EBA 5005 Practice Module in Specialized Predictive modelling & Forecasting

Stock Investment Customization using Predictive and Optimization Methods

Team Name

Group 2 Arcadia

Team Members

MA CHENGBIN A0261804J

XIE SITENG A0261982W

ZHOU JIECHENG A0261829W

CHEN LIUJUN A0261904H

LIN FANGZHOU A0261850H

Table of Contents

List of Figures

List of Tables

1. Introduction

Arcadia is a regional leading asset management firm in Southeast Asia. Our team is working on asset management about the stock investment. Facing clients with different investment preferences, our main task is to make the most suitable portfolio which can maximize returns and minimize risks as well as make accurate predictions on the stock price movement to identify potential investment opportunities.

2. Industry Overview

According to a report by Grand View Research, the global asset management market size was valued at USD 91.4 billion in 2022 and is expected to grow at a compound annual growth rate (CAGR) of 13.8% from 2022 to 2028. The report also predicts that the increasing demand for sophistication. The total market capitalization of all publicly traded securities worldwide rose from US$2.5 trillion in 1980 to US$93.7 trillion at the end of 2022.

The development of the stock market played an important role in promoting economic development. Therefore, accurate forecasting of stock prices has always been a hot topic for researchers. However, the stock prices often fluctuate greatly, showing complex nonlinear and randomness. There are many factors that affect the stock price, which makes it difficult to predict the stock price. Lacking experience and knowledge, many net worth users find it difficult to achieve consistent returns.

3. Business Problem & Objectives

3.1. Business Problem

1. Making informed investment decisions by accurate prediction on stock prices and Buy-and-sell-point based on rational hypotheses.
2. Provide customized asset allocation optimization solutions for clients with different investment preferences. By tailoring our services to each client's unique needs and goals, we can help them achieve long-term financial success.

3.2. Technical Objectives

1. Use a range of different algorithms and models (LSTM, Survival Analysis, ARIMA–GARCH, etc.) to predict stock prices. We will form an optimal strategy for each industry we research. This predictive method will also enable us to know the point to buy and the point to sell.
2. Formulate customer personas. Utilize a range of optimization algorithms and portfolio analysis tools (goal programming etc.) to model different asset allocation

strategies and optimize portfolios for risk tolerance, investment horizon, and return objectives based on clients' requirements.

4. Scope of work

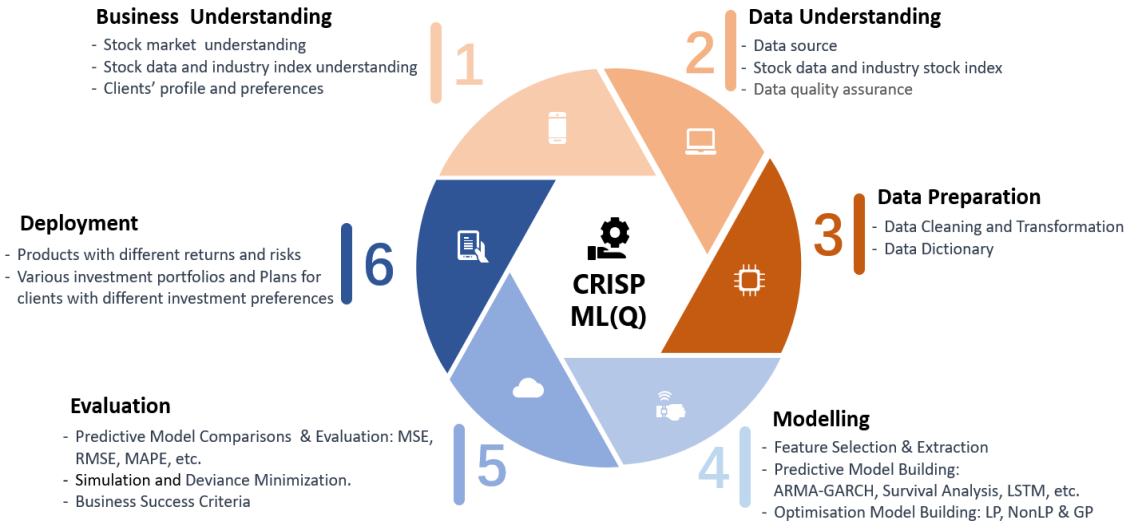

Figure 1. Project Scope

5. Effort Estimates and Timeline

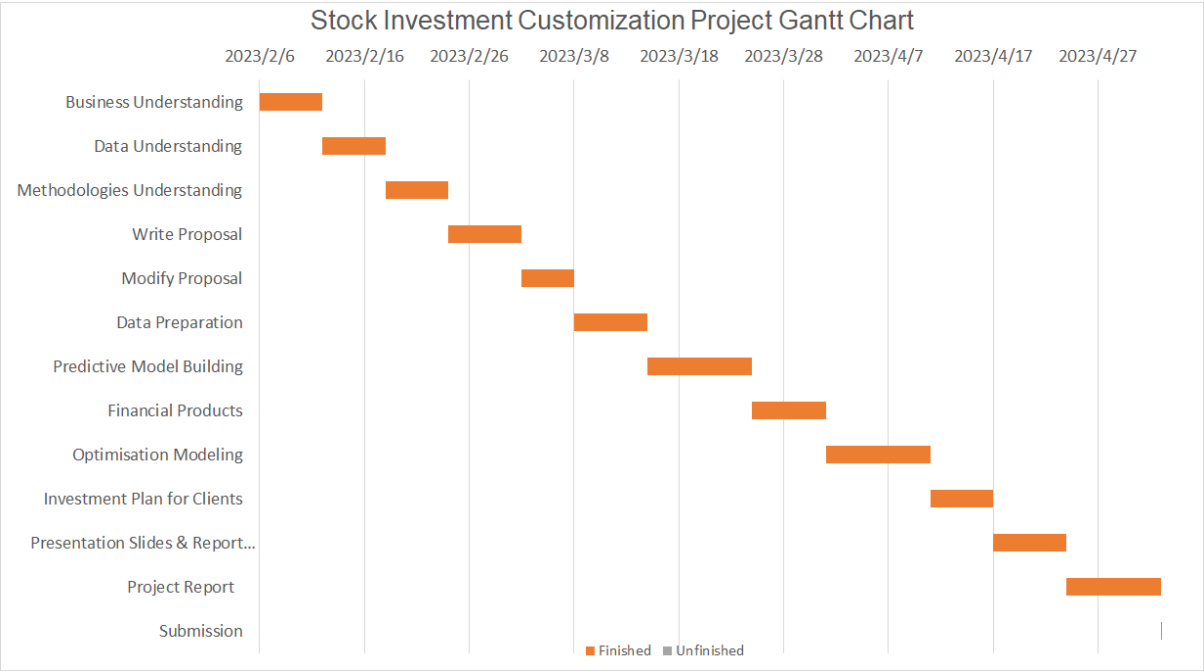

Figure 2. Project Gantt Chart

6. Dataset Used & Data Description

6.1. Source of Data, Type of Data

For Stock Price Prediction:

The source of data is from Wind (extract directly from the software) and Yahoo (through API). Our team collected stock related data in different industries including banking, medical, semiconductor, etc. The datasets contain three parts information

1. Historical trading data: (i) closing price (ii) KDJ (iii) turnover rate
2. Companies' fundamental data: (i) stock code (ii) ROE (iii) Debt Assets Ratio
3. Macroeconomic data: (i) GDP (ii) interest rate

For investment Portfolio Optimization:

1. Client profile and investment preference data: (i) investment amount (ii) risk tolerance (iii) return objectives (iv) special requirements
2. Internal constraints:
   - for each specific investment project: (i) there are minimum and maximum limits on investment (ii) annual expense ratio.
   - for some investment project: holding time requirements, for example, at least six months.

7. Data Understanding & Data Processing

The description of data set is shown in figure3.

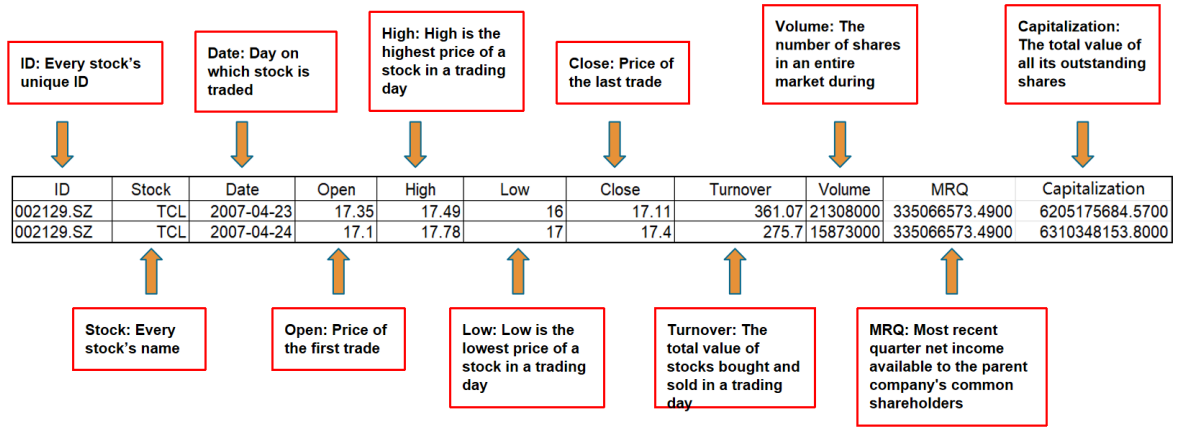

Figure 3. Data Sample

8. Predictive Model

8.1. ARMA GARCH

8.1.1 Introduction of ARMA GARCH

ARMA-GARCH models are a class of time series models that combine an autoregressive moving average (ARMA) model with a generalized autoregressive conditional heteroskedasticity (GARCH) model. The ARMA part of the model captures the conditional mean of the time series, while the GARCH part captures the conditional variance. The GARCH model is used to adjust the errors of the ARMA model, resulting in a more accurate overall model.

8.1.2 Procedure of fitting ARMA GARCH model

The model building follows the procedures in Figure 4. First, an ADF test should be conducted to determine stationarity. When comparing different specifications of ARMA-GARCH models, notice the significance of parameters, used information criteria based on AIC to avoid choosing very complex models. Also, comparing the other statistical results. After choosing the appropriate model, simulate the future return and calculate RMSE.

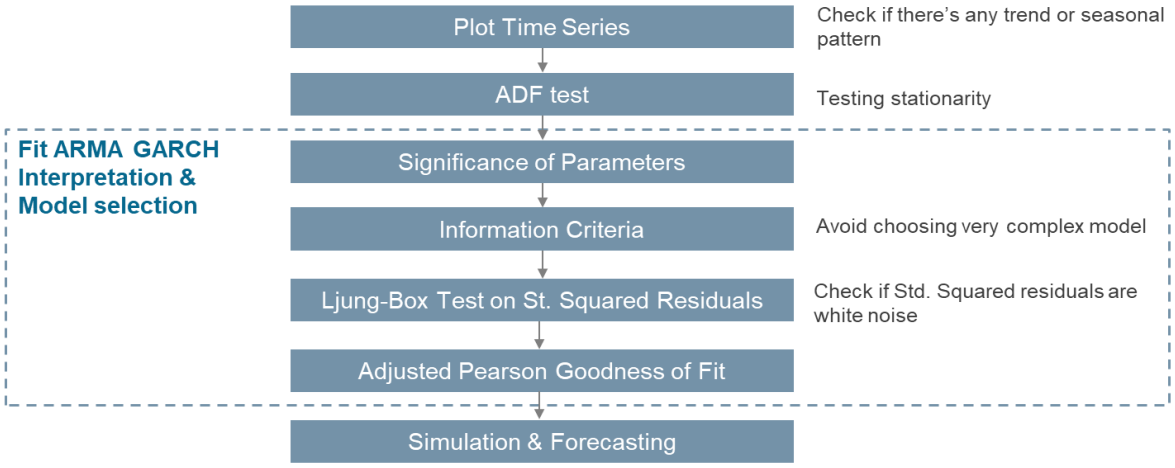

Figure 4. procedure of conducting ARMA-GARCH model

8.1.3 Modelling Results: Case study

Here we take Evemall as an example. After model interpretation and selection, we finally chose the gjrGARCH model with skewed student t-distribution for this stock. Table 1 shows that the estimates of parameters are statistically significant.

Table 1. Statistical Estimates of GARCH

	Estimate	Std.Error	t value	Pr(>|t|)
mu	0.002516	0.000091	27.6877	0
omega	0.000002	0.000001	2.0349	0.041856
alpha1	0.009496	0.00057	16.6551	0
beta1	0.999992	0.00034	2941.909	0
gamma1	-0.02412	0.001431	-16.8556	0
skew	1.124542	0.03805	29.5541	0
shape	5.026918	0.715242	7.0283	0

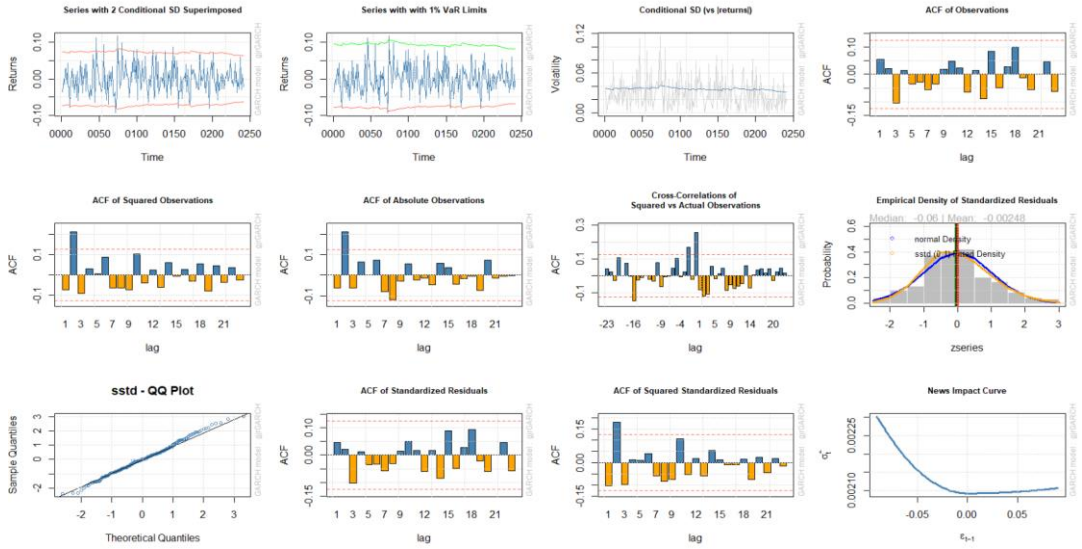

Figure 5. Out of RStudio for ARMA modelling

The result visualization includes the actual stock price curve, and the predicted curve, as well as RMSE.

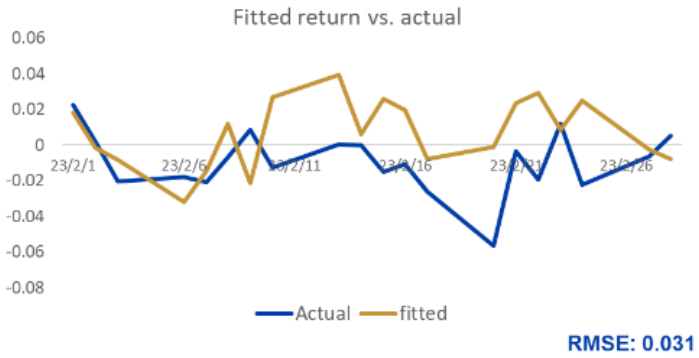

Figure 6. Prediction of GARCH model

8.2. LSTM

8.2.1 Introduction of LSTM

The stock market is notoriously unpredictable, making it challenging to predict stock prices accurately. However, with the rise of advanced machine learning techniques,

such as Long Short-Term Memory (LSTM), it has become possible to forecast stock prices with a high level of accuracy.
LSTM is a type of recurrent neural network that can remember and store past information to make better predictions. It has proven to be highly effective in solving sequence prediction problems, such as time series forecasting.

8.2.2 Processing Flow

(1) Data import

The data processing flow of LSTM involves several steps. Firstly, the input data is collected using the closing price of the stock.

code	name	date	open	highest	lowest	close	turnover	volume
002129.SZ	TCL中环	2022-01-04 10:00	177.86	178.49	173.25	173.33	524.55	12676262
002129.SZ	TCL中环	2022-01-04 10:30	173.37	173.67	171.04	171.64	347.83	8554656

Figure 7. Close price as input data

(2) feature scaling

Secondly, the data is normalized to ensure that all input values are on the same scale, making it easier for the model to learn patterns in the data.

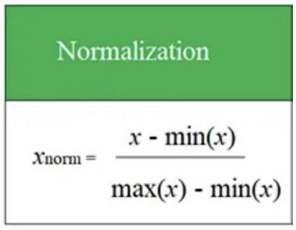

Figure 8. Data normalization

(3) Data structure creation

Then, we create the dataset by using the closing prices of the previous timestep days to predict the closing price of the next day.

Figure 9. Set x,y

We divide the dataset into training and test sets, with the total dataset covering the period from January 2022 to March 2023, and we aim to predict the prices from January to March 2023.

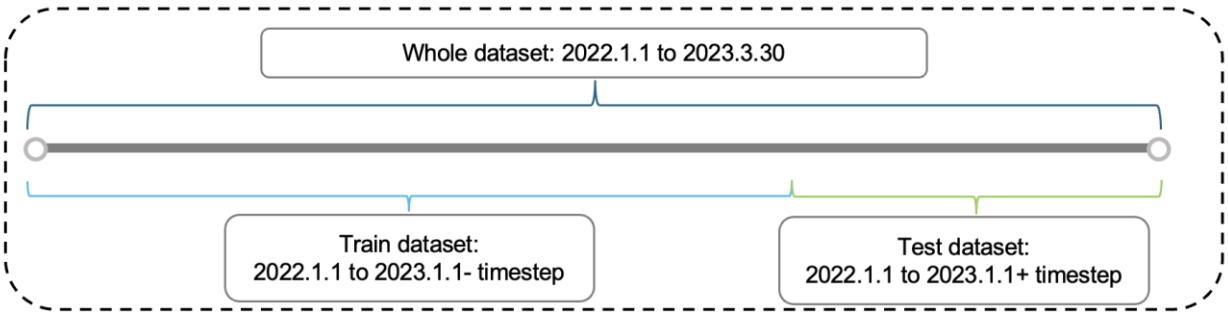

Figure 10. Dataset creation

(4) Modelling

The modeling process of LSTM involves hyperparameter tuning, where several parameters, such as timestep, number of layers, hidden dimension of each layer, batch size, and epoch, are optimized to improve the performance of the model. The impact of each parameter is individually explained to provide a better understanding of the modeling process.

Table 2. Pros and cons of hyperparameters

Hyperparameters	Pros	Cons
Time step ↑	remember historical information	computational complexity and training time.
Number of layers ↑	complexity and expressive power	the risk of overfitting and training time.
Hidden dimension of each layer ↑	complexity and expressive power	computational complexity and training time.
Batch size ↑	training speed and stability	consumes more memory resources and can lead to reduced generalization performance
Epochs ↑	fitting ability and performance	training time and the risk of overfitting.
Dropout ↑	prevent overfitting	increase training time and decrease model performance if the dropout rate is too high.

Here are the parameters we set

*Due to the limitation of time and computing power, we can't set too much values

batch_size = [64, 128]

time_steps = [30,40,60]

num_layers = [1, 2]

num_nodes = [32, 64]

epochs = [50, 100]

dropout = [0, 0.1]

(5) result and visualization

The final results of the LSTM model include the optimized model and the predicted prices of each stock from January to March 2023.

The result shows when the timestep is 30 days and epoch = 100 we will get the best models, but the structure and batch size varies for different stock.

Table 3. Network structure of each stock

Stock	Batch size	Num layers	Num nodes	Dropout
1	64	1	64	0
2	64	2	64	0.1
3	64	1	32	0
4	64	1	64	0
5	64	1	64	0
6	64	2	64	0
7	64	1	64	0.1
8	128	1	64	0
9	64	1	64	0
10	64	2	64	0.1
11	64	1	64	0
12	128	1	32	0.1

13	64	1	32	0
14	64	1	64	0.1
15	64	1	64	0
16	64	2	64	0
17	64	2	64	0
18	64	2	64	0
19	64	1	64	0.1
20	64	1	32	0.1

The result visualization includes the loss curve, the actual stock price curve, and the predicted curve, as well as RMSE.

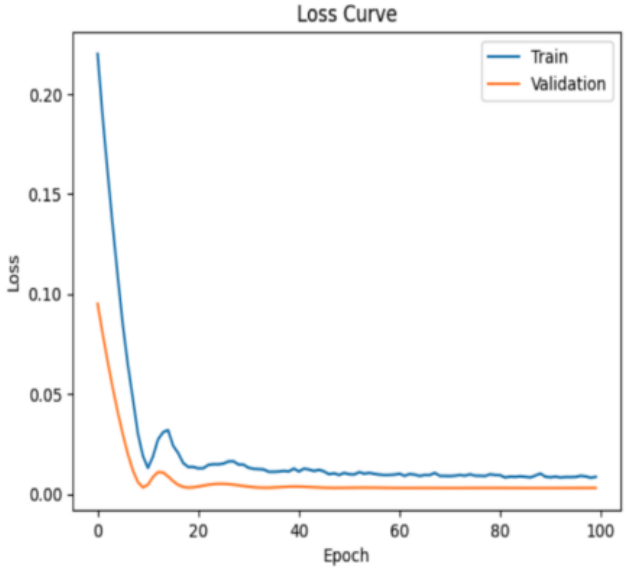

Figure 11. Loss curve example

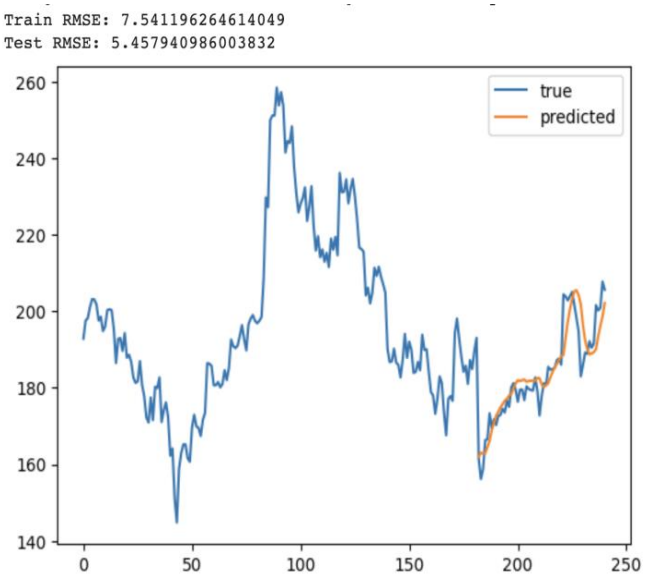

Figure 12. True and predicted curve with RMSE example

8.3. Survival Analysis

8.3.1. Context and Assumptions

From view of survival analysis, a stock with at least α one-day rise (denoted as R_α) is the rise event and a stock with at least β one-day drop (denoted as D_β) is the drop event. There are four statues could be used to describe the fluctuations of a stock:

R_α state: if a stock has at least α one-day rise;

$\check{R}_\alpha$ state: if a stock has no more than α one-day rise;

D_β state: if a stock has at least β one-day drop;

$\check{D}_\beta$ state: if a stock has no more than β one-day drop.

There are some core questions we seek to answer after doing the survival analysis:

(1) What is the proportion of a stock which will stay in states R_α or D_β past a certain time?

(2) At what rate will the stock price rise or fall?

Answering these questions will help us to find the best **buy-and-sell-point** by predicting return increase and decrease probability, that is also our goal to do survival analysis.

We first have some assumptions that explain how to define a death or living event for survival analysis in this case.

α represents return increase rate

β represents the return decrease rate

γ_t Gama t represents stock return at time t.

μ_t represents the turnover rate at time t.

There are two situations, and for each situation we will build a corresponding cox regression model to predict the probability.

For return increase situation, when α > 0.01, event is triggered and the event status is 1, else is 0. The duration is denoted as T_α which indicates each rise event's living time. For return decrease situation: when β < -0.01, event is triggered and the event status is 1; else is 0. The duration variable is denoted as T_β which indicates each drop event's living time.

8.3.2. Covariates Construction

We have chosen a set of technical indicators as covariates for our modeling efforts, aiming to predict future stock price movements based on historical visitation behavior. For instance, we have observed that a stock can become more susceptible to a sell-off after a prolonged market movement. To capture this insight, we have created covariates that track stock activities from the last significant event until the current observation time. Such covariates seek to predict the future stock price movement based on how their visitation behavior has been historically.

The covariates are shown below.

(1) Accumulated Rate of Change (AcROC): The cumulative gains from the time point when the latest "event" happened to the current time point.

$$x_1(T)=\sum_{t=T_c-T}^{T_c-1}\gamma_t$$

(2) Average Rate of Change (AvROC): The average gains from the time point when the latest "event" happened to the current time point.

$$x_2(T)=\frac{\sum_{t=T_c-T}^{T_c-1}\gamma_t}{T}$$

(3) Accumulated Turnover (AcT): The cumulative turnover rate from the time point when the latest "event" happened to the current time point.

$$x_3(T)=\sum_{t=T_c-T}^{T_c-1}\mu_t$$

(4) Average Turnover (AvT): The average turnover rate from the time point when the latest "event" happened to the current time point.

$$x_4(T)=\frac{\sum_{t=T_c-T}^{T_c-1}\mu_t}{T}$$

(5) Stochastic K% (K%): This covariate refers to the point of a current price in relation to its price range over a period; HHT and LLT mean lowest low and highest high in the last T days, respectively.

$$x_5(T)=\frac{P_{T_{c-1}}-LL_T}{HH_T-LL_T}*100\%$$

(6) Stochastic D% (D%): This covariate measures the average K% over the last n days.

$$x_6(T)=\frac{\sum_{t=T_c-T}^{T_c-1}*K_t\%}{T}$$

(7) Stochastic J% (J%): This covariate is a derived form of the stochastic with the only difference being an extra line.

$$x_7(T)=3K_{T_{c-1}}\%-2D_{c-1}\%$$

(8) Relative strength index (RSI): This covariate is intended to chart the current and historical strength or weakness of a stock on the closing prices of a recent trading period.

$$x_8(T)=100-\frac{100}{1+RS},\ RS=\frac{\sum_{t=T_c-T,\ \gamma_t\geq 0}^{T_c-1}\gamma_t}{\sum_{t=T_c-T}^{T_c-1}\gamma_t}$$

(9) Psychological Line (PSY): This covariate measures the ratio of the number of rising periods over the total number of periods.

$$x_9(T)=\frac{\sum_{t=T_c-T}^{T_c-1}I(\gamma_t)}{T}$$

Where $I(\gamma_t)$=1 if $\gamma_t \geq 0\ and\ \ 0\ otherwise$

8.3.3. Model Design

By setting the rise event threshold to be 1% and drop event threshold -1%. There are two models, one is the Rise Model, the other is the Drop Model.

(1) The equation for rise model:

$$h_{ri}(t) = e^{-(\beta_1 * x_{1i} + \beta_{2i} * x_{2i} + \beta_{3i} * x_{3i} + \cdots)} h_{r0}$$

$\mathrm{x_{1i}}, \mathrm{x_{2i}}, \ldots, \mathrm{x_{pi}}$ is $\mathbf{x}_\mathrm{m}$ which are the covariates for the rise model.

$h_{rj}(t)$ is the proportional hazard function for the rise event. The $h_{r0}(t)$ is the rise hazard function for an individual for whom the values of $\mathbf{x}_\mathrm{m}$ are zero. The function $h_{r0}(t)$ is called the baseline hazard function for the rise model.

(2) The equation for drop model:

$$h_{dj}(t) = e^{-(\beta_1 * x_{1j} + \beta_{2j} * x_{2j} + \beta_{3j} * x_{3j} + \cdots)} h_{d0}$$

$\mathrm{x_{1j}}, \mathrm{x_{2j}}, \ldots, \mathrm{x_{pj}}$ is $\mathbf{x}_\mathrm{n}$ which are the covariates for the drop model.

$h_{dj}(t)$ is the proportional hazard function for the drop event. $h_{\mathrm{d0}}(t)$ is the drop hazard function for an individual for whom the values of $\mathbf{x}_\mathrm{n}$ are zero. The function $h_{\mathrm{d0}}(t)$ is called the baseline hazard function for the drop model.

Note: (a)These variables can be continuous, categorical, binary, etc. (b)The values of these variables will be assumed to have been recorded at the time origin of the study. (3) Each stock will have corresponding fitted rise model and drop model.

8.3.4. Modeling Process

During the process, we use Wald Test to select significant variables for cox regression. We also conduct parameter selection for parameter penalizer. We use parameter grid to choose optimal parameter with highest **c-index.**

Parameter grid is [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2]

C-index: refers to the accordance and accuracy of the model. It ranges from 0-1, 0.5 means random prediction, above 0.7 means good prediction.

8.3.5. Modeling Results: Case Analysis

We made cox regression for each stock.

Here we take CGNPC (China Guangdong Nuclear Power) as an example, there are two tables showing variable significance for the rise model and drop model.

In the rise model, average turnover, stochastic K%, stochastic J% and relative strength index is statistically significant.

Table 4. Variable Significance for the Rise Model

covariate	coef	exp(coef)	se(coef)	p	-log2(p)
AcROC	-0.0293	0.9711	0.2783	0.9160	0.1265
AvROC	0.4397	1.5523	0.6599	0.5052	0.9805
AcT	0.0000	1.0000	0.0000	0.7626	0.3910
AvT	0.0016	1.0016	0.0005	0.0006	10.5996
K%	0.0102	1.0102	0.0027	0.0002	12.2754
D%	0.0015	1.0015	0.0037	0.6770	0.5628
J%	0.0038	1.0038	0.0009	0.0001	14.0510
RSI	0.0246	1.0249	0.0082	0.0027	8.5468
PSY	0.5493	1.7320	1.9480	0.7780	0.3622

In the drop model, average turnover, stochastic J% and psychological line is statistically significant.

Table 5. Variable Significance for the Drop Model

covariate	coef	exp(coef)	se(coef)	p	-log2(p)
AcROC	-0.1050	0.9004	0.2055	0.6095	0.7143
AvROC	-0.2466	0.7814	0.4888	0.6139	0.7040
AcT	0.0000	1.0000	0.0000	0.9637	0.0534
AvT	0.0017	1.0017	0.0004	0.0000	17.3876
K%	-0.0028	0.9972	0.0021	0.1837	2.4443
D%	0.0049	1.0049	0.0030	0.0970	3.3666
J%	-0.0017	0.9983	0.0007	0.0227	5.4588
RSI	0.0061	1.0061	0.0067	0.3608	1.4709
PSY	-4.3025	0.0135	1.5320	0.0050	7.6498

There are also four figures showing the baseline cumulative hazard function and survival function for both the rise model and drop model, respectively.

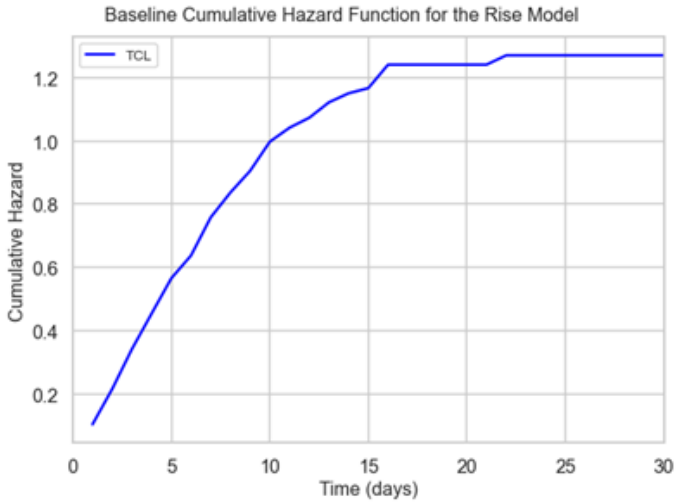

Figure 13. Baseline Cumulative Hazard Function for the Rise Model

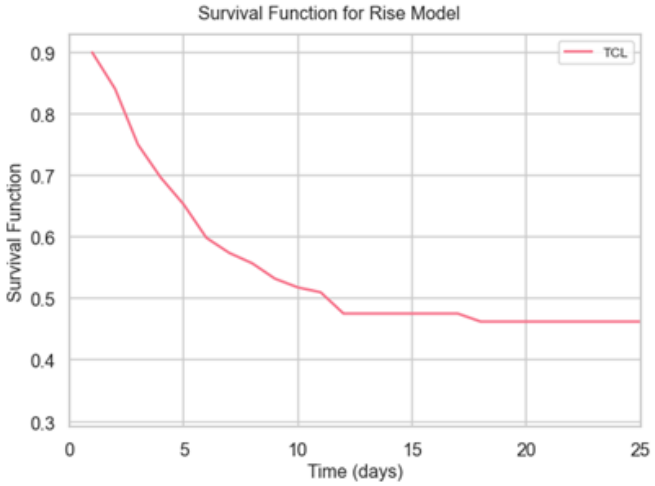

Figure 14. Survival Function for the Rise Model

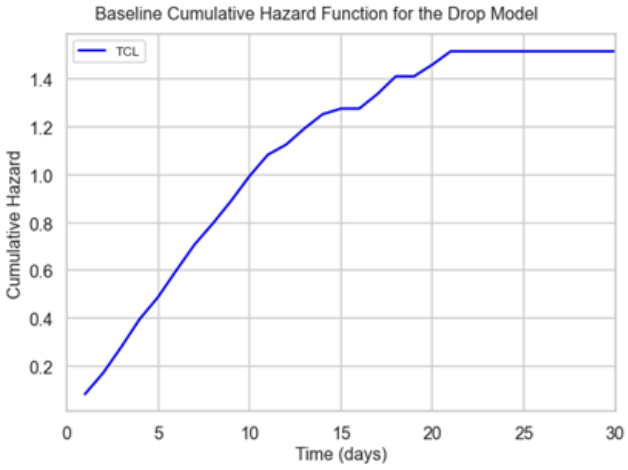

Figure 15. Baseline Cumulative Hazard Function for the Drop Model

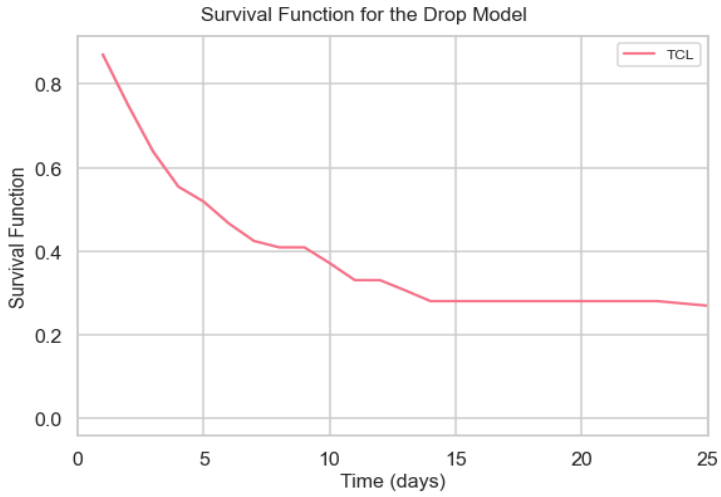

Figure 16. Survival Function for the Drop Model

We can notice that the survival functions have declining shape. Hence, the longer a stock stays in non-rise or non-drop state the more likely a rise or drop event will happen in the future. For example, in the rise model, the survival function has a value about 0.48 at 15 days, which in return shows that the risk to trigger the rise event is about 0.52.

8.4. Multiple Factor Regression

8.4.1. Factors in the Model

This section introduces the three factors used in the Fama-French 3 Factor Model, explaining their role in capturing different aspects of stock returns.

(1) SMB (Small Minus Big)

SMB measures the difference in returns between small and large market capitalization stocks. It indicates that small-cap stocks tend to outperform large-cap stocks over time.

(2) HML (High Minus Low)

HML captures the difference in returns between high and low book-to-market value stocks, showcasing the tendency for value stocks to outperform growth stocks over time.

(3) Market Return minus Risk-Free Rate

This factor represents the excess return of the market portfolio over the risk-free rate. It captures the overall market risk, indicating that higher risk should yield higher returns.

8.4.2. Data Processing and Model Implementation

This section describes the steps involved in data cleaning, factor calculation, and model implementation.

(1) Data Cleaning

The data cleaning process involves removing rows with null values, formatting date columns, and removing file suffixes.

(2) Time Horizon Selection

The time horizon for the analysis is from January 2022 to March 2023.

8.4.3. Defining the Factors

This subsection explains how each factor is calculated.

(1) Calculating SMB

We sort the market capitalization of the stocks and select the top five as large stocks and the bottom five as small stocks. Then, we calculate their returns to arrive at the value of SMB.

(2) Calculating HML

We calculate the BM ratio and sort and filter stocks. Then, we calculate the difference in return, which is the value of HML.

(3) Calculating Market Return minus Risk-Free Rate

We calculate the Rm and make it minus Rf to determine this factor.

8.4.4. Model Fitting and Alpha Calculation

We input the three factors into the model and calculate Alphas for each stock and day from January 4th to March 20th, setting the rollback days to 60 days.

8.4.5. Benefits of 60-Day Rollback Period

This section discusses the advantages of using a 60-day rollback period in the model.

(1) Data Smoothing

A longer time window smooths the data, reducing the impact of short-term fluctuations and improving the model's stability.

(2) More Data Points

Longer rollback days allow the model to use more data points for fitting, improving the model's predictive power.

(3) Cyclical Considerations

The 60-day time window covers about three months, capturing quarterly cyclical changes in the market.

(4) Better Generalization

Longer rollback days help the model generalize better over different time periods, maintaining forecasting performance even when market conditions change.

8.4.6 Trading Logic

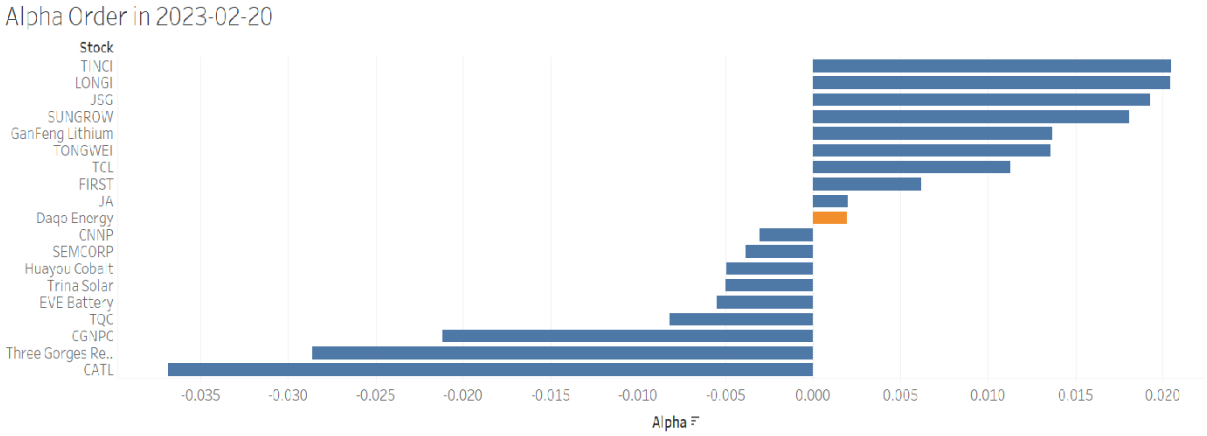

Figure 17. Trading Case1

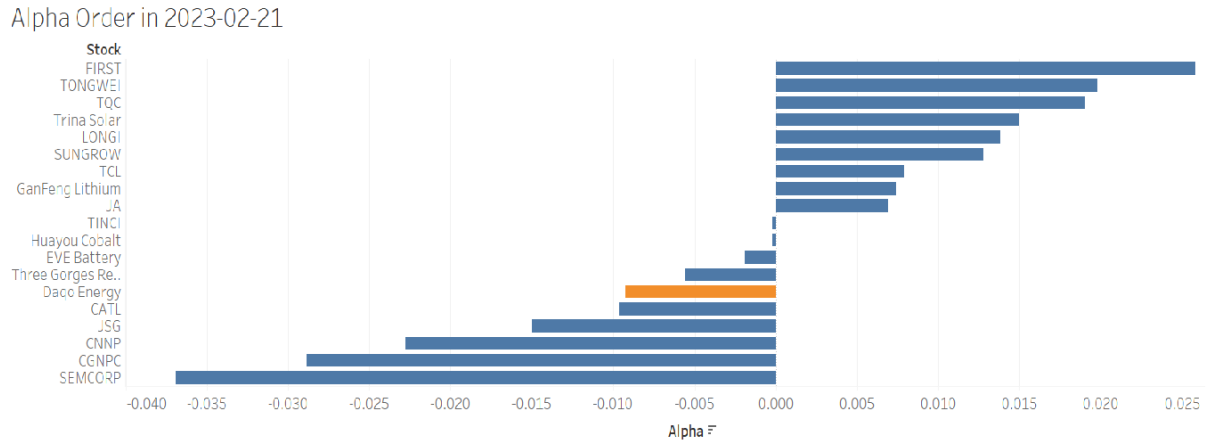

Figure 18. Trading Case2

We consider the stock to be undervalued when the alpha is below zero. So in the future it will return to its normal value, which we think should be bought at this time.

8.5 Model Comparison

We can compute the RMSE of predicted stock price for LSTM and Garch model and make comparison. The results can be seen in Figure 19.

It should be noted that the output of Survival model and Multiple Factor regression is different. Hence, we aren't able to calculate RMSE for these two types. However, we

can evaluate the trading strategy of different models by comparing risk and return in later discussion.

Stock	RMSE(LSTM)	RMSE(Garch)
TCL	5.02	16.4
DAQO	5.87	6.29
Semcorp	7.06	18.52
	.....	

Figure 19. comparison of LSTM and ARMA

9. Trading Strategy

9.1 Introduction

In this part, we evaluate the performance of different stock trading strategies by applying them to actual stock trading, using the predicted results from our forecasting models. The trading stocks and time period used for the evaluation were the same as the test period previously mentioned.

However, it should be noted that during this period, the weighted average stock price of the 20 stocks showed a continued downward trend, as seen in the candlestick chart provided in Figure 17. This posed a challenge for our predicted results and trading strategies, as short positions were not permitted in our trading.

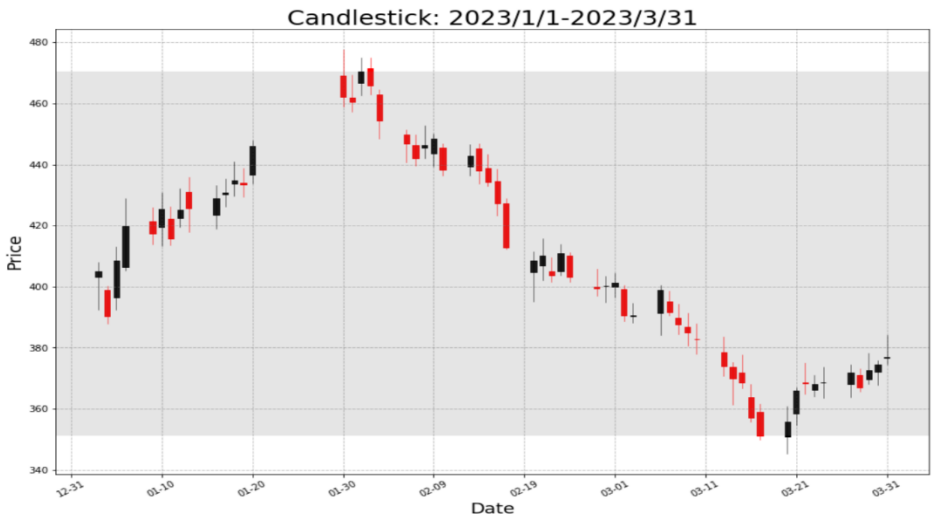

Figure 20. price trend of the 20 stocks

Despite this challenge, we present an analysis of our trading strategies and discuss the final investment products that we have developed for our clients. Through this report, we aim to provide insights into the effectiveness of various stock trading strategies and their potential applications in real-world investment scenarios.

9.2 Trading Strategy

In terms of our trading strategy, we traded only one stock from the 20 selected in each fitting, and every trade involved a full position buy or sell, based on the prediction results from Survival Functions, GARCH, and LSTM. For the multi-factor model, we expanded our trading portfolio to include all 20 stocks through periodic rebalancing.

(1) Survival Function

In the case of Survival Functions, we utilized the predicted daily probabilities of stock price increase and decrease to make buy or sell decisions. By regularly comparing the difference between the probabilities of the two events, we determined whether to buy or sell the stock.

$$\boldsymbol{Buy}\ stock\ with\ open\ price\ if\ P\alpha(T\alpha) - P\beta(T\beta) \geq \theta$$
$$\boldsymbol{Sell}\ stock\ with\ open\ price\ if\ P\beta(T\beta) - P\alpha(T\alpha) \geq \theta$$

Here $T\alpha$ refers to the number of days that event alpha hasn't happened and $P\alpha(T\alpha)$ is the predicted probability that event alpha will happen in the next $T\alpha$ days. The definition of $P\beta(T\beta)$ and $T\beta$ is also similar. Figure 18 shows the process we performed the trading during the test period.

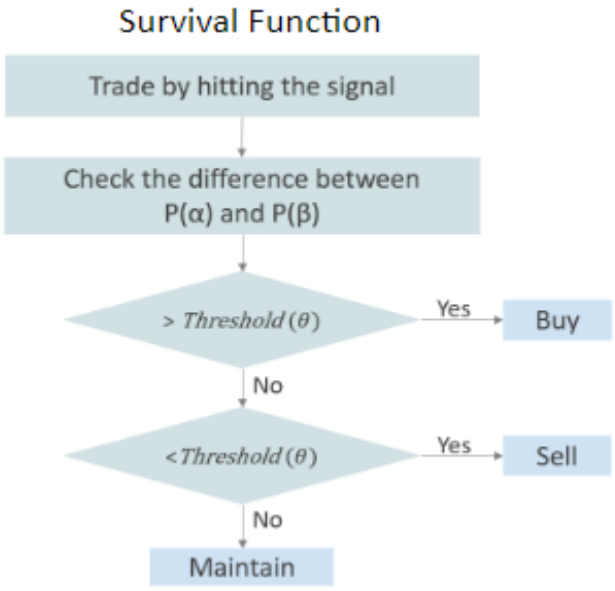

Figure 21. Trading Strategy of Survival Function

Parameters to be traversed:

a) The interval days we checked the trading signal (buy or sell)
b) The threshold settled for triggering the trading signal (same for event alpha and beta)

(2) GARCH & LSTM

For LSTM and GARCH, we converted their prediction results into daily stock returns. Using this information, we regularly evaluated whether the daily stock returns exceeded or fell below our established threshold to determine whether to buy or sell.

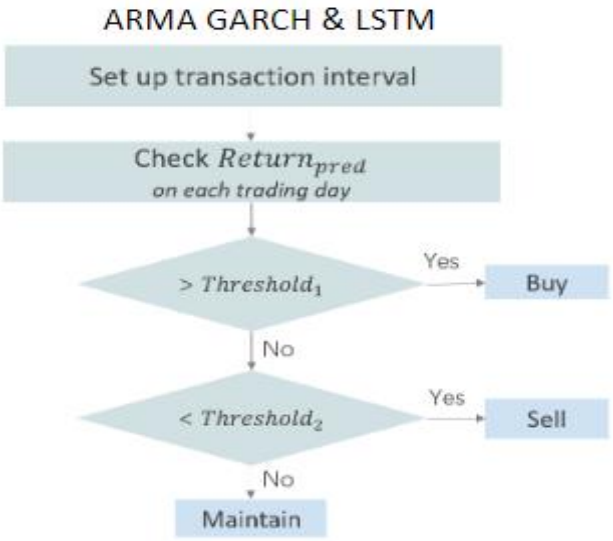

Figure 22. Trading Strategy of AMRA GARHC & LSTM

Parameters to be traversed:
a) The interval days we checked the trading signal (buy or sell)
b) The threshold settled for triggering the trading signal (separately settled for GARCH and LSTM)

(3) Multi-Factor

In the case of the multi-factor model, by regularly fitting the corresponding multi-factor regression equations for each stock, we identified undervalued stocks to buy and overvalued stocks to sell, thereby achieving portfolio rebalancing. The purchase quantity of each stock was weighted based on its alpha value, with the goal of maximizing returns while minimizing risk.

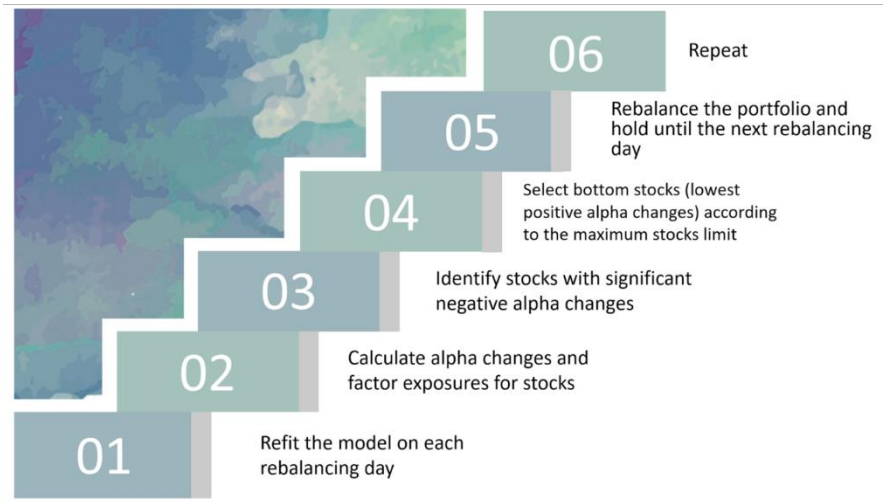

Figure 23. Trading Strategy of Multiple Factor Regression

Parameters to be traversed:

a) The interval days we rebalance the portfolio.
b) The upper limit of the stock we choose to hold.

9.3 Product & Optimization

For each simulated trading result, we treated it as the performance of a financial product. After fitting all the results, we initially obtained over 2100 products. We needed to filter these products to optimize our selections.

(1) Initial Filtering

Firstly, we believed that the number of trades in simulated trading should not be too small, otherwise there would be no difference from manually trading. Besides, we also filtered out products with minus return. Finally, we kept around 300 products for further optimization.

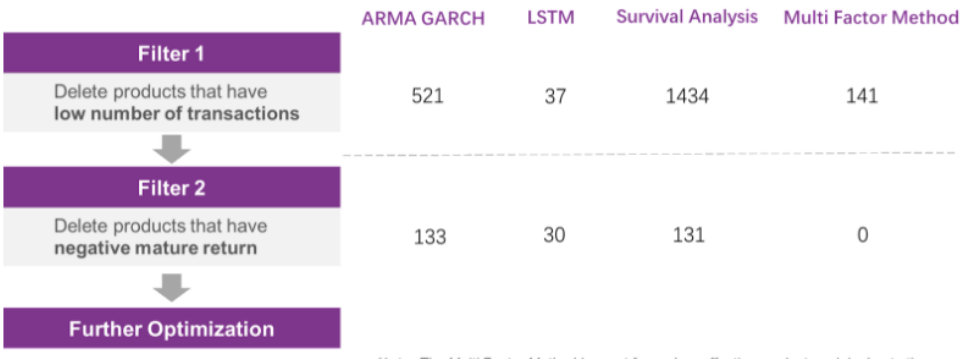

Figure 24. Filtering Stages

It is worth mentioning that the products fitted based on the multi-factor model were not retained. This is mainly because its return was greatly affected by the average market conditions of all 20 stocks, which might lead to losses.

(2) Efficient Frontier（NLP）& Monte Carlo

We believe that people's preferences for assets mainly fall into two categories: higher returns and lower risk. However, pursuing both goals simultaneously can lead to a conflict with the investment philosophy of "high returns, high risk" in investment theory. Therefore, achieving a balance between these two objectives is crucial.

To achieve this balance, we further filtered the available assets based on the Sharpe ratio and the Markowitz efficient frontier. The Sharpe ratio measures the ratio of the asset's excess return to its volatility. A higher Sharpe ratio represents a better risk-return tradeoff.

$$s_a = \frac{E[R_a - R_b]}{\sigma_a}$$

Here s_a refers to the Sharpe ratio of the financial product, $E[R_a - R_b]$ refers to the excess return of the product to the risk-free rate and σ_a is the volatility of the product's return.

The Markowitz efficient frontier is a nonlinear programming method that selects the portfolio with the highest expected return for a given level of risk based on all existing products. This results in a curve that can help us identify assets with higher returns for the same level of risk, which means the products near the curve will relatively have a higher Sharpe ratio.

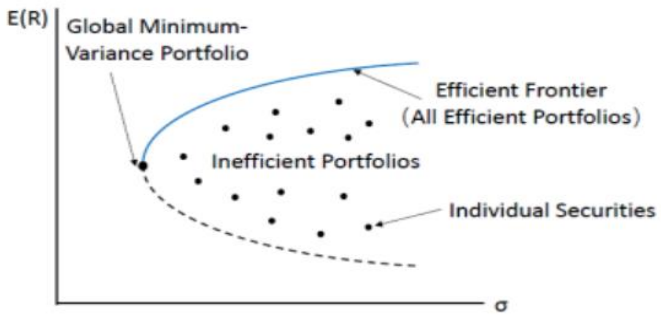

Figure 25. The Markowitz efficient frontier

After generating the efficient frontier based on the simulation results, we observed that the frontier was relatively sharp, and there were few asset points near it. This was mainly because the products we had retained after initial filtering distributed sparsely near the curve and all these points could not represent all possible asset allocations among the 20 stocks, which is necessary for generating the efficient frontier using the Markowitz approach.

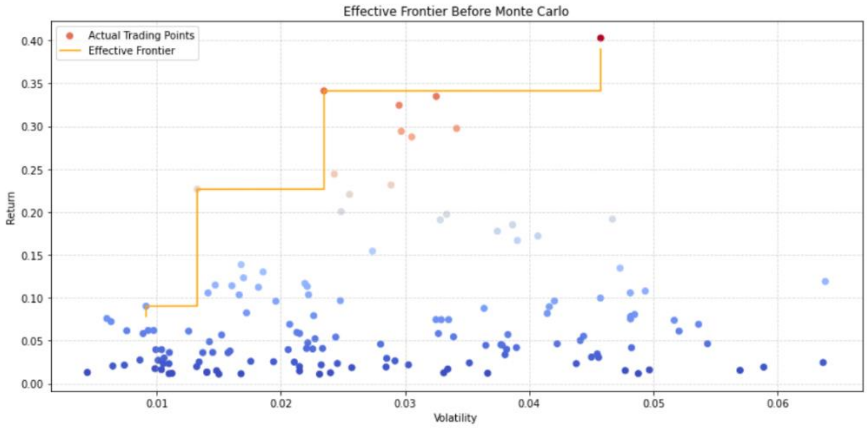

Figure 26. Effective Frontier before the Monte Carlo

Therefore, we further used the Monte Carlo method to randomly interpolate the gaps between the existing product points. This allowed us to generate additional points and re-draw the frontier using both the newly simulated points (blue) and the points generated from our trading (red). As a result, the frontier became smoother and passed through more actual product points, indicating that more products would be available for further allocation.

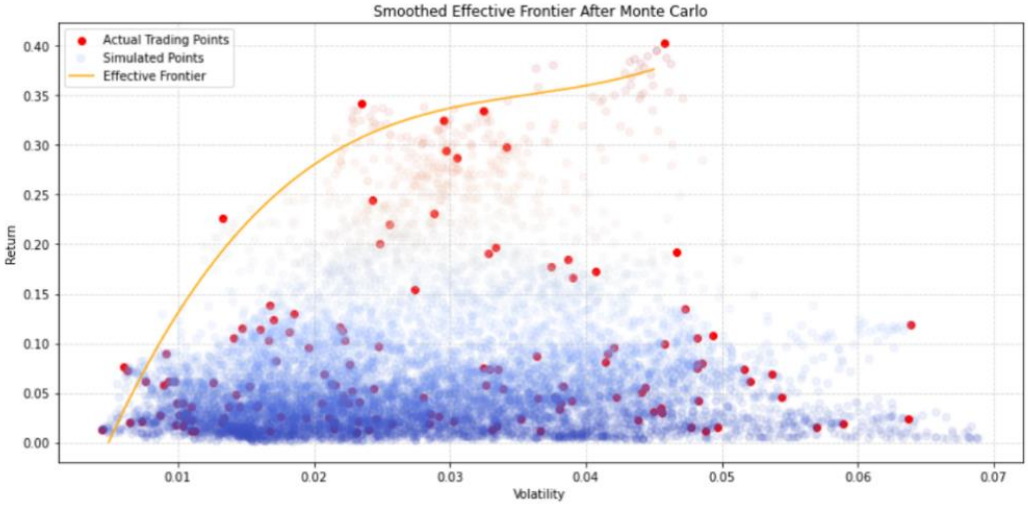

Figure 27. Smoothed Effective Frontier After Monte Carlo

At last, we selected 14 investment products for further goal programming by filtering the risk-return points that are close to the efficient frontier. These 14 products cover four stocks, including CGNPC, CNNP, TCL, and JSG. In addition to returns and risks, we also added maximum drawdown (largest back) and minimum investment as an optimizable indicator for future analysis.

strategy	stock_name	mature_return	volatility	step	Tradingcounts	minimum investment	Largest_back
GARCH_daily	CGNPC	1.30%	0.44%	3	6	1000	3.16%
GARCH_daily	CNNP	2.15%	0.74%	4	6	1000	7.48%
GARCH_daily	CGNPC	2.74%	0.75%	1	9	1000	3.16%
SUVIVAL_daily	CGNPC	3.96%	0.99%	1	11	1000	2.31%
LSTM_daily	CGNPC	6.15%	0.86%	2	5	1000	4.16%
LSTM_daily	CGNPC	9.02%	0.91%	2	6	1000	4.16%
GARCH_daily	TCL	11.48%	1.47%	3	5	5000	18.77%
LSTM_daily	JSG	13.86%	1.68%	2	6	5000	6.85%
GARCH_daily	TCL	24.41%	2.43%	1	8	10000	8.90%
LSTM_daily	TCL	29.41%	2.97%	1	5	10000	7.10%
GARCH_daily	TCL	29.74%	3.41%	2	7	10000	8.90%
GARCH_daily	TCL	32.44%	3.95%	2	5	30000	8.90%
LSTM_daily	TCL	34.13%	4.35%	1	5	30000	7.70%
GARCH_daily	TCL	40.28%	4.57%	2	5	50000	11.47%

Figure 28. Product samples after filtering

10. Optimization

10.1 Client Profile

Now we have a list of products, we also have a client's demand table with different levels. Please note that the values are just an example set by us, it varies for every different client.

Figure 29. Clients' profile with levels

10.2 goal programming using Solver

How can we customize the best investment plan for different clients? We use Excel Solver with goal programming to solve this problem.

We have three goals:

(1) Achieve the set target for return. (With deviations = d_{-1}, d_{+1})

(2) Keep the risk below or equal to the set target. (With deviations = d_{-2}, d_{+2})

(3) Ensure that the maximum drawdown is below or equal to the set target. (With deviations = d_{-3}, d_{+3})

The objective function is to minimize the total sum of all deviation ratios.

$$Z = \sum_{i=1}^{3} (d_{-i} + d_{+i})/t_i$$

We have 14 different products, and we set the allocation ratio of a client's total funds to each product as $weight_k$ (k=1 to 14). Here's an example of budget = 10000 SGD.

Budget = 10000					
Product	return	risk	maximum drawdown	weight	allocated value
1	1.30%	0.44%	3.16%	0.01	100
2	2.15%	0.74%	7.48%	0.01	100
3	2.74%	0.75%	3.16%	0.01	100
4	3.96%	0.99%	2.31%	0.01	100
5	6.15%	0.86%	4.16%	0.01	100
6	9.02%	0.91%	4.16%	0.01	100
7	11.48%	1.47%	18.77%	0.01	100
8	13.86%	1.68%	6.85%	0.01	100
9	24.41%	2.43%	8.90%	0.01	100
10	29.41%	2.97%	7.10%	0.01	100
11	29.74%	3.41%	8.90%	0.01	100
12	32.44%	3.95%	8.90%	0.01	100
13	34.13%	4.35%	7.70%	0.01	100
14	40.28%	4.57%	11.47%	0.87	8700
				sum_weight	
				1	

Figure 30. Client investment example

subject to the following six constraints:

(1) Goal 1 constraint with t_1 = target return

$$\sum_{k=1}^{14} weight_k * return_k + d_{-1} - d_{+1} \geq target\ return$$

(2) Goal 2 constraint with t_2 = set max risk

$$\sum_{k=1}^{14} weight_k * risk_k + d_{-2} - d_{+2} \leq set\ max\ risk$$

(3) Goal 3 constraint with t3 = set max drawdown

$$\sum_{k=1}^{14} weight_k * maximum\ drawdown_k + d_{-3} - d_{+3} \leq set\ max\ drawdown$$

(4) Budget constrain

$$\sum_{k=1}^{14} weight_k = 1$$

(5) All $weight_k$, d_i must be positive

$$\text{All variables} \geq 0$$

*(6) Ensure that goal i is not underachieved, this depends on the specific needs of each client

$$d_{-i} = 0.0$$

10.3 Result

Let's look at some typical demographic cases. Take middle class family as example, who prefer normal and stable investments and have middle risk tolerance. We artificially set a value for each dimension according to their levels, and then use Solver for optimization.

Clients Profile				
client	budget	risk	return	maximum drawdown
middle class family	50000	3%	15%	10%
retired seniors	10000	1%	10%	5%
rich guys	100000	5%	30%	15%
...	...	...	...	...

Figure 31. Client's profile

The optimization result shows a combination of products in different proportions.

Client investment					
products	**weights**	**actual investment**	**weighted risk**	**weighted return**	**weighted drawdown**
1	0	0	0.00%	0.00%	0.00%
2	0	0			
3	4.27431E-07	0.021371575			
4	0	0			
5	0	0			
6	0.082862081	4143.104071			
7	0.010681777	534.0888695			
8	0.026936752	1346.837588			
9	0.099612127	4980.606366			
10	0.155597893	7779.894637			
11	0.128025749	6401.287448			
12	0.14245312	7122.655992			
13	0.151428433	7571.421654			
14	0.20240164	10120.082			
	sum weights	sum investment			
	1	50000			

Figure 32. Optimized client's investment

It should be noted that there is no deviation because all goals have been met simultaneously, but not all clients' optimization results will be like this. If all goals cannot be met simultaneously, we will discuss with the client to decide which goal has a higher priority.

Goal Achievements	under	over
deviations	d-i	d+i
1. risk	0	0
2. return	0	0
3. maximum drawdown	0	0

proportional deviations	d-i/ti	d+i/ti
1. risk	0.00%	0.00%
2.. return	0.00%	0.00%
3. maximum drawdown	0.00%	0.00%

obiective	0.00%

target	autual
3.00%	3.43%
15.00%	30.00%
10.00%	8.62%

Figure 33. Achievement of objective

We have also conducted analyses for other typical demographics, such as retired seniors and rich people as the table shown before. Our optimization Excel file is attached with the report.

11. Conclusions

11.1 Outcome Discussion

(1) The products utilizing Survival Analysis have attributes of low return and low risk. Through our analytics, we find that the financial products of using survival analysis more fit for current and short-term investment in small amount.

(2) The Multifactor Method is not profitable. This is not able to construct profitable products in our analytics as this method is highly influenced by the market performance and new energy industry is in loss in our testing period which is Mar 2023. We do not suggest to utilize this method to do investment when the industry is in loss overall.

11.2 Limitations and Further Prospects

(1) The scope of stock is not wide enough.

For example, we may conduct stratified sampling based on the stocks' capitalization in the industry rather than just selecting the top 20 stocks by capitalization. Utilizing stratified sampling will make our stock samples become more common to represent the industry and have less bias. In addition, we can also choose stocks in multiple industries. In this way, stocks can be diverse and differentiated which will make it better to for diversification of risk and easier to create more diverse investment portfolio.

(2) The scope of test set is not wide enough.

In the future, we can choose longer period as test set which is not limited to 3 months. We can also use the rolling trading strategies and set rolling test window for all our predictive analytics.

(3) We do not have quantitative evaluation for customer risk.

Currently, the customer preference for risk is preset by us based on the experience. However, in the future, more quantitative and scientific method could be leveraged to collect real-world risk preference data from customers. For example, the conjoint analysis could be used to find different customers' preference for investment, and it can also include more investment preference indicators, not limited to return, and risk.

12. References

1. Aouni, B., Cola pinto, C. , & Torre, D. L. . (2014). Financial portfolio management through the goal programming model: current state-of-the-art. European Journal of Operational Research, 234(2), 536-545.
2. Lu Yin. (2022). Stock Price Prediction Bases on ARIMA-GARCH Model. Advances in Applied Mathematics, 11(1): 404-417.

3. Gao, G., Zhan, B. , Liu, L. , Jie, C. , & Wu, Z. . (2015). A survival analysis method for stock market prediction. International Conference on Behavioral. IEEE.
4. Sharma, H.P. and Sharma, D.K. (2006). A Multi-Objective Decision-Making Approach for Mutual Fund Portfolio. Journal of Business & Economics Research, 4, 13-24.
5. Tencia. (n.d.). Tencia/STOCKS_RNN: Stock price prediction with lstms in tensorflow. GitHub. Retrieved March 3, 2023, from https://github.com/tencia/stocks_rnn.