*

Minxuan Chen Yi Zheng

June 19, 2022

Abstract

Portfolio building is one of the major goals of quantitative trading. In our work, several related classical models and machine learning methods on this topic are reviewed and analyzed. Besides, we build two models, using ridge regression and LightGBM to fit our time series data, and finally give predictions and discussion.

1 Introduction of the Problem

1.1 Statement of the Problem

Japan Exchange Group, Inc. (JPX) is a holding company operating one of the largest stock exchanges in the world, Tokyo Stock Exchange (TSE), and derivatives exchanges Osaka Exchange (OSE) and Tokyo Commodity Exchange (TOCOM).

The goal of the problem is to build portfolios from the stocks that are eligible for predictions (around 2000 stocks from JPX) in order to obtain higher returns and lower risks. Specifically, we need to rank the stocks from highest to lowest expected returns and the difference in returns between the top and bottom 200 stocks will be used to evaluate the model.

We have access to financial data from the Japanese market, such as stock information, historical daily stock prices (including secondary stock prices) as well as some option prices.

1.2 Evaluation Metrics of the Model

Once a model is built, it will use the closing price ($C_{(k,t)}$) until business day (t) and other data every business day as input data for a stock (k), and predict rate of change ($r_{(k,t)}$) of closing price of the top 200 stocks and bottom 200 stocks on the following business day ($C_{(k,t+1)}$) to next following business day ($C_{(k,t+2)}$).

$$r_{(k,t)} = \frac{C_{(k,t+2)} - C_{(k,t+1)}}{C_{(k,t+1)}}$$

Within top 200 stock predicted (denoted as up_i $(i = 1, 2, \ldots, 200)$), multiply by their respective rate of change with linear weights of 2∼1 for rank 1∼200 and denote their sum as S_{up}.

$$S_{\text{up}} = \frac{\sum_{i=1}^{200} \left(r_{(\text{up}_i,t)} * \text{linearfunction}(2,1)_i\right)}{\text{Average}(\text{linearfunction}(2,1))}$$

Similarly, within bottom 200 stocks predicted (denoted as down_i $(i = 1, 2, \ldots, 200)$), multiply by their respective rate of change with linear weights of 2∼1 for bottom rank 1∼200 and denote their sum as S_{down}.

$$S_{\text{down}} = \frac{\sum_{i=1}^{200} \left(r_{(\text{down}_i,t)} * \text{linearfunction}(2,1)_i\right)}{\text{Average}(\text{linearfunction}(2,1))}$$

The result of subtracting S_{down} from S_{up} is R_{day} and is called"daily spread return".

$$R_{\text{day}} = S_{\text{up}} - S_{\text{down}}$$

The daily spread return is calculated every business day during the public/private period and obtained as a time series for that period. The mean divided by standard deviation of the time series of daily spread returns is used as the score. Score calculation formula (x is the business day of public/private period):

$$\text{Score } = \frac{\text{Average}\left(R_{\text{day}_1-\text{day}_x}\right)}{\text{std}\left(R_{\text{day}_1-\text{day}_x}\right)}$$

1.3 Significance of the Problem

Machine learning is revolutionizing every aspect of our lives, including the financial world. In the past, some might hold the point of view that econometrics is enough for financial analysis. However, with the explosion of data, people have to try new methods to adapt to this change. Today, more than 70% of global capital transactions are carried out by computers or programs, and half of them are handled by quantitative or programmed managers.

Therefore, choosing this problem can help us conform to the trend of the times, improving our understanding of quantitative trading and experiencing the whole process of quantitative trading, which is of good practical significance.

2 Background and Related Work

The background of the problem is actually quantitative trading. Quantitative trading is mostly based on mathematical model instead of subjective judgements of human, using computer technology to select various "high probability" events that can bring excess returns (ER, often called alpha return in the industry) from huge historical data to formulate strategies, which greatly reduces irrational investment decisions when the market is extremely frenetic or pessimistic.

One of the major goals of quantitative trading is exactly building portfolios, which is what we aim to do in this problem. To achieve this goal, many experts have proposed different asset pricing models since 1950s in the US, such as modern portfolio theory (MPT), capital asset pricing model (CAPM) and single/multi-factor model, all of which are foundations of today's quantitative trading. Besides, in the 21st century, the booming machine learning and deep learning methods have provided most powerful tools for traders to make use of those classical models and find the optimal solutions.

2.1 Modern Portfolio Theory

H. Markowitz (1952) first proposed a method in portfolio building [1]. In his work, he studied what kinds of assets investors should choose as their investment objects, and how much the investment in various assets should account for in the total investment, so that investors can obtain the highest return at a certain risk level, or the least risk at a certain return level.

The combination with the largest return under various risk levels is called the efficient combination, and the set of all efficient combinations is the efficient frontier of the combination. Therefore, the key issue of asset selection is how to establish an efficient frontier for investors in the face of a large number of investment objects.

Like other economic models, this model is based on a series of assumptions, which mainly include:

- The securities market is completely efficient;
- Securities investors are rational;
- The return on securities is random variables whose properties are described by the mean and variance;
- The returns of securities are normally distributed;
- Each asset can be divided infinitely;
- There are no taxes and transaction costs.

The Markowitz portfolio model uses variance as a measure of risk. Variance has good mathematical properties. When the variance is used to measure the total risk of financial and asset portfolios, the variance of the portfolio can be decomposed into the variance of the individual asset returns in the portfolio and the covariance between the returns of each asset.

Based on all the assumptions above, the method can be summed up as a quadratic programming problem.

- A portfolio can be represented by a vector $\vec{P} = (w_1, w_2, \ldots, w_n)$, where w_i is the proportion invested in asset i, and therefore $\sum_{i=1}^{n} w_i = 1$.
- Rate of return is a high dimensional random variable $\vec{R} = (r_1, r_2, \ldots, r_n)$.
- Rate of return of the portfolio should be $R_P = \sum_{i=1}^{n} w_i r_i$.
- Variance of rate of return is used as a measurement of risk, which is $\text{var}(R_P) = \sum_{i=1}^{n} \sum_{j=1}^{n} w_i w_j \,\text{cov}(r_i, r_j)$.

Given the expected rate of return μ, the problem becomes:

$$\min_{\vec{P}} \text{var}\,(R_P) = \sum_{i=1}^{n} \sum_{j=1}^{n} w_i w_j \,\text{cov}\,(r_i, r_j)$$
$$\text{s.t. } \mathbb{E}\,(R_P) = \sum_{i=1}^{n} w_i \mathbb{E}\,(r_i) \geq \mu, \sum_{i=1}^{n} w_i = 1$$

If we randomly generate a large number of weight combinations and allocate the stocks according to these weight combinations, then we can get the corresponding risks and returns of each combination. The boundary of the points would look like this:

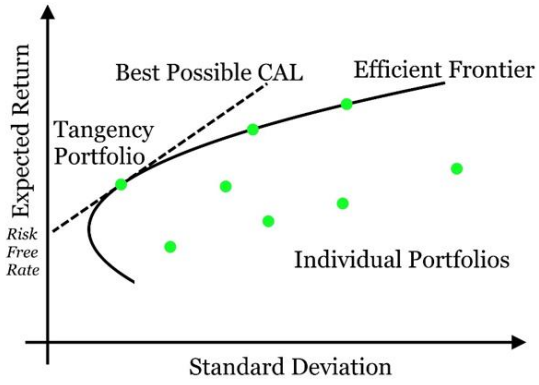

Figure 1: "Markowitz bullet", efficient frontier of different combinations

2.2 Capital Asset Pricing Model

Based on portfolio theory, W.F. Sharpe (1964) developed the capital asset pricing model (CAPM), the cornerstone of modern finance [2]. The capital asset pricing model gives a concise and accurate description of the relationship between the risk of securities assets and its expected rate of return. It is the backbone of modern financial market price theory and has been widely used in investment decision-making and corporate financial management.

Several assumptions need to be satisfied:

- Mean-variance efficient assumption. The expected return and standard deviation of the portfolio in a single investment period are used to evaluate different investment portfolios. For a certain rate of return, if the standard deviation of one risky asset portfolio is less than the standard deviation of any other risky asset portfolio, it is the average value-variance validity combination.
- Frictionless market assumption. No transaction costs, no taxes in the asset market, no limit to short-selling, and the trading asset is completely divisible.
- Riskless asset assumption. All investors can use the same risk-free interest rate to borrow or lend money.
- Homogeneous beliefs assumption. Investors' return on various assets, standard deviation and covariance all have the same expectation.
- The investment period is the same for all investors, and all information is free and immediately accessible.

CAPM expresses the relationship between the rate of return of any risky asset relative to the rate of return of the market portfolio:

$$\mathbb{E}(r_s) = r_f + \beta_s (\mathbb{E}(r_M) - r_f)$$

- r_s is the rate of return of a specific risky asset;
- r_f is the rate of return of a risk-free asset;
- r_M is the rate of return of the market portfolio;
- β_s is the sensitivity of the risky asset relative to the market portfolio, which can be computed as $\beta_s = \frac{\text{cov}(r_s, r_M)}{\text{var}(r_M)}$. It reflects the correlation between the return of the risky asset and the fluctuation of the return of the market portfolio. During a bull market, buying a stock of $\beta_s > 1$, one may make more than the average market return; during a bear market, holding a stock of $\beta_s < 1$, one can avoid

excessive losses; if the $\beta_s < 0$, the risky asset can help the investor make money during a bear market (such as shorting a stock).

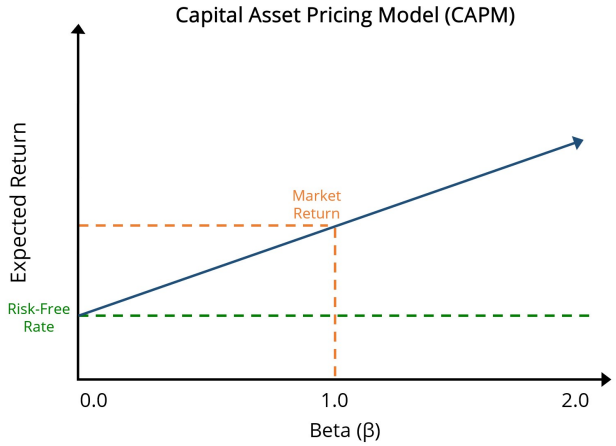

Figure 2: expected return and beta in CAPM

Later, many have tried to relax the assumptions in the model. F. Black et al. (1972) showed that in the absence of pure risk-free assets, the capital asset pricing model still holds [3], and this model is often referred to as a two-factor model. S. A. Ross (1977) showed that it would not be possible to obtain a portfolio with a beta of zero if there exists short sales [4]. This means that CAPM is only established in two cases: first, short selling is not allowed, but there are risk-free assets; second, there are no risk-free assets, but short selling is allowed.

In real application, other forms of CAPM model are also developed, such as non-linear CAPM, conditional CAPM, intertemporal CAPM, etc.

2.3 Multi-Factor Model

As stated in the last part, Black CAPM model is also referred to as two-factor model, then it naturally comes to multi-factor model. A multi-factor model is a combination of various elements or factors that are correlated with asset returns. The model uses said factors to explain market equilibrium and asset prices. In multi-factor models, different factors are associated with certain characteristics (such as risk), and it helps determine the weight or importance of that factor when computing asset price or return. Broadly, three types of multi-factor models can be classified based on the type of factors employed: macroeconomic factor models, fundamental factor models, and statistical factor models.

Eugene F. Fama et al. (1993) discovered the three-factor model, an improved theory of the capital asset pricing model [5]. The model is a linear regression model

with three independent variables, namely the price-to-book ratio, the company size, and the return of the market portfolio. The dependent variable is the return of the stock. Fama found that the unpredictable stock returns can be explained by these three simple data with small residuals. In short, with these three factors, you can roughly estimate the expected return of a stock. The proposed model is based on the empirical research results of the historical returns of the US stock market, and the purpose is to explain which risk premium factors affect the average return of the stock market.

The model can be expressed as:

$$R_{it} - R_{ft} = a_{it} + \beta_1 (R_{Mt} - R_{ft}) + \beta_2 SMB_t + \beta_3 HML_t + \epsilon_{it}$$

where:

- R_{it} = Total return of a stock or portfolio i at time t
- R_{ft} = Risk-free rate of return at time t
- R_{Mt} = Total market portfolio return at time t
- $R_{it} - R_{ft}$ = Expected excess return
- $R_{Mt} - R_{ft}$ = Excess return on the market portfolio
- SMB_t = Size premium (small minus big)
- HML_t = Value premium (high munus low)
- β = Factor coefficients

It is a better approach than the Capital Asset Pricing Model (CAPM), as CAPM only explains 70% of a portfolio's diversified returns, whereas Fama-French explains roughly 90%.

Therefore, after the three-factor model was proposed, experts began to explore multi-factor model and tried to improve the three-factor model. The improvement went mainly in two directions: one is to increase the factor, the other is to enhance the predictive model, applying machine learning and neural networks more and more.

Along the first direction, more models were built. Mark M. Carhart (1997) proposed four-factor model, which is based on the three-factor model, adding a factor called momentum [6]. The concept of the momentum of an asset can be used to predict future asset returns. It is a bit controversial, as it uses risk-based, as well as behavioral-based, explanations to determine returns. The Cahart model is considered a superior one, given its explanatory power of around 95%. In 2015, Fama and French proposed their five-factor model [7]. The Fama-French five-factor model also builds on the three-factor model and introduces two more factors: profitability (RMW) and investment (CMA).

It uses the return of stocks with high operating profitability minus the return of stocks with low or negative operating profitability.

In 2014, Campell R. Harvey et al. reviewed more than 300 factors that had ever appeared in literature [8]. Today, more and more factors are being discovered and used in real transactions.

2.4 Machine Learning Approaches

If we jump out of quantitative trading, this problem is about financial times series forecast or just time series prediction. From this perspective, there are mainly three methods. The first is traditional statistical models like ARIMA or Holt-Winters. The second is classical machine learning based on feature engineering which core idea is to convert time series forecast to supervised learning. The last is deep learning like DeepAR, LSTM, WaveNet and Transformer.

Using machine learning (ML) to forecast time series has been around for 20 years. In 2001, the usage of SVR by Francis E.H Tay and Lijuan Cao may be the first attempt of ML in financial times series forecast [9]. They use RBF as kernel and extract 5 indicators to predict the change of price. The process of creating features, selecting kernels and tuning was the paradigm at that time and has long been the primary choice. In 2010s, traders found that tree based model was useful in transaction since they were more interpretable. In 2018, Laura Alessandretti et al. used gradient boosting decision trees (GBDT) on cryptocurrency prices forecast [10]. They want to forecast return on investment (ROI) of day t by data from $t-w$ to $t-1$. They use predicted ROI to build portfolios and use some indicator like Sharpe ratio to estimate the model performance. From the perspective of ideas and methods, this work is not novel, but their process of combining machine learning and quantitative trading is clear and close to real trading, which has guiding significance for our task.

3 Model Selection and Building

3.1 Model Selection

In real transactions, traders usually treat the portfolio building problem as a classification or regression problem. Here, in our problem, we think that it is better to treat it as a regression problem rather than a classification problem.

Treating portfolio building as a classification problem do bring some benefits, for example, smaller computational load and relatively higher accuracy. If using classification, we may label those stocks with high return as class “+1” and label those with negative return as class “-1”, and those between them can be labeled as class “0”. The

proportion of the three categories in the training set can be determined as needed. By training such a classifier, we may buy those stocks that are in predicted class "+1" and sell those in predicted class "-1". And this method is actually being used by some quantitative investment company.

However, in our problem, this method may not work and we have to take the other approach for the following reasons. First, the number of stocks classified into the three classes is indeterminate. For example, we may get the result that there are 190 stocks in class "+1", 210 in class "-1", but we need 200 in each of the two classes. What's more, even if we get exactly 200 stocks in each of the two classes, we cannot determine which one should rank 1 and which one should rank 200, as stocks with different rankings have different weights to buy. Therefore, by training a regression model, we may get the predicted rate of return and sort the stocks according to this.

Since we've determined that the problem should be a regression problem, we naturally turn to multi-factor model, which is suitable for regression methods. Therefore, the main task becomes factor mining, placing great demands on our feature engineering process.

3.2 Feature Engineering

Since this task is not only a supervised learning but also a time series prediction, we must consider its chronological character. We don't use methods like ARIMA in time series analysis (TSA). So generally, there are two kinds of features. One is cross-sectional and the other is time-series. As for cross section, some intuitive idea is to use mean, minimum or maximum of prices, i.e. some simple combination and calculation. We also need financial factors mentioned above to bring in expertise. However, many useful factors are not just cross section. They consider present and history simultaneously, so these factors are mixed. Furthermore, we want an "one for all" model, so there must be features reflecting differences between stocks. As for time series, a significant and effective approach is lag feature. We directly use values at prior timestamps as variables and the length of forward time window can be set manually. Moreover, seasonal and period patterns are also common in time series. To recognize them, we can add some categorical variables representing week, month or quarter.

In summary, we've made features in the following aspects:

- Raw data, containing high, low, open, close and volume.
- Intuitive transformation, like taking logarithm.
- Coded categorical variables, using different methods like one hot encoding, statistics based.

- Quantitative finance factors.
- Lag features.

It's worth mentioning that due to our "one for all" model, we have to calculate features for each stocks and then combine them while preserving the original time order.

To compare the effect of features, we use `Pipeline` and `FeatureUnion` from package `sklearn` to customize our feature engineering.

3.3 Cross-validation on Time Series

For classification or regression problems with time characteristics, stratified k-fold or k-fold cannot simply be used for cross-validation, let alone shuffle, which will bring certain problems of time series feature intersection, such as using future data to predict the past data. Such cross-validation results are of little business significance, and it is easy to cause overfitting.

Thus, in our work, we did cross-validation on our time series data as follows. This method is called `TimeSeriesSplit`, and can be used from package `sklearn`. It will generate folds over a sliding window over time. The length of the training sequence grows over time, with each subsequent fold retaining the full sequence history, while the test sequence length for each fold is constant. In our implementation, it is slightly different since we want the stocks in training and validation sets are complete over the time period. That is, if day t is the last day in training set, we want all stocks in day t are included. So it's more like a "Group-`TimeSeriesSplit`", not just cutting by time window.

Our training and validation set is from 2017/01/04 to 2021/12/03 and test set is from 2021/12/06 to 2022/04/28. Every day, there are about 2000 stocks.

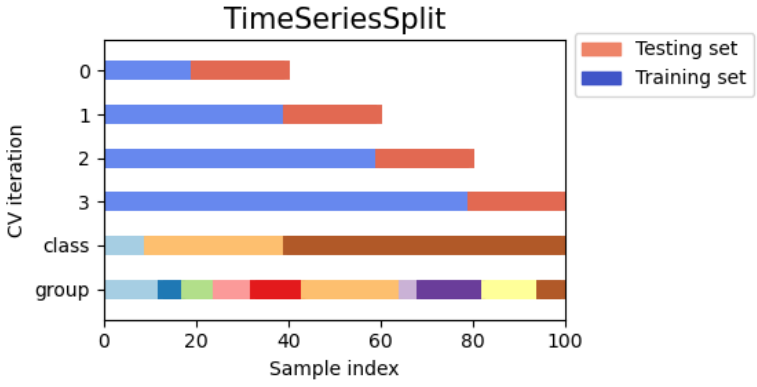

Figure 3: cross-validation on our time series data

3.4 Training Methods

Generally, our prediction target is $r_{k,t}$ mentioned above and we use sum of squared error (SSE) as our metric.

3.4.1 Ridge

We use ridge regression as baseline. Considering the limited ability of linear models, especially for transaction data with high noise, we classify stocks by industry and for each industry, we train a model. Actually this is an implementation of multi-factor model.

3.4.2 LightGBM

In practice, tree-based methods are favored in finance. We train an "one for all" model with categorical variables that distinguish industry, indices, market, etc. We use package `optuna` to automatically tune the parameters.

4 Results and Comparison

4.1 Ridge

Table 1: Result of ridge regression

penalty α	SSE
0.001	134.743
0.01	131.592
0.1	124.350
1	121.065

It's easy to predict that this model won't perform very well since we don't have features tell the difference between stocks and the relationship between industries is weaken (actually disappears) as well.

Here we choose some stocks of different industries and plot their true target value and predicted value over time period of the test set.

As we regard ridge regression as baseline, we don't do serious cross-validation. We try different α penalty on full training set and calculate SSE on test set. Here is the result.

First, the reason why we use SSE is because actually from the plots linear model is pretty much useless in the scene. However, due to the fact that sample size is very

Figure 4: comparison between different α (upper: α=0.001, lower: α=0.01)

large (200,000), MSE can still be small (1% of target value), causing the illusion that the model performs well.

Second, it's evident that with penalty α increasing, SSE decreases. However, it's strange to see from the plots when α is relatively large, predicted values can show some trend consistent with true values. We think the reason is large α will force the coefficients to zero, so the fitted value will be just the intercept, i.e. mean of true value. Moreover, since we actually train a model for an industry, not for one specific stock, it is no surprise that the mean is around zero.

In summary, we think that high noise in trade data greatly affects the linear model and the large number of explanatory variables and severe multicollinearity restrict its performance as well.

4.2 LightGBM

Here are plots of predicted value and true value given by `LightGBM`. We can find the curves are no longer as flat as ridge regression, which shows that the boosting method has stronger fitting ability. Actually, this is understandable because boosting method will pay more attention to the parts with poor fitting results.

The SSE is 110.920.

In summary, we think the fitting ability of boosting method is enough, however using one model to capture patterns of all stocks is much too ideal. We need multiple models responsible for different signals.

Figure 5: results of LightGBM

5 Data

Our data can be accessed through this link.

6 Author Contributions

Minxuan did feature engineering and parameter tuning, and mainly wrote the "model selection and building" and "results and comparison" parts of this report; Yi did model fitting and parameter tuning, and mainly wrote the "introduction of the problem" and "background and related work" parts of this report. Other parts are done by co-writing.

References

[1] H. Markowitz. Portfolio selection. *The Journal of Finance*, 7:77–91, 1952.

[2] W.F. Sharpe. Capital asset prices: A theory of market equilibrium under conditions of risk. *The Journal of Finance*, 19:425–442, 1964.

[3] Fischer Black, Michael C. Jensen, Myron Scholes, et al. The capital asset pricing model: Some empirical tests. 1972.

[4] Stephen A. Ross. The determination of financial structure: the incentive-signalling approach. *The bell journal of economics*, pages 23–40, 1977.

[5] Eugene F. Fama and Kenneth R. French. Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, 33(1):3–56, 1993.

[6] Mark M. Carhart. On persistence in mutual fund performance. *The Journal of Finance*, 52(1):57–82, 1997.

[7] Eugene F. Fama and Kenneth R. French. A five-factor asset pricing model. *Journal of Financial Economics*, 116(1):1–22, 2015.

[8] Campbell R. Harvey, Yan Liu, and Heqing Zhu. . . . and the cross-section of expected returns. Working Paper 20592, National Bureau of Economic Research, October 2014.

[9] Francis E.H Tay and Lijuan Cao. Application of support vector machines in financial time series forecasting. *Omega*, 29(4):309–317, 2001.

[10] Laura Alessandretti, Abeer ElBahrawy, Luca Maria Aiello, and Andrea Baronchelli. Anticipating cryptocurrency prices using machine learning. *Complexity*, 2018, 2018.