Summary Report Optiver 19

This version was compiled on May 28, 2023

510470582 510074058 510173199 510434207 500636170 500627341

Abstract

The objective of this report is to devise a model capable of predicting a stock's price volatility, using its order book as the basis for input. The chosen classifier, the Random Forest classifier, has been deployed to identify the most suitable models among the ARMA-GARCH, Linear Regression, and HAV-RV models. The most suitable model is defined as the one that can forecast volatility with the lowest MSE and QLIKE values, among other factors. We have adopted an iterative learning approach whereby training inputs are incorporated only when actual outputs match our predictions. In instances where the predictions are incorrect, these are recorded and reintroduced into the training set for continued learning. Additionally, we conduct periodic assessments of the classifier model after every 50 time ids. Through these approaches, we have managed to enhance the accuracy of our classifier in determining the most suitable model for the stock input from 43% to a commendable 95% in terms of out-of-sample accuracy.

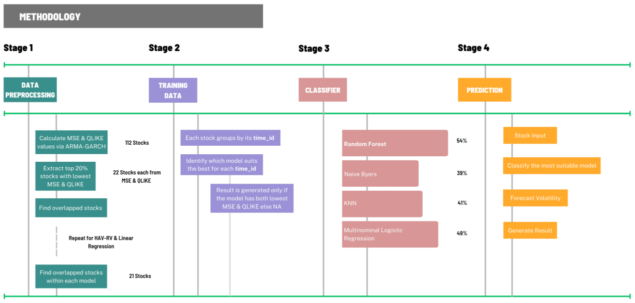

METHODOLOGY
Stage 1
Stage 2
Stage 3
Stage 4
DATA PREPROCESSING
TRAINING DATA
CLASSIFIER
PREDICTION
Calculate MSE & QLIKE values via ARMA-GARCH
112 Stocks
Extract top 20% stocks with lowest MSE & QLIKE
22 Stocks each from MSE & QLIKE
Find overlapped stocks
Repeat for HAV-RV & Linear Regression
Find overlapped stocks within each model
21 Stocks
Each stock groups by its time_id
Identify which model suits the best for each time_id
Result is generated only if the model has both lowest MSE & QLIKE else NA
Random Forest
54%
Naive Byers
39%
KNN
41%
Multinominal Logistic Regression
49%
Stock Input
Classify the most suitable model
Forecast Volatility
Generate Result

figure 1. key figure explaining our final product

1. Introduction

Research Question: What is the most suitable method to forecast future volatility.

Forecasting future stock price volatility is a critical component in the decision-making and risk management processes of financial managers. However, it poses a challenge for investors to select the most suitable forecasting model given a specific set of order book data. Instead of attempting to develop a singular model to accurately forecast volatility for all stocks, our project focuses on the performance of a classifier. This classifier can categorize a given stock based on its key features and, consequently, determine the most suitable method for forecasting its volatility.

Several forecasting models are commonly used in financial markets, including ARMA-GARCH, HAV-RV, and linear regression. These models offer different approaches to forecasting stock price volatility, and their performance varies depending on the unique features of each stock. Different stocks exhibit unique characteristics and respond differently to various market conditions. By considering features such as the bid-ask spread, historical volatility patterns, average stock price, and other relevant factors, we can classify stocks into three model groups.

The objective of our final product is to facilitate the decision-making process for financial managers and enable more effective risk management strategies.

2. Data Set

The data set collection comprises 112 individual stock order book snapshots provided by Optiver. Each stock order book includes approximately 3830 different buckets of orders, each spanning 600 seconds. The original data set contains the following features:

- **Time id** : Indicates a continuous time bracket, integer value
- **Seconds in bracket** : integer value in seconds
- **Bid price 1** : Highest bid price
- **Ask price 1** : Lowest ask price
- **Bid price 2** : Second highest bid price
- **Ask price 2** : Second lowest ask price
- **Bid size 1** : Order size of Highest bid
- **ask size 1** : Order size of lowest ask
- **Bid size 2** : Order size of second Highest bid second
- **Bid size 2** : Order size of second lowest ask
- **Stock id**: Id for a specific stock

All prices are rounded to 6 d.p

3. Data Preprocessing

Extra Features.

We further add on the features to help forecasting the volatility and classify stock

- **WAP**: Weighted average price (calculated only from best bid and ask)
- **VWAP**: Volume-weighted average price

- **BAS**: Bid Ask Spread for the stock
- **Price volatility**: Price volatility of rolling window size of 10 second

Forcasting methods. code reference

The folloing methods haven been selected to forecast volatility:

- ARMA-GARCH model
- HAV-RV model
- linear regression model

Linear Regression Model.
The regression analysis assigns weights based on recency and treats realized volatility as individual data points rather than a time series. For every unique time id in each stock, we employ weighted least squares to fit the model. This method helps mitigate the impact of outliers and increases efficiency by treating realized volatility as standalone data points, thus circumventing complex computations.

The selected feature to forecast volatility are Bid Ask Spread, total order and the weighted average price, with the general formula of

$$volatility = \beta_0 \ + \ \beta_1(WAP) \ + \ \beta_2(BAS) \ + \ \beta_3(order\ volume)$$

Where order volume is the sum of all bid and ask size

However, this model also comes with its own set of limitations. The assumption of linearity and normality might not be true, since there are at most 540 data (first 9 minuets), the normality assumption cannot be relied on Central Limit Theorem hence the validity and reliability of the model sometimes could not be guaranteed.

HAV-RV model.
The Heterogeneous auto regressive realised volatility (HAV-RV) model have the advantage of easy to understand and interpret and less computational cost to forecast volatility is modified to fit the short period of order book.

The modified model is Weighted Least Square (WLS) estimation with w_t defined as

$$w_t = \frac{RV_{t-1}}{\sqrt{RQ_{t-1}}}$$

where

$$RQ_t = \frac{M}{3}\sum_{i=1}^{M} r_{ti}^4$$

is the realised quarticity in period t.
The final formula to forecast the realised volatility is

$$RV_t = \beta_0 + \beta_1 RV_{t-1} + \frac{\beta_2}{5}\sum_{i=1}^{5} RV_{t-i} + u_t$$

Since it only consider the past 5 realised volatility due to the limitation of the amount of data we have, the model might not capture most important information out to forecast volatility.

ARMA-GARCH model.
Auto Regressive Moving Average (ARMA) and General Auto Regressive Conditional Heteroskedasticity Model (GARCH) have the advantage of capture long-term mean reversion of volatility, it is auto regressive, which means the weight applied to the observation can be adjusted to fit the past observations, it can account for asymmetry of volatility and capture near term persistence in volatility.

The model used is ARMA-GARCH(1,1) model which includes one past observation and one past error from ARMA, one past variance and one past squared residual for GARCH. To forecast volatility, the model simulates 30 orders of lagged square residual included in the model and 1000 orders of lagged variances included in model, then take the mean as the forcased volatility.

The limitation of the model include the ARMA does not capture any asymmetry volatility and there exist risk of over-specificity to a particular sample leading to reduce the accuracy of out of sample volatility forecasting.

Produce the preliminary training. Code reference

The initial training data for our product is produced through the following process.

1. Apply three forecasting methods to all the 112 stocks and calculate the mean square error (MSE) and empirical quasi-likelihood (QLIKE) for each individual time bracket on the data with extra features.

2. Select the top 20% of stocks that performs the best for each forecasting method

3. For each of the selected data, remove the feature of Time id, Seconds in bracket and stock id, take the mean value of of the rest of the feature for that specific time bracket, and label which forecasting method is most suitable for the given details in a new column “category actual”
   The benefit of choosing the top performing stocks in the time bracket is it generally have stronger features that a model needs to minimise the MSE and QLIKE value, hence it more beneficial for the classifier to do the cluster and classify which forecast method we should use on a given order book data.

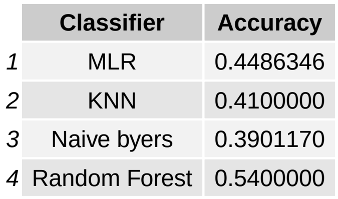
4. Model training

Classifier selection.

To determine which forecasting method is most suitable to avoid the weakness of each model, classifying stock based on the initial feature is important. We initially used Naive Byers , Random forest , K Nearest Neighbor (KNN) and multinomial logistic regression as classifier and choose the classifier of highest accuracy to further develop.

To evaluate the accuracy, the classifiers are trained use the initial training data, and then evaluated the accuracy use additionally 1000 unseen data from other time id and stock.

code reference for all classifier



	Classifier	Accuracy
1	MLR	0.4486346
2	KNN	0.4100000
3	Naive byers	0.3901170
4	Random Forest	0.5400000

figure 2. Table of accuracy of classifying forecasting method on inital training data

Iterative Learning Approach. Code Reference

From the initial result, The Random Forest classifier will be further developed to classify the stock into the corresponding forecasting method.

The random forest classifier is tuned with minimum node for sub tree is 10, with 500 sub trees in the random forest to capture the complex and highly correlated data feature.

Since ARMA-GARCH model use different approach to forecast realise volatility, it was determined that it either performs really well compared to the other two model, or it is not usable, therefore, the random forest was used in two steps, first determine if ARMA-GARCH is suitable to forecast the volatility, if not, further test which forecast method is more suitable between linear regression and HAV-RV model.

Further more, the accuracy was further improved by increase the sample size of training data. The approach has been tested on different ways, simply generate more training data by all time id in a stock or only reintroduce the inaccurate classification and re-evaluate the classifier.

From the result of small experiment, the accuracy of only re-introduce the inaccurate classification have significant improvement on the performance, hence the final model training method for our classifier is random forest classifier, evaluate the accuracy through classifying unseen data, re-introduce the inaccurate classified to sample training data, then re-train random forest classifier for every 50 time ids on each of the 100 stocks.

The following figure shows how our accuracy improved during our iterative process, all accuracy is evaluated from unseen data.

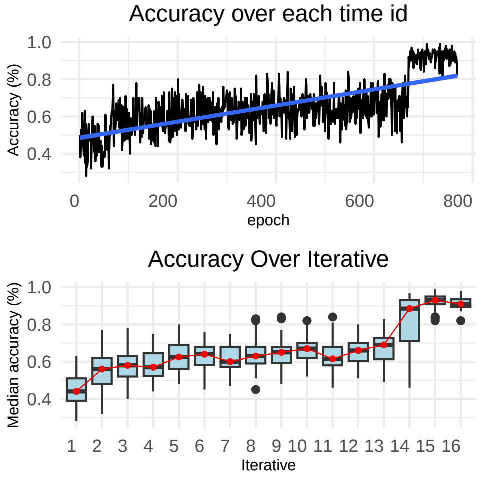


figure 3 Improvement of accuracy followes a trend of linear abline over each time id (up), and improvement of median accuracy over each iteration of re evaluating classifier (down)

From the initial data, the accuracy of classifying accurately is 0.52, however after 16 iterative of re evaluating and improving the model, the trained classier have accuracy of above 0.9 when classifying unseen data, which is a huge improvement and evidence of the iterative approach of adding new data works as expected.

5. Analysis

Looking at the final model, further examination is conducted into the classifier to understand why it functions as it does.

The first layer of random forest, classifying whether ARMA-GARCH model is suitable for forecast the volatility for the given data, is mostly dominant by the feature Bid ask Spread with the highest mean decrease accuracy of 138.23 (2d.p) follow by the best ask and bid size, with a mean decrease accuracy of approximately 75. The ARMA-GARCH forecast method is more suitable for less liquid market and less order size, the first classifier have an out of bag estimate of error at the rate of 18.3%.

The second layer of random forest, classifying whether HAV-RV or linear regression forecasting method is more suitable to forecast the volatility for the given data, the order size, BAS and price volatility is the key feature to classify which forecasting method we should use, with the more liquid market, larger order size and higher historical price volatility, HAV-RV method will perform better than linear regression method to forecast the future minuet volatility.

6. Results

code reference: final product

Our final product for forecasting price volatility is a function takes an input of any 9 continuous minuets of order book data, the function will output which forecasting method is most suitable for the given data and also the forecast volatility for the future minuet using the given model.

7. Limitation and Conclusion

Limitation.

One of the limitation for our product is even though it can accurately identify the most suitable forecast method to forecast the future volatility, it cannot guaranteed the range of error the forecasted volatility is, our product have a high chance to forecast the volatility through the best method out of the 3 selected method but cannot ensure that the prediction have small margin of error.

Furthermore, due to the initial data provided we only have a 10 minuets continuous bracket, our final product is only trained and tested on forecasting the future one minuets volatility, we cannot evaluate how our final product performs if the product is used to forecast a longer period of volatility.

Discussion.

If we had more time for the project, we could explore some potential optimizations. If new models have been discovered, we can apply those new models to our classifier and see if the accuracy improves. We can enhance the accuracy and effectiveness of our product by integrating novel models.

While the method of iterative approach of model training has improved the accuracy of our predictions, we can still try more advanced training strategies, such as active learning, to further enhance the performance of our models.

For those three models, there are some implications that can be made to optimize the final product. For the ARMA-GARCH model, we can try a different combination of past observation and past error from ARMA and past variance and past squared residual from GARCH instead of just one from each (1,1) to determine the most suitable parameters for each stock's volatility forecasting. For linear regression, we can explore different feature combinations and selection techniques. For HAV-RV, we can explore some alternatives for calculating historical and realized volatility.

Conclusion.

In this study, the random forest classifier showed the highest accuracy to classify which forecasting method is most suitable for given data, with the iterative learning approach to train our classifier, the accuracy for classifying most suitable method has increased up to average of 92%.
Through this project, we have successfully predicted stock price volatility utilizing machine learning and different models. Despite some limitations, we still see significant potential in this approach.

The benefit of this product is instead of fine turning a model to forecast volatility, it identify the most suitable basic forecasting method and hence forecast the volatility using identified method. The flexibility for this approach avoid the case of forcing to predict using an unsuitable method and produce a more stable outcome. The innovative in our product, including the way initial training data is selected, the iterative approach of improving accuracy of classifier and the idea of finding the most suitable method compare to others not only works on this project, but also the idea could be extends in future data related project as new approach to new problem.

contribution.

Ray:

- Turning ARMA-GARCH model
- Turning Naive Byers classifier
- Develop ideas to approach problem
- Report
    - Forecast method
    - Model selection
    - Figures and formula
    - Report formatting

Kelly:

- Turning ARMA-GARCH model
- Random forest fine turning
- Design key figure
- Report
    - introduction
    - Abstract
    - Discussion
    - Produce initial training data

Jasmine:

- Turning HAV-RV model
- Turning knn classifier
- Report
    - methodology

Han Wen:

- Key figure
- Turning HAV-RV model
- Presentation script
- Plot result

Mengjia:

- presentation script
- Turning linear regression forecast method
- Turning multinominal logistic regression method

Yizhe:

- Turning linear regression model

Appendix

git hub repository.

https://github.sydney.edu.au/rche9080/DATA3888_report

Data.

https://drive.google.com/drive/folders/1rFqjRh8ftAKbD0zAH4RyfTypFsjcyNlK?usp=sharing

Reference

Lazard Asset Management. (n.d.). Predicting volatility: Lazard research. https://www.lazardassetmanagement.com/docs/-m0-/22430/predictingvolatility_lazardresearch_en.pdf

VWAP definition. (n.d.). IG. Retrieved May 26, 2023, from https://www.ig.com/au/glossary-trading-terms/

vwap-definition#:~:text=VWAP%20is%20the%20abbreviation%20for

Ganti, A. (2021). Weighted Average Definition. Investopedia. https://www.investopedia.com/terms/w/weightedaverage.asp

Tidyverse. (n.d.). Www.tidyverse.org. https://www.tidyverse.org/

Simple Data Frames. (n.d.). Tibble.tidyverse.org. https://tibble.tidyverse.org/ Create Elegant Data Visualisations Using the Grammar of Graphics. (2019). Tidyverse.org. https://ggplot2.tidyverse.org/

A Grammar of Data Manipulation. (n.d.). Dplyr.tidyverse.org. https://dplyr.tidyverse.org/index.html

Package “rugarch.” (2020). https://cran.r-project.org/web/packages/rugarch/rugarch.pdf

Liaw, A., Wiener, M., Maintainer, A., & Liaw. (2015). Title Breiman and Cutler’s Random Forests for Classification and Regression Description Classification and regression based on a forest of trees using random inputs. https://cran.r-project.org/web/packages/randomForest/randomForest.pdf

A Short Introduction to the caret Package. (n.d.). Cran.r-Project.org. https://cran.r-project.org/web/packages/caret/vignettes/caret.html

Title Recursive Partitioning and Regression Trees Depends R (>= 2.15. (2015). https://cran.r-project.org/web/packages/rpart/rpart.pdf

partykit/partykit. (2023, May 27). GitHub. https://github.com/partykit/partykit

cran. (2023). GitHub - cran/e1071: This is a read-only mirror of the CRAN R package repository. e1071 — Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. GitHub. https://github.com/cran/e1071

class function - RDocumentation. (n.d.). Www.rdocumentation.org. Retrieved May 27, 2023, from https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/class

cran/cvTools. (2021, June 7). GitHub. https://github.com/cran/cvTools/blob/master/R/cvTool.R