Portfolio Analysis

Yunran Chen

1 Introduction

The project aims to create a profitable investment portfolio with consistent returns while managing risks effectively using provided datasets. I chose a security reference dataset, a risk factor dataset, and imported a risk-free dataset from an external source. To start, I performed data preprocessing to filter securities and time periods. Afterward, I engaged in feature engineering and utilized XGBoost to choose variables and build a model for predicting returns.

Using the estimated returns, I constructed a portfolio employing the mean-variance model, taking into account a dollar neutral constraint and a trading cost constraint. The original dataset was divided into three parts for model training, hyperparameter tuning, and model evaluation. The resulting portfolio strategy demonstrated satisfactory performance.

2 Data Preprocessing

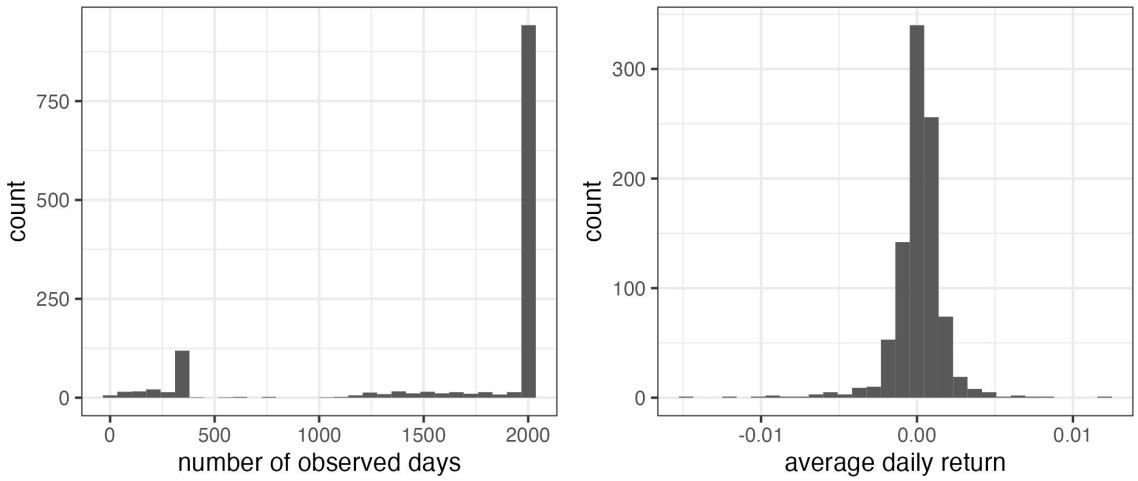

Figure 1: (Left) Distribution of observed days for the selected 1283 securities. (Right) Distribution of average daily returns for the selected 906 securities.

For simplicity, I'm focusing on the security reference dataset and risk factor datasets, as they are well-documented. In the data preprocessing stage, I combined and merged the

security reference dataset with the risk factor dataset. I specifically considered securities that are eligible for trading at all times, resulting in 1283 securities.

In Figure 1(left), you can see the distribution of the number of days each of these 1283 securities has been observed. Interestingly, 70.6% of these securities (906 out of 1283) have observations across 2013 days, spanning from 2010 to 2017. To ensure more complete data, I focused on these 906 securities for further analysis.

Figure 1(right) illustrates the average return weighted by volume for each security among the selected 906 securities. The graph indicates a symmetrical distribution around 0, with most average daily returns falling between -0.01 and 0.01.

3 Feature Engineering

To make accurate return predictions, it's essential to extract valuable features from the data. My features are sourced from three main places. First, I use features directly from the original dataset, such as the day's closing price (*close_price*), traded volume (*volume*), and six risk factors (*rf1* to *rf6*). Second, I derive new features from the existing data. This includes calculating moving averages over 5, 10, 15, and 20 days, representing weekly, bi-weekly, tri-weekly, and monthly averages. I also compute standard deviations over 10, 20, 60, 120, and 250 days, representing half-month, one-month, a season, half-year, and one-year volatility. Additionally, I incorporate lagged returns for 1 to 5 days and cumulative returns from historical data. To address seasonality, I create a new feature (*dayofweek*) based on the date of the data (*data_date*), capturing the day of the week for that date. I further generate four features from the 8-digit security group identifier (*group_id*), representing Level I (*l1*), Level II (*l2*), Level III (*l3*), and Level IV (*l4*) groupings. For example, if two securities share the same Level I value, their *l1* value should be identical. Third, I introduce the 1-month daily treasury rate as the risk-free rate from an external source[1]. The risk-free rate is divided by 250 to align with the scale of daily returns for securities.

4 Feature Selection and Return Prediction

XGBoost (eXtreme Gradient Boosting) is a powerful algorithm for implementing gradient-boosted decision trees. It's commonly used for predictive modeling in regression and classification scenarios, especially when dealing with large datasets and a high number of features. Built upon decision trees and gradient boosting, XGBoost combines multiple weak models to create a strong-performing model.

To train and assess the model, I divided the data into three sets: training (2010-01-01 to 2014-12-31), validation (2015-01-01 to 2015-12-31), and testing (2016-01-01 to 2017-12-31). The training set was used to fit the XGBoost model, and return predictions were made on the validation set. The hyperparameter (maximum depth of the tree) that minimizes the square root of the mean square error of prediction was selected. The final model used default settings, with a learning rate of 0.3 and a maximum tree depth of 6.

[1] `https://home.treasury.gov/resource-center/data-chart-center/interest-rates/TextView?type=daily_treasury_bill_rates&field_tdr_date_value_month=202311`

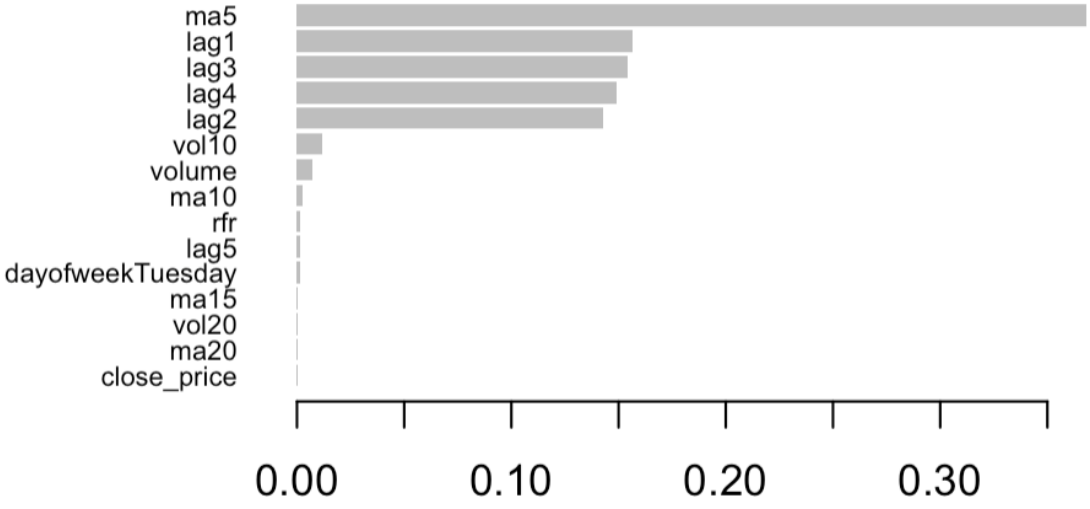

Figure 2: Feature importance of XGBoost

Figure 2 illustrates the important features for return prediction. The length of each bar indicates the relative importance of each feature. Notably, the moving average of returns within 5-day windows plays a crucial role in predicting returns. Lagged daily returns also emerge as important features. Features such as volatility over 10 days, trading volume, moving average of returns within 10-day sliding windows, and the risk-free rate, while less critical, still contribute to return prediction.

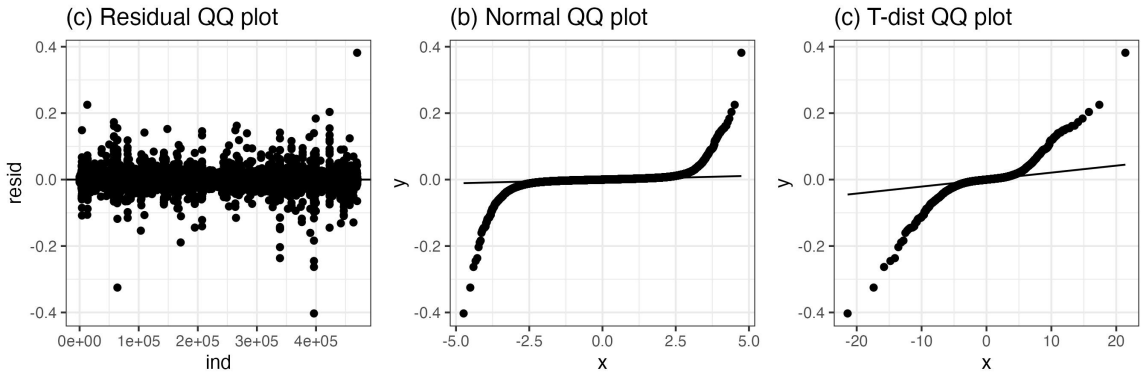

Figure 3: Summary plots for residuals

Figure 3 contains plots summarizing how well XGBoost performed on the training set. In Figure 3(a), there's a scatter plot of residuals, illustrating the differences between predicted and actual values. The majority of residuals are between -0.08 and 0.08, with only a few points significantly under or overestimated (outside the range of [-0.2, 0.2]). Approximately 98.99% of the points are within 3 times the standard deviation from the mean.

Figures 3(b) and (c) display QQ-plots of standardized residuals, comparing them against both normal and t-distributions (with degrees of freedom around 3). These plots indicate that the bulk of the residuals are more likely to be normal distributed, but the tails of the standardized residuals deviate from both the normal distribution and the t-distribution. This suggests that the tail of data follows a distribution with very heavy tails.

5 Portfolio Construction

I'm exploring a portfolio strategy that considers market-neutral constraints and includes transaction costs. In this approach, market neutrality assumes that the sum of weights for long positions equals 1, and the sum of weights for short positions equals -1. Transaction costs are represented as the L1-norm of the difference between current weights and lag-1 weights adjusted by lag-1 return.

To incorporate transaction costs into the objective function, I calculate daily portfolio return as the actual portfolio return minus trading costs. The market-neutral assumption is integrated as constraints in the optimization problem. The final model is expressed as a constrained optimization problem:

$$\begin{aligned} \max_{w_t} \quad & w_t^T r_t - \lambda w_t^T \Sigma w_t - \|w_t - w_{t-1} \odot (1 + r_{t-1})\|_1 \\ \text{s.t.} \quad & 1^T w_{t,+} = 1 \\ & 1^T w_{t,-} = -1 \end{aligned} \tag{1}$$

Here, w_t is a vector representing weights for each security in the portfolio at time t, and r_t is a vector indicating estimated daily return. The matrix Σ represents volatility, estimated by a sample covariance matrix from training data. The parameters w_{t-1} and r_{t-1} refer to lag-1 weights and lag-1 return, respectively. $w_{t,+}$ and $w_{t,-}$ denote the positive and negative parts of w_t, and λ is a tuning parameter reflecting risk aversion.

I conduct the analysis on a validation set to select the optimal value for λ. The trading frequency is simplified to weekly, with portfolios constructed only on Mondays, resulting in 47 time points in the validation set. Some missing values exist in both the training data and the validation set. For estimating Σ, I assume missing data are random in the training set, calculating the sample covariance matrix using all the observations for which both variables have valid values. For estimating r_t, predictions from a previously trained XGBoost model are used. Only one missing value in the data is imputed using the previously estimated return.

I consider four potential values for λ (1, 5, 10, 50). As λ increases, risk decreases, typically leading to lower returns. The choice of λ depends on an individual's risk tolerance. In this case, I opt for a moderate risk tolerance, choosing $\lambda = 10$ since the returns under $\lambda = 5$ and $\lambda = 10$ are similar, but the risk under $\lambda = 10$ is smaller.

I used a trained XGBoost model on test data to predict daily returns. Then, I applied a mean-variance model (with $\lambda = 10$) to the test set to create a portfolio.

6 Portfolio Assessment

The portfolio has an expected return of 19.71%, and risk of 0.076 (standard deviance). The percentage of profitable days is 100%, this suggests the strategy has relative high profits, relative low risk, and is always making money. The mean of Sharpe ratio is 2.57, suggests the risk-adjusted performance of portfolio is really good. Weekly turnover rate approximate 1 most of time, suggesting the optimal strategy is high frequency trading. Table 6 presents summary statistics for portfolio assessment, such as return, turnover, Sharpe ratio, cumulative return, and drawdown. Figure 4 presents daily return, daily cumulative return, Sharpe ratio, and turnover rate for portfolio at each day. Figure 5 presents histogram of these statistics.

Return	Turnover	Sharpe	Cumulative Return	Drawdown
Min.	0.03509	0.00	0.4138	0
1st Qu.	0.14685	100.87	1.9143	0
Median	0.18270	101.25	2.3728	0
Mean	0.19706	98.03	2.5711	0
3rd Qu.	0.22758	101.66	2.9614	0
Max.	0.47061	102.44	6.1465	0

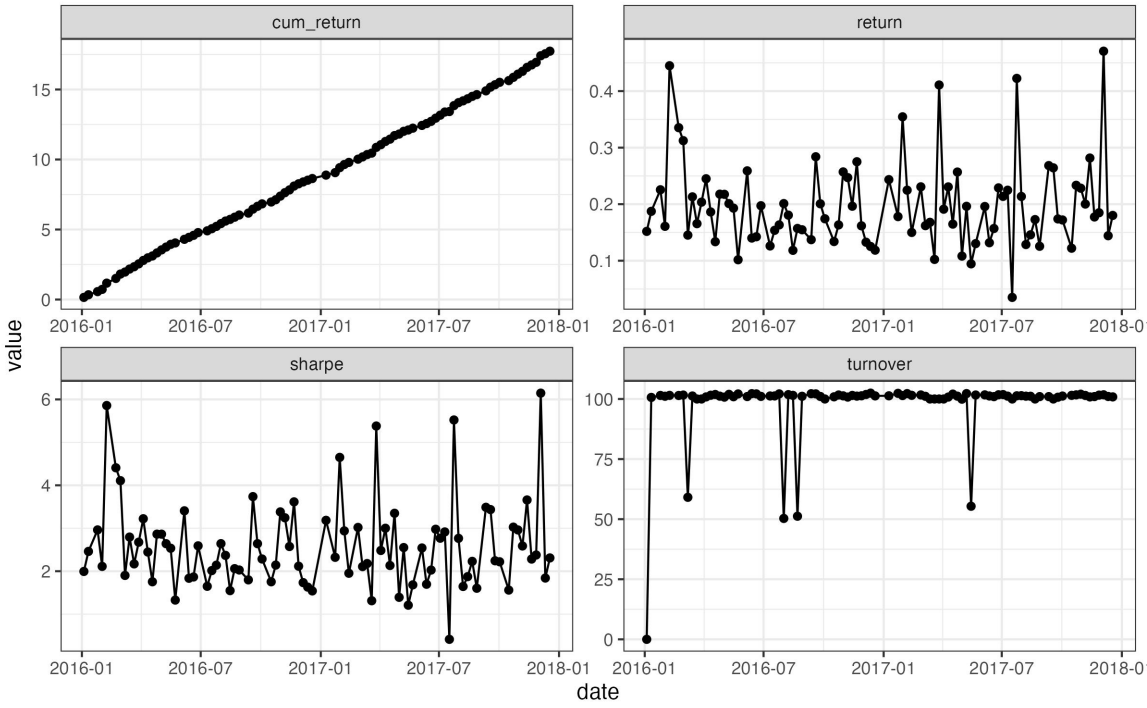

Figure 4: Portfolio Assessment. Plots represent cumulative return, return, daily Sharpe ratio, and turnover rate (%)

We can observe the cumulative return increase with time and can reach above 15 for

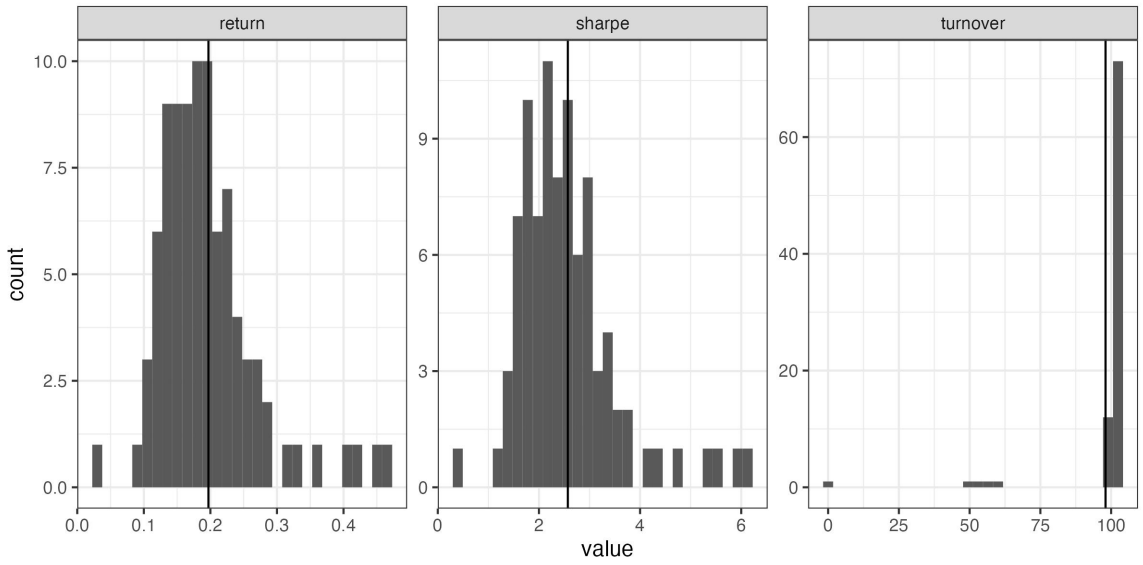

Figure 5: Histogram of return, Sharpe ratio, turnover rate

two years. And daily return is above 0.1 most of time. The Sharpe ratio is above 1 for most of time (good accept to), % is above 2, which is considered to be very good, % above 3 which is considered to be excellent. The portfolio strategy is high frequency trading have turnover rate around 100% for most time. It picks only a few securities and the dominant one switch from week to week (Port

7 Conclusion

The current solution incorporates trading costs into the objective function, promoting less frequent trading due to its impact. Additionally, we optimize the portfolio every five trading days, with rebalancing occurring on Mondays. This approach not only considers trading costs but also reduces computational expenses.

Another enhancement involves the estimation of risk (Σ) in equation (1). To streamline the model, we assume a constant variance by taking the sample variance of historical data (training set). An improvement would be to allow for time-varying risk estimation in the model, with a simple alternative being to calculate sample variance based on historical data using sliding windows.

In our analysis, we focus on the minimum constraints required by the problem settings. However, in practical scenarios, other considerations can be beneficial. For instance, incorporating a penalty for high turnover ($||w_t - w_{t-1}||_1 \leq \tau$) or adding a maximum position constraint to prevent the dominant holding of a particular security ($||w||_\infty \leq u$). Encouraging sparsity by limiting the number of nonzero positions ($||w||_0 \leq K$) is another useful consideration.

While the applied model is a mean-variance portfolio, practical performance can be influenced by skewness and kurtosis of financial returns. Considering more sophisticated

risk measures such as value-at-risk (VaR) and conditional value-at-risk (CVaR) could provide valuable insights. The choice of the risk aversion parameter λ depends on individual risk preferences.

It's important to note that the chosen dataset is a small subset of the provided data. The selection is primarily based on dataset completeness. Observations reveal that a few securities with records limited to a relatively short period exhibit very high returns, exceeding 0.1. Careful dataset selection could contribute to more profitable outcomes.