Time Series

Explore Stock Market

May 14, 2020

Luo Jinying

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

Contents

1 Introduction **1**

2 Data Description **1**

3 Forecasting **1**

3.1 ARMA model 1

3.2 ARMA-GARCH model 4

3.3 Neural Network (NN) Model 7

3.3.1 Introduction of NN 7

3.3.2 NN Model Application 8

3.4 Random Forest 9

3.5 Best Model Selection 9

4 Portfolio Construction **10**

5 Conclusion **12**

6 Appendix **13**

6.1 A. Time series plots of each stock 13

6.2 B. Model forecasting from Week 209 to Week 260 14

1 Introduction

Nowadays more and more people are investing their money in stock market. Both investors and speculators are exploring new trading strategies by analyzing market information. My research aims to figure out a suitable model to forecast log stock returns. Based on the model, we will construct a portfolio with high and stable excess return and low variance.

2 Data Description

The whole dataset consists of weekly log returns of SP500 index and 10 stocks for 260 weeks. There is no missing value. The general information of each stock is showed in the table below. Obviously, these 10 stocks have higher volatility than SP500 index. The mean of each stock is close to 0, especially for SP500 index.
The dataset is divided into training data set and test data set. Training data set includes stock weekly log returns from week 1 to week 208. Test set involves log returns from week 209 to week 260.

Table: Summary of each stock

Stock	SP500	Stock 1	Stock 2	Stock 3	Stock 4	...	Stock 9	Stock 10
Mean	-0.0007	-0.0042	0.0011	-0.0111	0.0014	...	-0.0019	0.0026
Sd.	0.0265	0.0900	0.0416	0.1767	0.0868	...	0.1124	0.0844

3 Forecasting

3.1 ARMA model

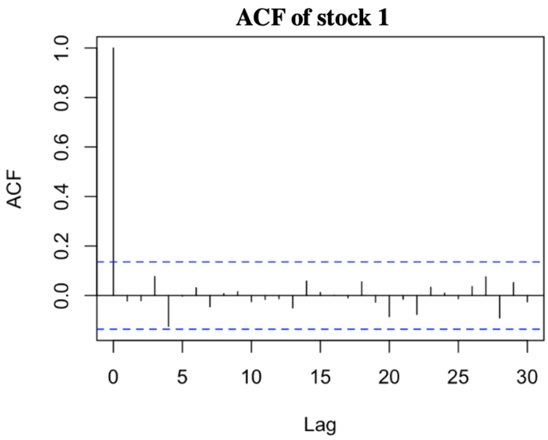
Step 1: **Plot time series diagrams (Appendix.A) to observe the pattern of each stock based on training data set**. The mean of weekly return is fairly close to 0 for each stock. In general, there is no obvious persistency or trends embedded of each stock. Log returns fluctuate around the constant mean. The data is roughly stationary by observation. To further check the stationary property, for each stock, we take difference of d=1,2,3 and plot the differencing time series plots. Compare these plots with the original time series plots. We find that the general pattern of series after doing difference does not change too much. Thus, to make the model concise, we do not take difference in this data set. However, there are several significant volatility clusters from week to week. It may indicate ARMA model is not enough to capture the whole whole relation of the series.

Step 2: **Plot sample ACF and PACF of each stock. Test unit roots**. As the value of ACF and PACF are all close to 0 and insignificant, it seems unnecessary to take difference, which coincide with the result in step 1. We also adopt augmented Dickey-Fuller test (ADF) and Phillips-Perron (PP) test, to test the null hypothesis that the log returns series has a unit root versus the alternate hypothesis that there is no unit root. Hopefully, all the p-value of the two methods indicates that there is no unit root and we can reject the null hypothesis.

Table: Results of unit roots test

p-value	SP500	Stock 1	Stock 2	Stock 3	Stock 4	...	Stock 9	Stock 10
ADF	<0.01	<0.01	<0.01	<0.01	<0.01	...	<0.01	<0.01
PP	<0.01	<0.01	<0.01	<0.01	<0.01	...	<0.01	<0.01

Step 3: **According to the plots of ACF and PACF, build ARMA(p,q) models and estimate the coefficients through MLE**. If ACF and PACF show some significant spikes, we can choose several models as the candidates accordingly. Nevertheless, in our training data set, ACF and PACF of each stock do not present significant spikes. Hence, we suspect that the model is supposed to be a parsimonious model. We select small value of p and q such as 0, 1 and 2. After that, the coefficients of the model is estimated by MLE. Each model will generate AIC value for each stock. Then, we sum up all the AIC values of all the stocks for each model. Select three models with the smallest total AIC as the candidates. From the results below, Compare total AIC of each model, we choose ARMA(1,0), ARMA(0,1) and ARMA(1,1) as the candidates.



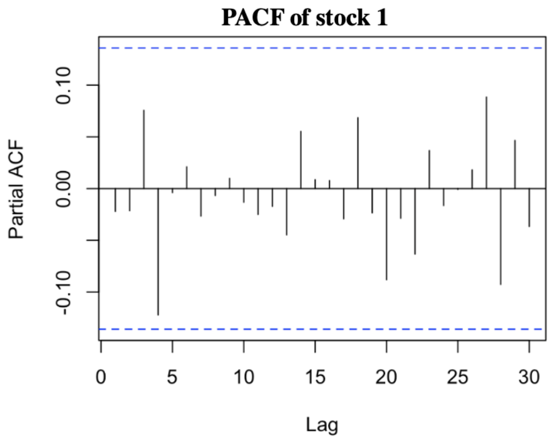
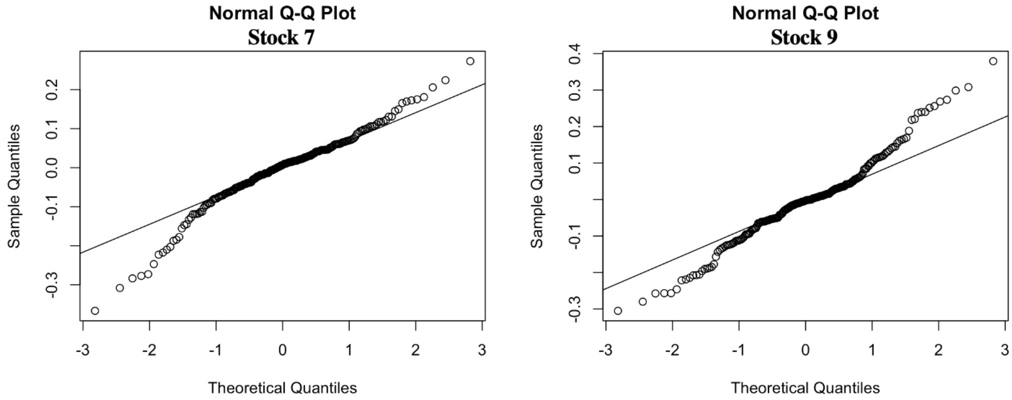


Table: Comparison of each model

ARMA(p,q)	Stk 1	Stk 2	Stk 3	Stk 4	...	Stk 9	Stk 10	Total AIC
ARMA(1,0)	-373.5	-699.4	-115.6	-398.6	...	-303.3	-396.6	-4615.3
ARMA(0,1)	...	...	...	...	...	...	...	-4614.7
ARMA(1,1)	...	...	...	...	...	...	...	-4613.4
ARMA(2,1)	...	...	...	...	...	...	...	-4602.3
ARMA(1,2)	...	...	...	...	...	...	...	-4602.8
ARMA(0,2)	...	...	...	...	...	...	...	-4609.4
ARMA(2,0)	...	...	...	...	...	...	...	-4609.2
ARMA(2,2)	...	...	...	...	...	...	...	-4612.3

Step 4:**Diagnose whether the residuals are iid normal distribution**. From normality perspective, no matter it is ARMA(1,0), ARMA(0,1) or ARMA(1,1), the Q-Q plots of most

of stocks in each model shows that the residuals do not follow normal distribution as the head and tail segment parts deviate a lot from the line. Take ARMA(0,1) model as an example. We can clearly see the evidence that the residuals of the following two stocks do not follow normality



From the independence perspective, the sample ACF of residuals for stocks 4,6,10 have several significant spikes in each model. It means that the residuals may not be independent. Although the results of Ljung-Box Q test showed in following table that the p-value of residual is larger than 0.05, squared residuals for stock 1, 7, and 8 are significant at 5% significance level, which also gives the evidence to support that the residuals are not independent. It indicates that GARCH model may be needed to capture the relationship.

Therefore, considering the normality and independence, the residuals do not follow iid normal distribution.

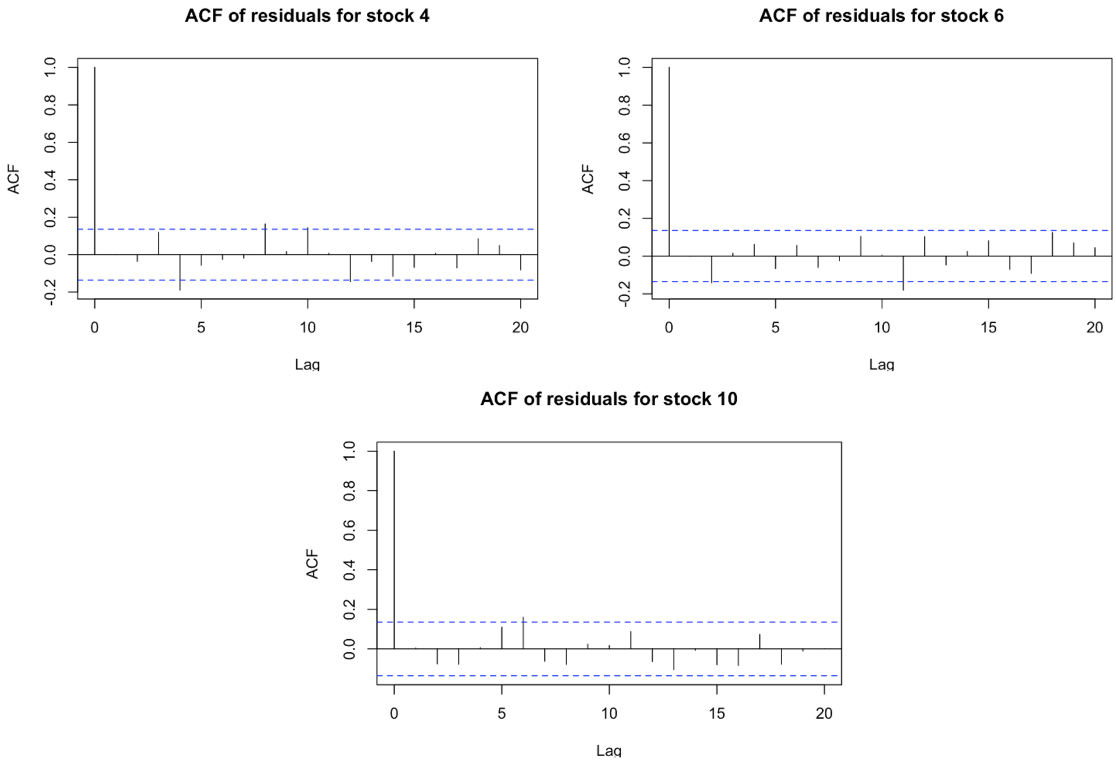


Table: Ljung-Box Q test for residuals

p-value	Stk 1	Stk 2	Stk 3	Stk 4	Stk 5	...	Stk 8	Stk 9	Stk 10
ARMA(1,0)	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9
ARMA(0,1)	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9
ARMA(1,1)	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8

Table: Ljung-Box Q test for squared residuals

p-value	Stk 1	Stk 2	Stk 3	Stk 4	...	Stk 7	Stk 8	Stk 9	Stk 10
ARMA(1,0)	0.04	0.8	0.8	0.8	...	0.002	0.0001	0.6	0.6
ARMA(0,1)	0.04	0.8	0.8	0.8	...	0.002	0.0002	0.6	0.6
ARMA(1,1)	0.04	0.8	0.8	0.8	...	0.003	0.0002	0.6	0.6

Step 5: **Model selection**. Although the residuals of the three candidates do not follow iid normality, we still need to choose the optimal one to capture the mean of the series. Apart from this, GARCH model may needed which is showed in ”3.2 GARCH” to capture the variance relation. Thus, for ARMA model, we select the ARMA(1,0) with smallest AIC from the three candidates as showed in step 3.

Step 6: **Performance evaluation**. We apply ARMA(1,0) model to have one-step, two-step and three-step forecasting i.e. h=1,2,3 on the test data. We adopt rolling scheme to test the models' performance. Starting from Week 208, each model will forecast the log returns of the 10 stocks based on the past 208 observations. Take one-step model as an example. If we would like to predict the log returns in Week 210, the information from Week 2 to Week 209 will be adopted to estimate the coefficients of the model. After Week 208, the forecasted log returns will be compared with the true log returns by calculating test MSE, which is a measurement of the performance for the model.

$$MSE(h) = \frac{1}{10(52-h+1)} \sum_{i=1}^{10} \sum_{s=n_1}^{n_1+52-h} (\hat{x}_{s,i}(h) - x_{s+h,i})^2$$

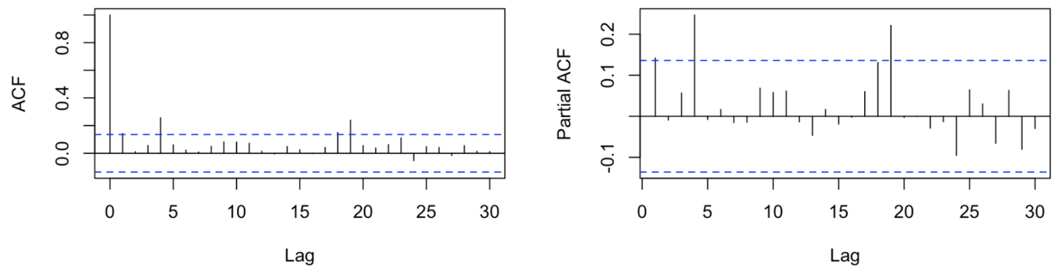
Table: Evaluation of ARMA(1,0)

MSE(h)	h=1	h=2	h=3
ARMA(1,0)	0.004546	0.004548	0.004537

3.2 ARMA-GARCH model

Typically, for financial series, the return, does not have a constant conditional variance, and highly volatile periods tend to be clustered together. From the time series plots of each stock (Appendix A.), we do observe there are several volatility clusters. There may exist a dependence of sudden bursts of variability in a return on the series own past. The ACF and PACF of the squared residuals below also appear that there may be some dependence left in the residuals. Furthermore, we have already applied Ljung-Box Q test to examine whether the squared residual has some dependence or not (in 3.1 ARMA Model - step 3). We figure out that squared residuals of stock 1,7,8 are statistically significant under 5% significance level in ARMA(1,0) model. All of the above evidence implies that the series has GARCH errors. So, it is necessary to include GARCH model to capture the dependent relationship.

ACF and PACF of the squares of the residuals from the ARMA(1,0) fit on stock 1



ACF and PACF of the squares of the residuals from the ARMA(1,0) fit on stock 2

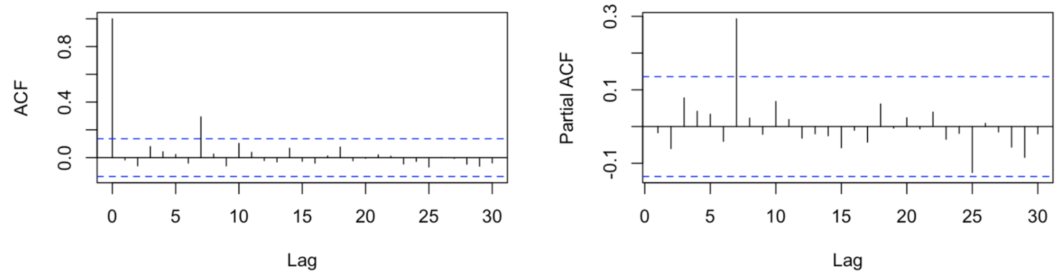


Table: Ljung-Box Q test for squared residuals in ARMA(1,0)

Stock	Stk 1	Stk 2	Stk 3	Stk 4	Stk 5	Stk 6	Stk 7	Stk 8	Stk 9	Stk 10
p-value	0.046	0.85	0.89	0.77	0.51	0.25	3.5e-4	2.2e-4	0.60	0.76

Step 1: **Biuld ARMA(1,0)-GARCH(p,q) model**. The most common setting for GARCH model is GARCH(1,0) and GARCH(1,1). Thus, we tried ARMA(1,0)-GARCH(1,1) and ARMA(1,0)-GARCH(1,0). From AIC and BIC criterion, the performance of these two models is fairly close. ARMA(1,0)-GARCH(1,1) performs sightly better in terms of AIC and BIC. We choose both of them as candidates.

Comparison of the two candidates

Model	Stock	AIC	BIC	SIC
ARMA(1,0)+ GARCH(1,1)	stk1	-1.95	-1.85	-1.95
	stk2	-3.42	-3.33	-3.42
	stk3	-1.27	-1.17	-1.27
	stk4	-2.40	-2.30	-2.40
	stk5	-3.58	-3.49	-3.58
	stk6	-2.80	-2.70	-2.80
	stk7	-2.11	-2.02	-2.12
	stk8	-3.35	-3.25	-3.35
	stk9	-1.47	-1.37	-1.47
	stk10	-2.02	-1.92	-2.02
	Total	-24.37	-23.40	-24.38
ARMA(1,0)+ GARCH(1,0)	stk1	-1.88	-1.80	-1.88
	stk2	-3.40	-3.32	-3.41
	stk3	-1.22	-1.14	-1.22
	stk4	-2.32	-2.24	-2.32
	stk5	-3.53	-3.45	-3.53
	stk6	-2.74	-2.66	-2.74
	stk7	-2.00	-1.92	-2.00
	stk8	-3.36	-3.28	-3.36
	stk9	-1.48	-1.40	-1.48
	stk10	-1.96	-1.87	-1.96
	Total	-23.89	-23.09	-23.90

Step 2: **Diagnose ARMA(1,0)-GARCH(1,1) and ARMA(1,0)-GARCH(1,0) models through Ljung-Box Q statistics for the squared standardized residuals series.** If there is no serial correlation in the squared standardized residuals series, it means that there is no conditional heteroskedasticity remaining in the residuals. The residuals can satisfy the condition of normality. From the table below, for ARMA(1,0)-GARCH(1,1) model, p-value is statistically insignificant at 5% significance level for most cases except for stock 2. The model can perform good enough to do estimation and forecasting. However, as for ARMA(1,0)-GARCH(1,0) model, half of the stocks are statistically significant at 5% significance level, which means this model cannot capture the volatility well. Thus, ARMA(1,0)-GARCH(1,0) cannot be chosen. The optimal model is ARMA(1,0)-GARCH(1,1).

Table: ARMA(1,0)-GARCH(1,1): Results of Ljung-Box Q test on squared residuals

p-value	Stk 1	Stk 2	Stk 3	Stk 4	Stk 5	Stk 6	Stk 7	Stk 8	Stk 9	Stk 10
h=1	0.99	1.2e-3	0.98	0.99	0.99	0.72	0.90	0.82	0.99	0.85
h=2	0.99	1.9e-3	0.99	0.99	0.07	0.82	0.79	0.99	0.93	0.96
h=3	0.82	0.01	0.99	0.99	0.17	0.75	0.92	0.99	0.88	0.98

Table: ARMA(1,0)-GARCH(1,0): Results of Ljung-Box Q test on squared residuals

p-value	Stk 1	Stk 2	Stk 3	Stk 4	Stk 5	Stk 6	Stk 7	Stk 8	Stk 9	Stk 10
h=1	0.07	4.6e-3	0.01	1.00	0.01	3.0e-3	8.5e-3	0.99	0.99	0.29
h=2	0.21	0.03	0.06	1.00	6.6e-3	6.6e-6	2.7e-3	0.99	0.93	0.53
h=3	0.07	0.09	0.18	1.000	0.03	2.6e-7	0.01	0.99	0.87	0.76

Step 3: **Evaluate ARMA(1,0)-GARCH(1,1) performance**. Based on the results from above steps, we finally choose ARMA(1,0)-GARCH(1,1) as the optimal model. The performance of this model on test data set is showed in the following table. It has MSE(h=1)=0.004546, MSE(h=2)=0.004548 and MSE(h=3)=0.004537.

$$Mean\ equation\ ARMA(1,0) : r_t = \alpha + \phi r_{t-1} + w_t$$

$$w_t = \sigma_t \epsilon_t$$

$$Variance\ equation\ GARCH(1,1) : \sigma_t^2 = \alpha_0 + \alpha_1 r_{t-1}^2 + \beta_1 \sigma_{t-1}^2$$

Table: Evaluation of the ARMA(1,0)-GARCH(1,1)

h	h=1	h=2	h=3
MSE	0.004482	0.004607	0.004596

3.3 Neural Network (NN) Model

Apart from ARMA-GARCH model to forecast the stock market, we also try machine learning method. We know that Neural Network are quite flexible model. It can capture the non-linear relationship between variables and usually has an accurate result.

3.3.1 Introduction of NN

A complete NN model contains the input layer, hidden layer and output layer. When designing a neural network, the number of nodes in the input layer and the output layer is usually fixed, while the middle layer can be freely specified. It is remarkable that this model can "think" like human brains and "learn" by itself. Learning rate controls the update step size in each iteration of the algorithm. If it is too large, the accuracy is very low. If it is too small, it will take longer time to complete iterations. The NN model can learn things through two process.

Firstly, "go forward". Every epoch, inputs x_i go through the model from left to right, they will be multiplied by different weights. Then, sum up all the inputs multiplied by weights and get the expressions.

$$a = \sum_i w_i x_i = w^T x$$

After that, the output goes into activation function to get the output of the neuron, where f can by any suitable functions such as sigmoid function, ReLU, logistic function and so on.

$$y = f(a)$$

Then, the calculation moves forward to the next and the next layer until the last layer (output layer).

Secondly, "go backward". Compared with real y, there exist error between fitted y and real y. The model will get the feedback of the error and will apply the designated method to reduce the error. Next time, it will adjust its weight values and "go forward". In this model, we adopt Back Propagation method which is based on gradient descent to reduce the error effectively. The formula is as follows, where w_c is the current weights and w_+ is the updated weights.

$$w_+ = w_c - \eta \Delta H(w_c)^{-1} E(w_c)$$

Through hundreds of "go forward" and "go backward", the weight values are adjusted over and over again so that the error will be smaller and smaller. Therefore, the parameters of this model are trained.

3.3.2 NN Model Application

After analysis of ARMA-GARCH model, we know that the model is supposed to be parsimonious, which means there is no need for h in x_{t-h} to be large. Thus, we use previous three weeks (Model 1), two weeks (Model 2), and only one week (Model 3) data to predict the following weeks log return and choose the most suitable one.

Model 1 takes x_t, x_{t-1}, x_{t-2} as inputs.

Model 2 takes x_t, x_{t-1} as inputs.

Model 3 takes x_t as inputs.

The following process show how we apply model 1 on stock 1 to have 1-step forecast . x_t is treated as the output. Similar steps will be repeated to have 2-step and 3-step forecast.

Step 1: **Rearrange the data structure**. The structure of the data is showed in the following figure. Ri represents log return in week i for stock 1. Data in x_t, x_{t-1}, x_{t-2} serves as inputs and x_{t+1} serves as output.

Step 2: **Train the NN model based on the training data**. we set training epoachs as 1000 and learning rate 0.1 for each training. After several adjustment, we choose 10 hidden layers. In R the coding is nnet(V4 ~., data, maxit=1000, size=10, decay=0.1, linout=T). Notice that "linout=T" means the results will be switched for linear output units, otherwise, the default setting is in logistic units which only gives the positive number.

Step 3: **One-step forecast**. After inputting log returns in week 206, 207 and 208, we can predict the log return in Week 209.

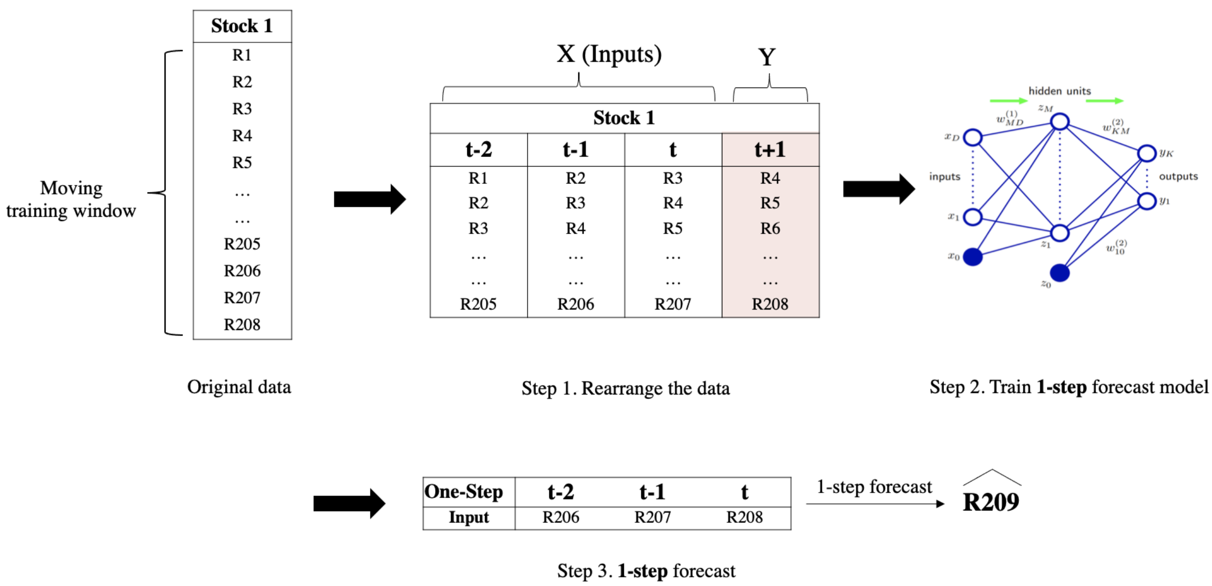


Figure: Process of Applying NN model in 1-step Forecast

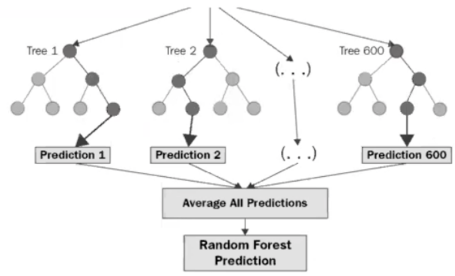
Apart from these, we also use previous two weeks (Model 2) and previous only one week (Model 3) log return as inputs to do forecasting. Still set training epoachs as 1000, learning rate 0.1 and hidden layers 10. For each model, we calculate train MSE for step 1, step 2 and step 3 forecasting. Finally, model 1 has the smallest MSE when h=2,3. Thus, in terms of NN model, we choose model 1 as the optimal NN model to predict future log return.

Table: Performance of NN model

Model (Inputs)	MSE(h)=1	MSE(h)=2	MSE(h)=3
Model 1 (x_t, x_{t-1}, x_{t-2})	0.004598	0.004621	0.004611
Model 2 (x_t, x_{t-1})	0.004573	0.004633	0.004612
Model 3 (x_t)	0.004568	0.004629	0.004619

3.4 Random Forest

NN model does not improve the results of this time series data. We suspect that in the small data set, the NN model may easily overfit the data. In order to tackle this problem, we fit random forest model. This algorithm is applied in various industries such as banking and stock market to predict behavior and outcomes. Its outcome is based on the predictions of the decision tree and take average of the results from the trees. Therefore, we also try this model in our series data.



Step 1: **Rearrange the data structure**. This step is the same as NN model mentioned in 3.3.

Step 2: **Train the random forest model based on the training data**. After several adjustment, we set ntree=200.

Step 3: **Do forecasting**. We take the value of x_{t-1}, x_t as inputs to predict the future values.

The performance of the model is presented in the table.

Table: Performance of Random Forest model

MSE(h)	h=1	h=2	h=3
Random Forest	0.004866	0.005530	0.006039

3.5 Best Model Selection

We firstly try ARMA(1,0) model. Although this model has the smallest MSE(h) when h=2 and 3 among all the models, the residual of this model does not follow normality iid condition. Due to this reason, we introduce GARCH(1,1) to capture the potential relation. After that, we apply several machine learning methods that are commonly used in financial stock market. However, these methods do not improve the results. From the results below, ARMA(1,0)-GARCH(1,1) has the smallest MSE when h=1,2 and 3. We choose this model as the optimal model for this data set. The detailed forecasing performance on Week 209 to Week 260 is showed in Appendix B.

Table: Comparison of different models

MSE(h)	h=1	h=2	h=3
ARMA(1,0)	0.004546	0.004548	0.004537
ARMA(1,0)-GARCH(1,1)	0.004482	0.004607	0.004596
NN Model	0.004598	0.004621	0.004611
Random Forest	0.004866	0.005530	0.006039

4 Portfolio Construction

Step 1: We use the optimal one-step model from 3.2 to forecast the following one week log return of each stock. From Week 209, every week, the information of the previous 208 days is used to estimate the coefficients of ARMA(1,0)-GARCH(1,1) model. In other words, the estimated parameters of the model are rolling and changing as time goes by. At every week t, we use the model to forecast the log returns in week t+1.

Step 2: Based on the predicted results, we will construct a portfolio by longing and shorting or only longing stocks and compute real log return of the portfolio. The log return of the portfolio for Week t+1 is

$$r_{t+1} = w_{t,1}x_{t+1,1} + ... + w_{t,10}x_{t+1,10}$$

Then, compare the log return with SP500 index. After that, we will consider excessive return

$$e_s = r_s - r_{0,s}$$

where $r_{0,s}$ is the log return of SP500 in Week s. Our goal is to maximize the ratio

$$\frac{\hat{\mu}_e}{\hat{\sigma}_e}$$

Here, we design four strategies.

- Strategy 1: We will firstly use ARMA(1,0)-GARCH(1,1) to forecast the log returns of each stock in Week t+1. According to the predicted log return in Week t+1, at each Week t, we will order the stocks, long the stocks with positive forecasted log returns, short stocks with negative forecasted log returns. Then, we use true log returns from Week 209 to Week 260 to calculate returns that are generated from the portfolio. Compare the return with SP500 index. The weight of each stock should follow the rule that if stocks have positive predicted log returns on Week t+1, each of them will be given 0.1 weight. If stocks have negative predicted log returns on Week t+1, each of them will have remaining equal weight.

$$w_{t,i} = 0.1,\ if\ stock\ i\ has\ positive\ forecasted\ log\ returns\ in\ Week\ t+1$$
$$w_{t,i} = \frac{1 - 0.1 \times number\ of\ stocks\ with\ positive\ forecasted\ log\ returns}{number\ of\ stocks\ with\ negative\ forecasted\ log\ returns},\ otherwise \tag{1}$$

- Strategy 2: We still use ARMA(1,0)-GARCH(1,1) to forecast the log returns of each stock in Week t+1. Order the stocks according to the predicted log return in Week t+1. This time we will only long 5 stocks with the highest forecasted log returns. Then, we calculate log returns of the portfolio. Compare the return with SP500 index. The weight allocation of this strategy is 0.2 for each stock. Each week, we will refresh the components of the portfolio.
- Strategy 3: Use ARMA(1,0)-GARCH(1,1) to forecast the log returns of each stock in Week t+1. Order the stocks according to the predicted log return in Week t+1. We long the stocks with positive forecasted log returns and calculate log returns of the portfolio. Compare the return with SP500 index. The equal weight will be allocated to the selected stocks. Each week, we will refresh the components of the portfolio.
- Strategy 4: The goal is to maximize the ratio of $\frac{\hat{\mu}_e}{\hat{\sigma}_e}$. It means we need a strategy to construct a portfolio that can generate higher return than SP500. At the same time, the return of the portfolio is expected to have low standard deviation. Thus, we connect this problem with sharpe ratio problem to maximize the ratio by choosing optimal weight. Based on Markowitz's Modern Portfolio Theory (MPT) and Maximum Sharpe ratio portfolio (MSRP), we need to optimize the following equations to get the optimal weight, where $\hat{r}_{ti}$ is 1-step forecast value of stock. $\bar{r}_m$ is the average value of SP500's log return in the moving training window. Due Σ represents the covariance of each stock based on training data. Here, the stock is not allowed to be shorted. After getting the weight, we compute $\frac{\hat{\mu}_e}{\hat{\sigma}_e}$, which is 0.08567.

$$\begin{aligned} &\max_{w} \frac{w^T \hat{r}_{ti} - \bar{r}_m}{\sqrt{w^T \Sigma w}} \\ &subject\ to\ 1^T w = 1 \\ &w \geq 0 \end{aligned} \tag{2}$$

Table: Comparison of 4 strategies

Target Ratio	Strategy 1	Strategy 2	Strategy 3	Strategy 4
Total return	1.1325	0.34028	0.3256	0.08566
Var(return)	6.99e-3	5.5e-3	4.16e-4	4.21e-3
Ratio	0.2605	0.2771	0.3072	0.08028

For strategy 1, $\frac{\hat{\mu}_e}{\hat{\sigma}_e} = 0.2605$. For strategy 2, $\frac{\hat{\mu}_e}{\hat{\sigma}_e} = 0.2771$. For strategy 3, $\frac{\hat{\mu}_e}{\hat{\sigma}_e} = 0.3072$. For strategy 4, $\frac{\hat{\mu}_e}{\hat{\sigma}_e} = 0.08028$.

In terms of ratio, strategy 3 has the best performance as it has the lowest variance among the three strategies. However, its total return is the lowest. When the economy is bad, the stock market is more unstable and fluctuating, which is even harder to predict. Only depending on the model to long stocks with positive forecasted returns will bring more risks. It is undeniable that this strategy may perform pretty good when the economy is booming.

As for strategy 2, although in this data set, it has higher ratio than strategy 1, this strategy is sensitive and is fairly depends on the whole economic conditions. For example, if we are in bull market, it is better to adopt strategy 2. At this period, lots of stocks present optimistic growing trend. Long 5 stocks with the highest forecasted log returns may bring higher return. However, when it is in bear market, the economy is in a gloomy. The stocks may fluctuate a lot they are more likely to present negative returns. At this time it is not sagacious to only long the stocks. Instead, strategy 1 at this time may bring more benefits.

As for strategy 4, it seems rational to allocate the weight through optimization problem. However, through this strategy, whether the result is good or not depends on the performance of the model. If the model can accurately predict the future returns, this strategy can generate the largest profit. Nevertheless, the stock market fairly hard to forecast. Thus, this method fails.

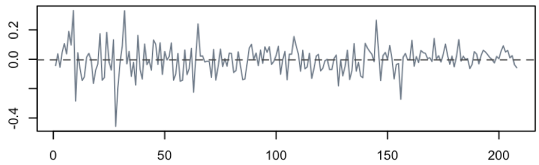
To sum up, in this data set, we aims to maximize the ratio. Strategy 3 performs the best. If we hope the strategy that can have a stable and relative high ratio no matter the economy is good or in a gloomy, then strategy 1 will be a more suitable choice.

5 Conclusion

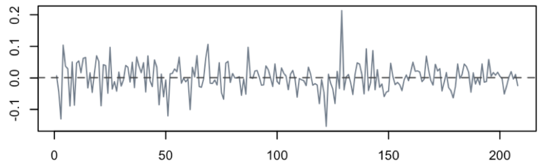
ARMA(1,0) model can generally capture the level trend of stocks. Considering the phenomenon of volatility clusters, ARMA(1,0)-GARCH(1,1) model is eventually chosen as the optimal model. We use this model to forecast log returns of each stocks from Week 208. Four strategies are given to construct a portfolio with high $\frac{\hat{\mu}_e}{\hat{\sigma}_e}$ ratio. Strategy 1, namely longing the stocks with positive forecasted log returns, shorting stocks with negative forecasted log returns will be a wise and stable choice no matter the economy is in a boom or gloomy.

6 Appendix

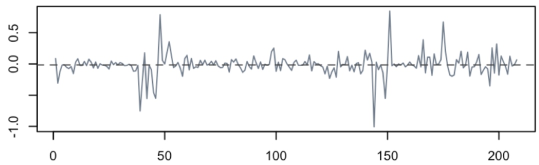
6.1 A. Time series plots of each stock



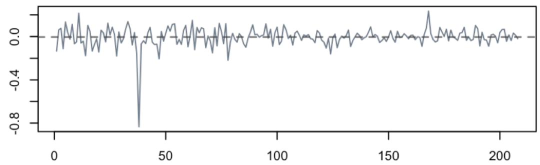
Week return of stock 1



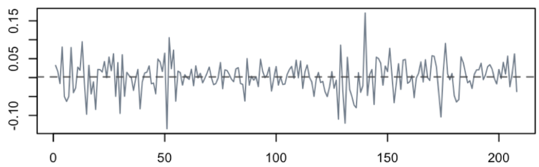
Week return of stock 2



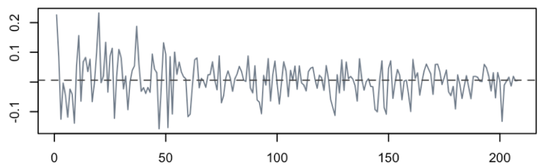
Week return of stock 3



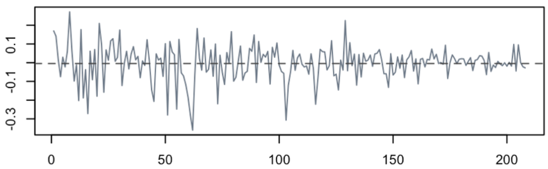
Week return of stock 4



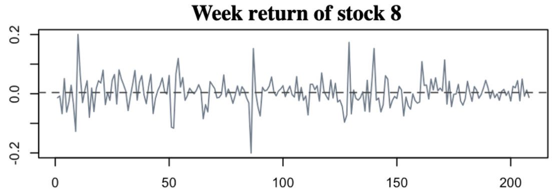
Week return of stock 5



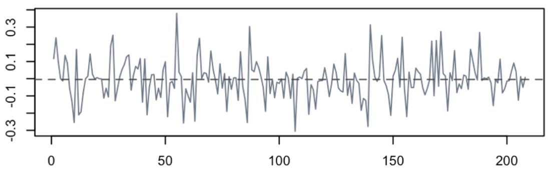
Week return of stock 6



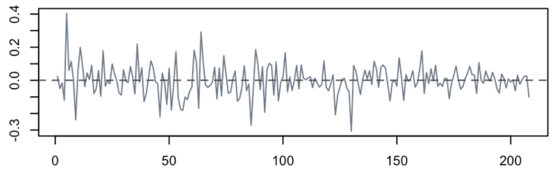
Week return of stock 7



Week return of stock 8



Week return of stock 9



Week return of stock 10

6.2 B. Model forecasting from Week 209 to Week 260