Portfolio Optimization using Volatility Prediction

Abstract

The goal of this project is to construct an investment portfolio that offers varieties of depth needed to efficiently manage risks while maintaining an affordable price through using only the resources available to the average consumer. With asset management firms claiming upwards of 1.5% per risk-managed portfolio, affordability and the ability to create portfolios without the expertise of outside firms has become a priority. As will be determined, the best method to opt for when making these risk-adjusted portfolios is to invest in cost efficient ETFs. An ETF, or an exchange traded fund, is a type of security composed of different proportions of a multitude of stocks. By offering diversification and cost-effectiveness, investing in ETFs stands as the ideal alternative for ordinary consumers to build a portfolio with a composition of stocks through expensive consultants and brokers. The right mix of stocks and bonds drives the variance and returns of the constructed portfolio. The mix is typically revisited every month and the portfolio is rebalanced if necessary. However, change in the return and variance of the portfolio within a month timeframe can hurt the portfolio performance. A suitable technique for predicting the variance and the return one month ahead is critical. This article discusses a portfolio rebalancing technique along with various forecasting methods for standard deviation , the single most driver of the portfolio risk vg . Variance is predominating factor for the equity portfolio performance and detailed investigation was dedicated to the variance forecasting process

Introduction

The approach to build a model for selecting the most viable stocks/ETFs is centered around the sharpe ratios of the portfolio of the selected stocks, a measure of risk-adjusted returns. Traditionally, sharpe ratios are calculated using the historic mean of excess returns divided by the historical standard deviation, but it is preferable if we can work with the sharpe ratio representing the future. Creating models for the mean and standard deviations of the time series becomes crucial in that context. The three main parts of the portfolio construction are described below

Finding the Right Mix

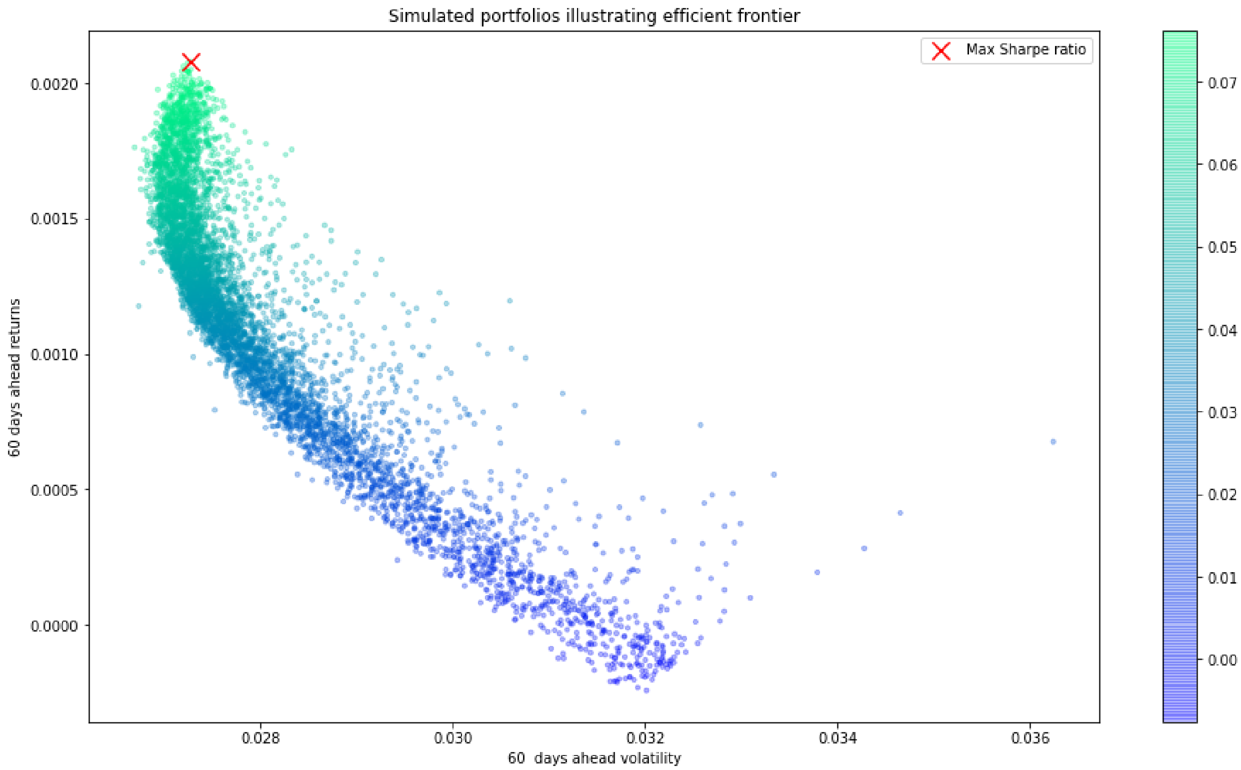
The process begins by choosing a set of random numbers between 0 and 1 to act as weights on the stocks composed in the portfolio. These numbers should all add up to one and the size of the set is however many stocks there are in the portfolio. This will allow us to ensure that the best combination is chosen. After this, a time series (spaced 1 day) of the price percentage return of the portfolio should be constructed using the stocks from the portfolio. This includes assigning a random number to each stock in the portfolio [the random number reflects the quantity of the stock], multiplying it with the stock price and summing the values for all the stocks in the portfolio to derive the value of the portfolio on a given day and using these portfolio values to calculate the daily percentage changes . This time series represents the historical values of the constructed portfolio. Next, the 10-day rolling standard deviation and return of the time series mentioned above must be calculated. A rolling window will be used in order to prevent any outliers from certain periods of time that would skew the data.

Forecasting the Return and Volatility of the Constructed Portfolio

Next step is to find the difference series [more suitable for linear regression models, since the difference series is close to stationary] for the 10-day rolling standard deviation and mean. After having found the difference series, a linear regression model should be utilized in order to determine the historical trend of the 10-day rolling window standard deviations and means, which will then be used to make predictions for the future. Since, of course, the series in hand doesn't resemble a perfectly linear shape, the linear regression model has errors that need to be accounted for. To use the linear model, a detrending has to be conducted in order to assess whether or not the linear regression was a viable predictive model. If the detrended linear regression, or the errors, is normally distributed and the R^2 value is high , we could use the linear regression result for predicting the future standard deviation. If the error is normally distributed, but the R^2 is poor, we can say that the model should be enhanced with more independent variables to improve the R^2 e.g. various lagged time series.We are facing this problem in our dataset but will continue with the poor R^2, since we our goal is to describe a broad brush concept.Finally, with the information gathered, a prediction of future standard deviations for the next 30 days can be developed using the historical slope of the linear regression model of the standard deviation differences. The predicted standard deviation differences should be added to the last observed standard deviation value to derive the prediction for the standard deviation. After having repeated this entire process using the rolling windows of the means, models have been made for both the prediction of means and standard deviations over the course of the next 30 days.

Sharpe ratio calculation using predicted return and the volatility

Next, the crucial sharpe ratio value can be calculated by dividing the means by the standard deviations [forecasted] for all the possible random values that created different mixes of stocks corresponding to different portfolios [The impact of the risk free rate is ignored while calculating sharpe ratio]. Using these sharpe ratios, a chart should be made that plots mean vs standard deviation values and whichever random values results in the highest sharpe ratio is the weighting that the portfolio should be made for.



Comparing volatility models

The lineThe linear regression based prediction is never adequate for stock price prediction. Models e.g. ARMA+GARCH is an industry standard for this. New advancement in Neural Network/LSTM has shown better performance in predicting stock prices.

Comparisons from a number of volatility models will be presented here.

<u>Linear Regression on Rolling Volatility Data</u>

y→ Timeseries of the portfolio value
X→ Time in days
y_std→ 10 days rolling std
y_mean-->10 days rolling mean
y_std_diff→ Daily change in the 10 days rolling std
y_mean_diff→ Daily change in the 10 days rolling mean

Prediction equations
y_std = y_std[t_observed]+_std_diff

y_mean = y_mean[t_observed]+_mean_diff

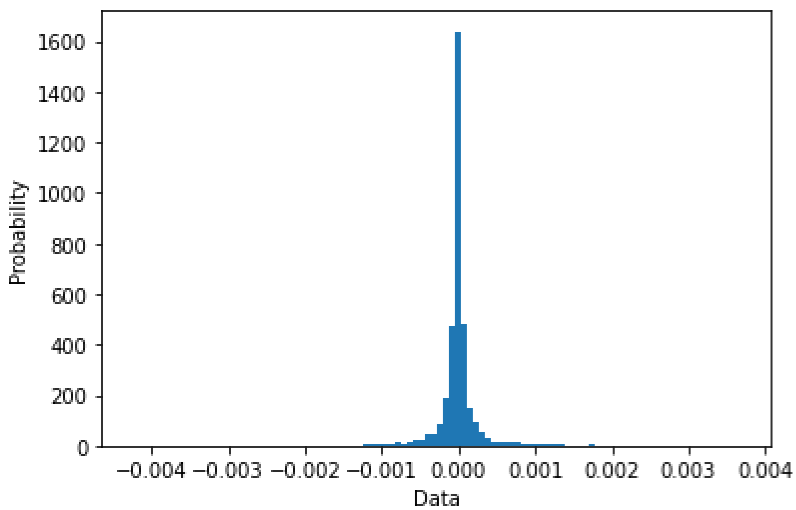


Figure 0: Error distribution is normal

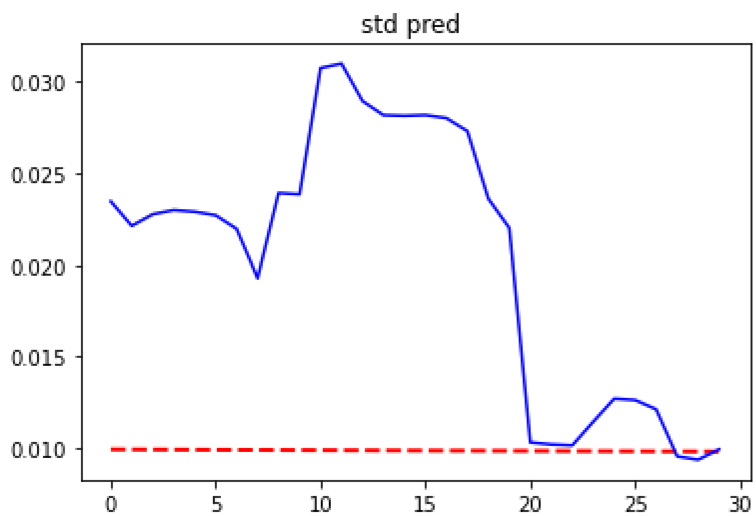


Figure 1: Linear Regression Forecast

Detail of the model implementation (described above)

Rolling Window Size	Forecasting Horizon	P-Value	AIC	R-Squared
10 days	30 days	0.951	-3.351×10^4	0

NOTE: Model is not suitable for use due to unsatisfactory results

ARIMA-GARCH on Rolling Volatility Data

Autoregressive Model

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + .. + \beta_p Y_{t-p} + \epsilon_1$$

y_t is the dependent variable of regression at time t

.

y_{t-1} is the dependent variable of regression at time t-1

.

y_{t-2} is the dependent variable of regression at time t-1

.

.

.

β_1 is the regression coefficient of y_{t-1}

.

β_2 is the regression coefficient of y_{t-2}

.

.

.

ε_t is the independent and normally distributed error

ε_t is not auto-correlated

Moving Average Model

$$Y_t = \alpha + \epsilon_t + \phi_1\epsilon_{t-1} + \phi_2\epsilon_{t-2} + .. + \phi_q\epsilon_{t-q}$$

ε $_t$ is the dependent variable of regression at time t

.

ε $_{t-1}$ is the dependent variable of regression at time t-1

.

ε $_{t-2}$ is the dependent variable of regression at time t-1

.

.

.

β $_1$ is the regression coefficient of y_{t-1}

.

β $_2$ is the regression coefficient of y_{t-2}

.

.

.

ε $_t$ is the independent and normally distributed error

ε $_t$ is not auto-correlated

<u>ARIMA Model (combination of previous two models)</u>

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + .. + \beta_p Y_{t-p}\epsilon_t + \phi_1\epsilon_{t-1} + \phi_2\epsilon_{t-2} + .. + \phi_q\epsilon_{t-q}$$

Data means that there is a trend and an error. From linear regression to ARIMA, the trend is better captured

ARCH/GARCH Model

$$\sigma_t^2 = \omega + \alpha\varepsilon_{t-1}^2 + \beta\sigma_{t-1}^2$$

It is modeling the standard deviation of the error, where error is random. The distribution of the error is normal or t.

ARIMA Configuration

Number of autoregressive lag terms: 1
Number of moving average lag terms: 1
Number of seasonal lag terms: 2
Optimization method used: LBFGS

GARCH Configuration

ARIMA residuals are used as GARCH data fit

Number of error-phase lag terms: 2
Number of standard deviation-phase lag terms: 2

The final prediction is an addition of the ARIMA forecasting and GARCH forecasting (standard deviation)

The data was split into train and test datasets. The length of the test dataset is 60 days

Because one phase lag is being used for the autoregressive and moving average models, the forecasting is heavily dependent on the preceding training point.

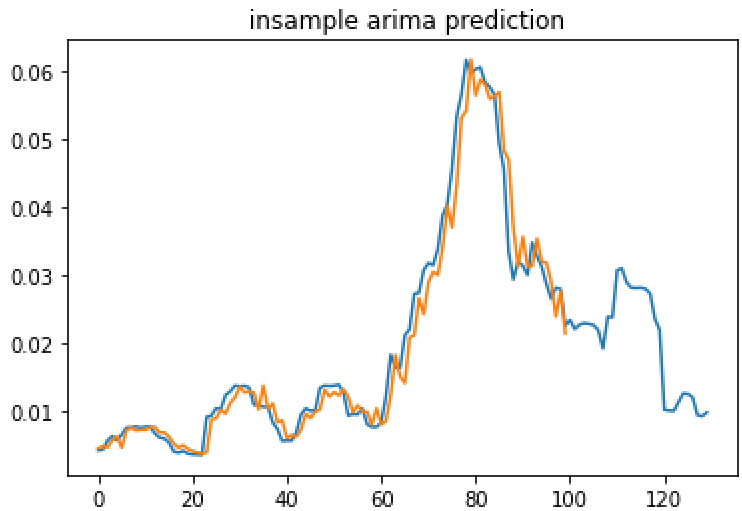


Figure 1: ARIMA in sample fit

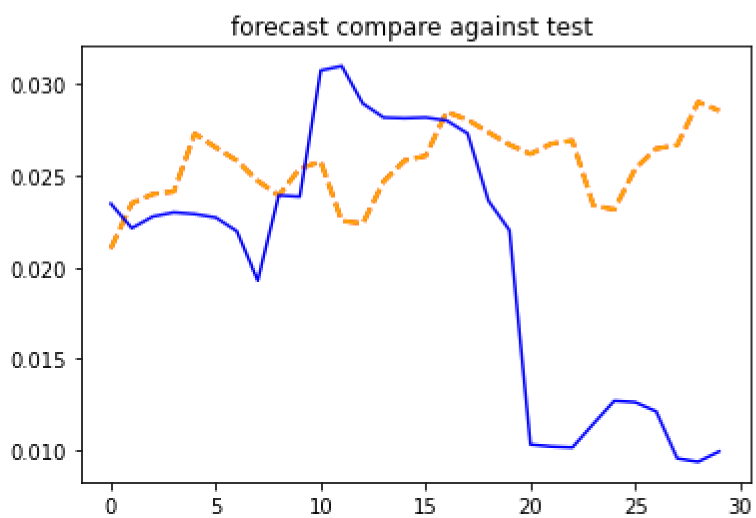


Figure 2: ARIMA GARCH prediction

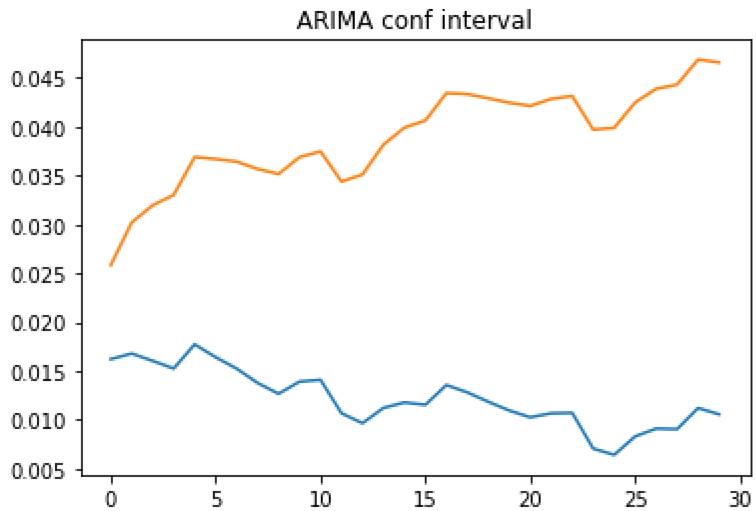


Figure 3: Confidence Interval Of Prediction

Note: Orange depicts prediction and blue depicts observed data

ARIMA Results Table

Rolling Window Size	Forecasting Horizon	P-Value	AIC	Log-Likelihood
10 days	30 days	0.054	-33251.306	16627.653

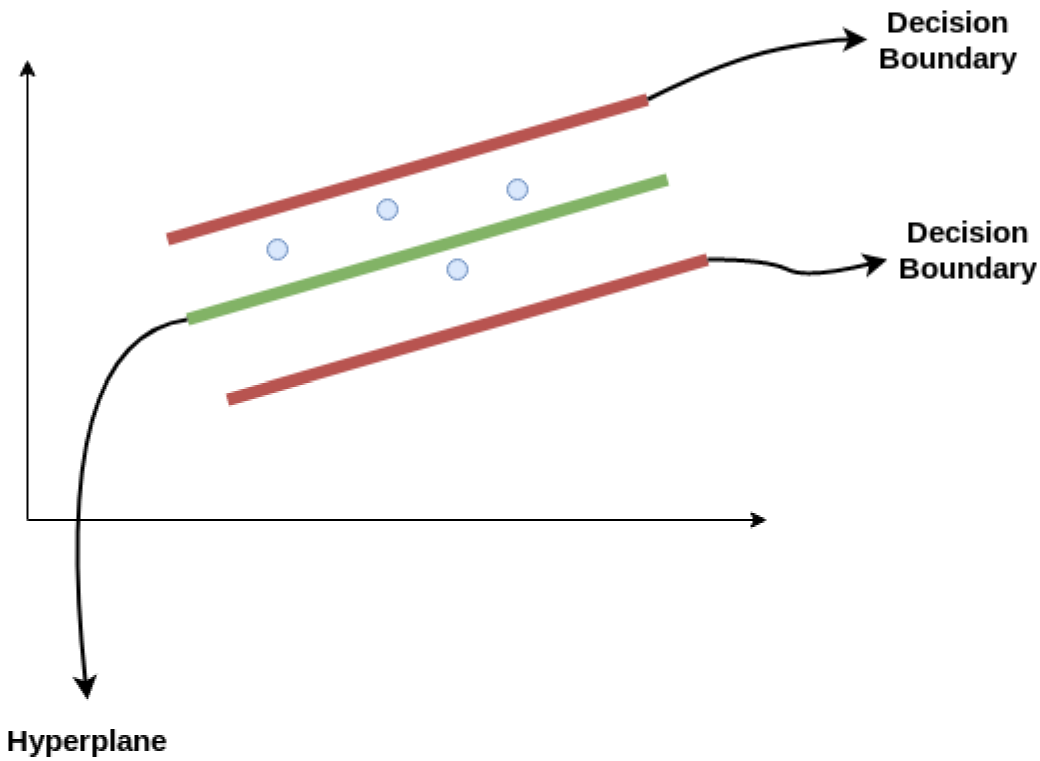
GARCH Results Table

Rolling Window Size	Forecasting Horizon	P-Value	AIC	Log-Likelihood
10 days	30 days	0.641	16302.3	-8148.13

NOTE: The GARCH model didn't fit well so only ARIMA will be used. The final prediction will be an ARIMA prediction (as given below)

```
MSE Test Sample
9.175535188772978e-05
```

SVR on Rolling Volatility Data



Consider these two red lines as the decision boundary and the green line as the hyperplane. **Our objective, when we are moving on with SVR, is to basically consider the points that are within the decision boundary line.** Our best fit line is the hyperplane that has a maximum number of points

The data was split into train and test datasets. The length of the test dataset is 60 days

SVR Configuration

Carnel: RBF
Scoring Mechanism: Mean-Squared Error
Parameter Selection: Grid Search

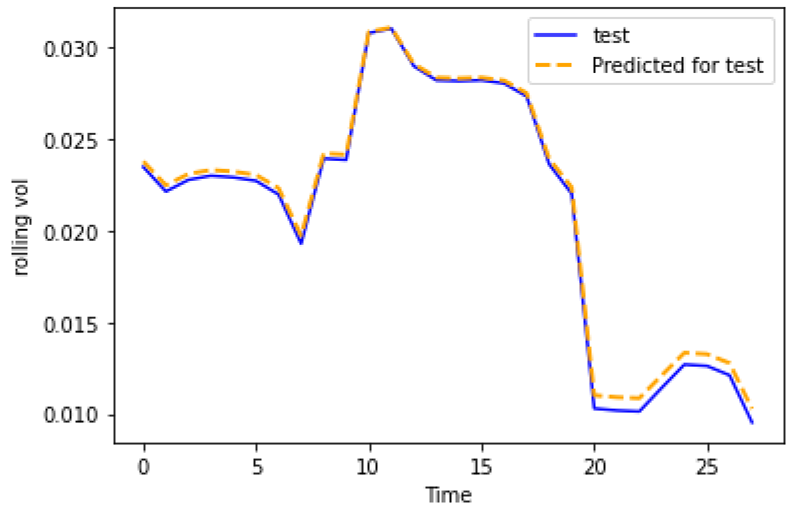


Figure 4: SVR Forecast

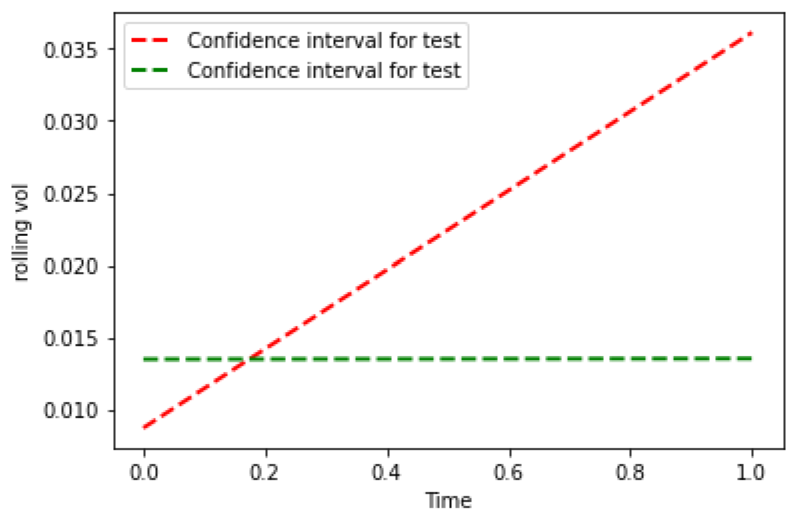


Figure 5: Confidence Interval Of Prediction

```
MSE Test Sample
1.880934869094722e-07
```

Prophet On Rolling Volatility Data

Similar to a generalized additive model (GAM), with time as a regressor, Prophet fits several linear and non-linear functions of time as components. In its simplest form;

y(t) = g(t) + s(t) + h(t) + e(t)

where:

g(t)

- trend models non-periodic changes (i.e. growth over time)

s(t)

- seasonality presents periodic changes (i.e. weekly, monthly, yearly)

h(t)

- ties in effects of holidays (on potentially irregular schedules ≥ 1 day(s))

e(t)

- covers idiosyncratic changes not accommodated by the model

In other words, the procedure's equation can be written;

$$\begin{aligned} y(t) = {} & \text{piecewise\_trend}(t) + \\ & \text{seasonality}(t) + \\ & \text{holiday\_effects}(t) + \\ & \text{i.i.d. noise} \end{aligned}$$

Prophet is essentially "framing the forecasting problem as a curve-fitting exercise" rather than looking explicitly at the time based dependence of each observation.

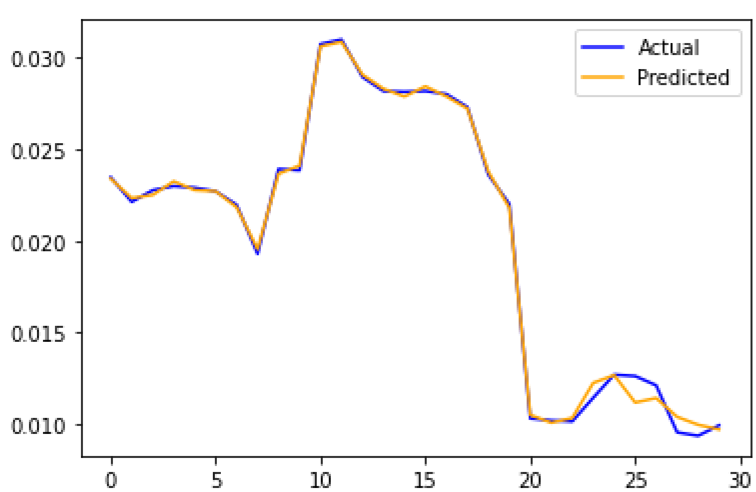


Figure 6: Prophet Forecast

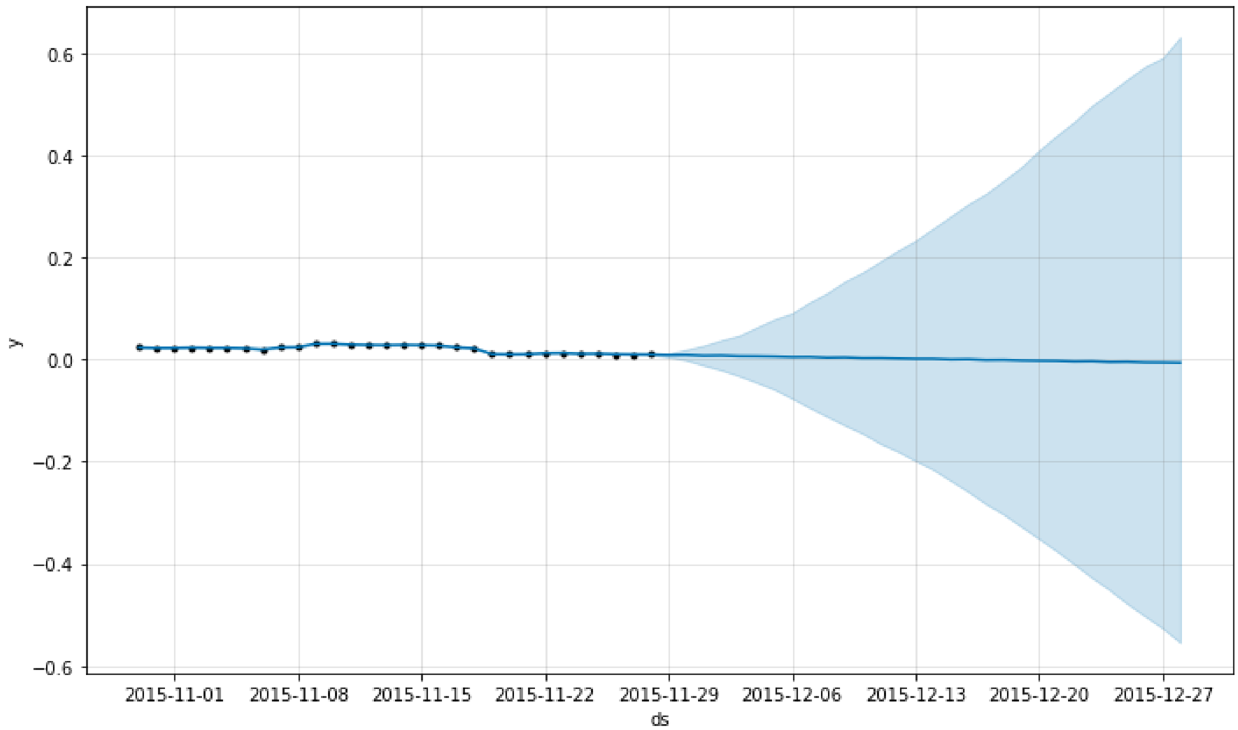


Figure 7: Confidence Interval Of Prediction

```
MSE Test Sample
1.7252555846070363e-07
```

<u>Gradient Boosting</u>

Gradient boosting involves three elements:

1. A loss function to be optimized.
2. A weak learner to make predictions.
3. An additive model to add weak learners to minimize the loss function.

1. Loss Function

The loss function used depends on the type of problem being solved.It must be differentiable, but many standard loss functions are supported and can be customized.For example, regression may use a squared error.

2. Weak Learner

Decision trees are used as the weak learner in gradient boosting.Specifically regression trees are used that output real values for splits and whose output can be added together, allowing subsequent models outputs to be added and “correct” the residuals in the predictions.Trees are constructed in a greedy manner.

3. Additive Model

Trees are added one at a time, and existing trees in the model are not changed.A gradient descent procedure is used to minimize the loss when adding trees.Weak learner sub-models or more specifically decision trees are added to reduce losses.

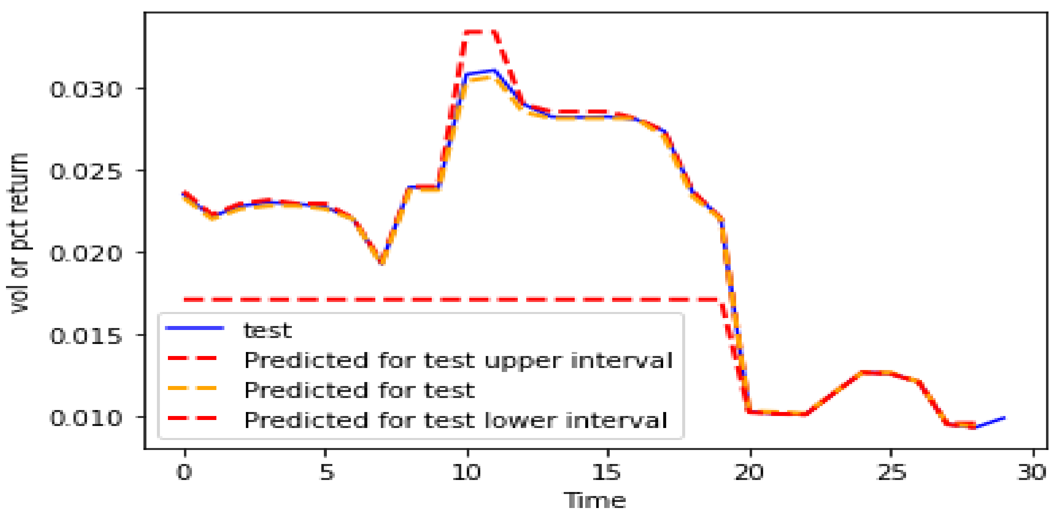


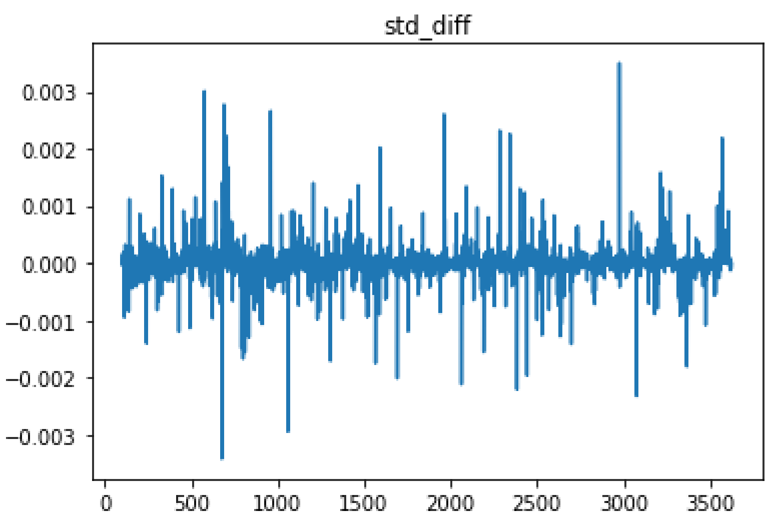
Figure 8: Gradient Boost Forecast and confidence interval

```
MSE Test Sample

3.612655833867181e-08
```

Appendix

A0: Stationarity check for the 10days rolling standard deviation on the percent difference time series



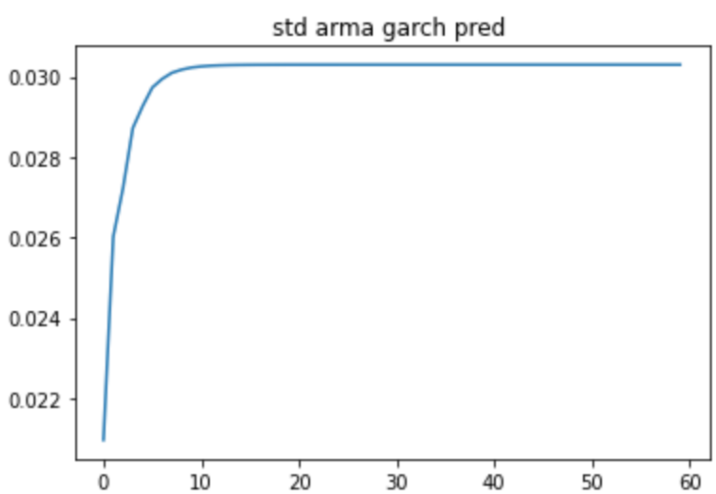
<u>A1:</u>

Various ARIMA configurations tried to reach the best outcomes. The scenarios tried are presented below.

**WHEN TESTING VOLATILITY MODELS ALL SECURITIES HAVE SAME WEIGHT INSIDE THE PORTFOLIO

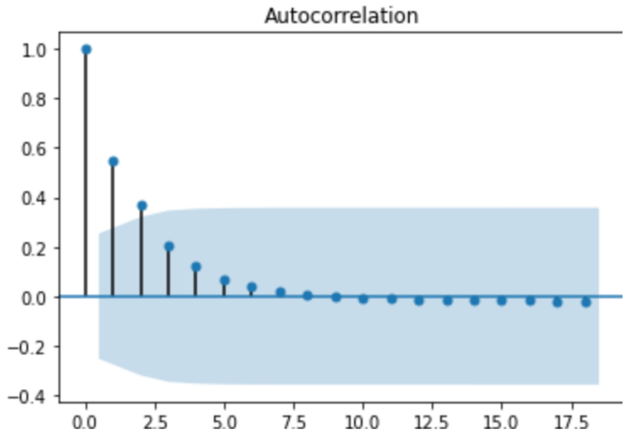
We will choose the best volatility model and use it to find the optimal portfolio using random weights of securities in the portfolio

Model	Coefficient Name	P-value Max	Log-likelihood	Sum-Square Error	AIC	BIC
ARIMA-ARCH (1, 1) - (2, 2)	ARIMA Coefficients	0.039	9552.376	-	-19094.752	-19063.816
ARIMA-ARCH (1, 1) - (2, 2)	ARCH Coefficients	1.640e-5	9828.22	-	-19650.4	-19631.9
ARIMA-ARCH (1, 1) - (5, 5)	ARIMA Coefficients	0.0	9620.501	-	-19237.002	-19224.61
ARIMA-ARCH (1, 1) - (5, 5)	ARCH Coefficients	1.958e-3	9958.68	-	-19905.4	-19868.2
ARIMA-GARCH (1, 1) - (1, 1, 1)	ARIMA Coefficients	0.0	9620.501	-	-19237.002	-19224.611
ARIMA-GARCH (1, 1) - (1, 1, 1)	GARCH Coefficients	7.883e-2	10261.3	-	-20512.5	-20481.5
ARIMA-GARCH (1, 1) - (5, 1, 5)	ARIMA Coefficients	0.0	9620.501	-	-19237.002	-19224.611
ARIMA-GARCH (1, 1) - (5, 1, 5)	GARCH Coefficients	0.930	10046.9	-	-20069.8	-19995.4

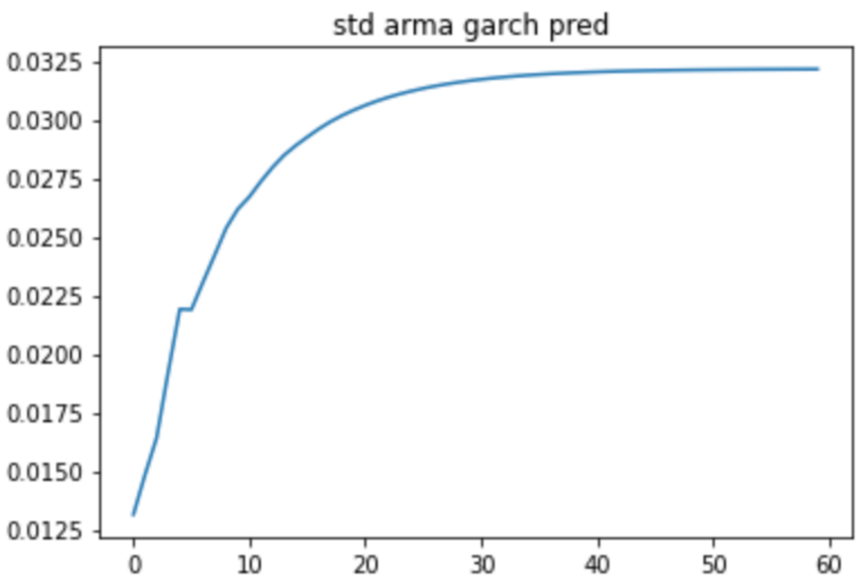


Prediction of

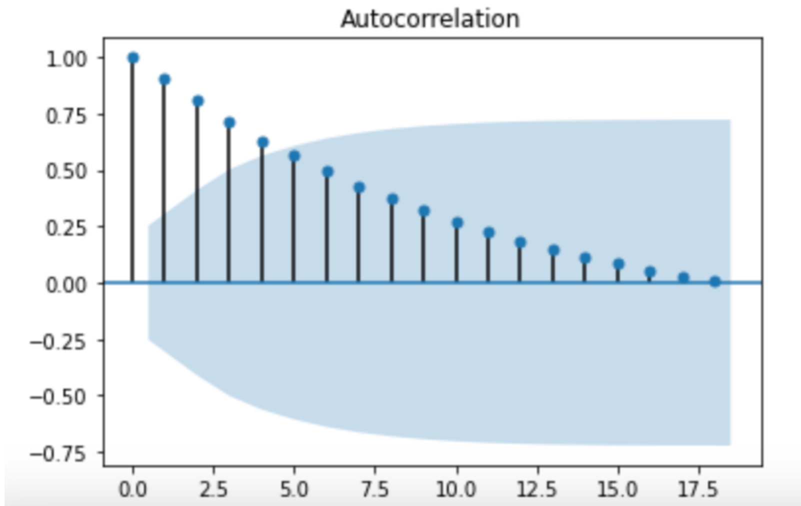
ARIMA-ARCH
(1, 1) - (2, 2)



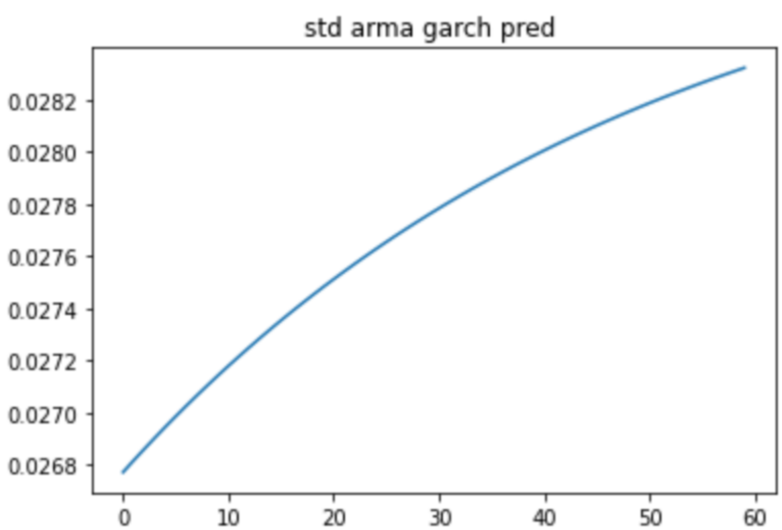
Autocorrelation sharply decaying



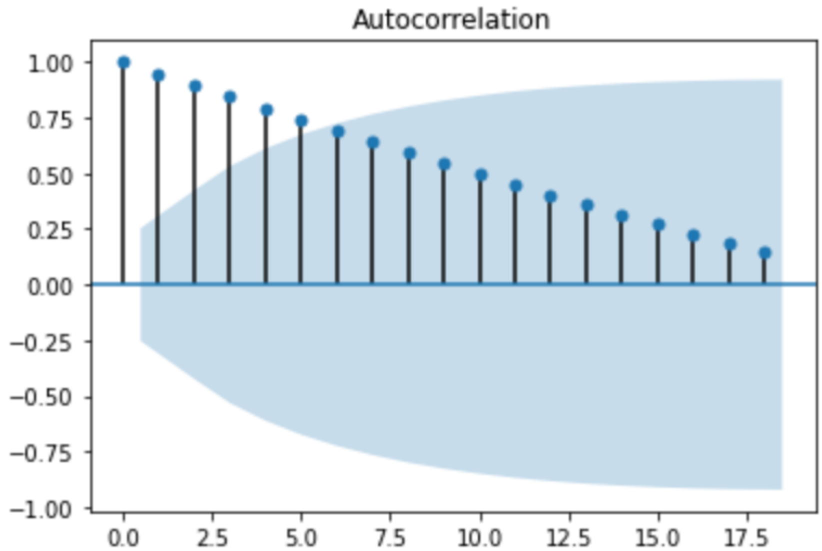
Prediction of
ARIMA-ARCH
(1, 1) - (5, 5)



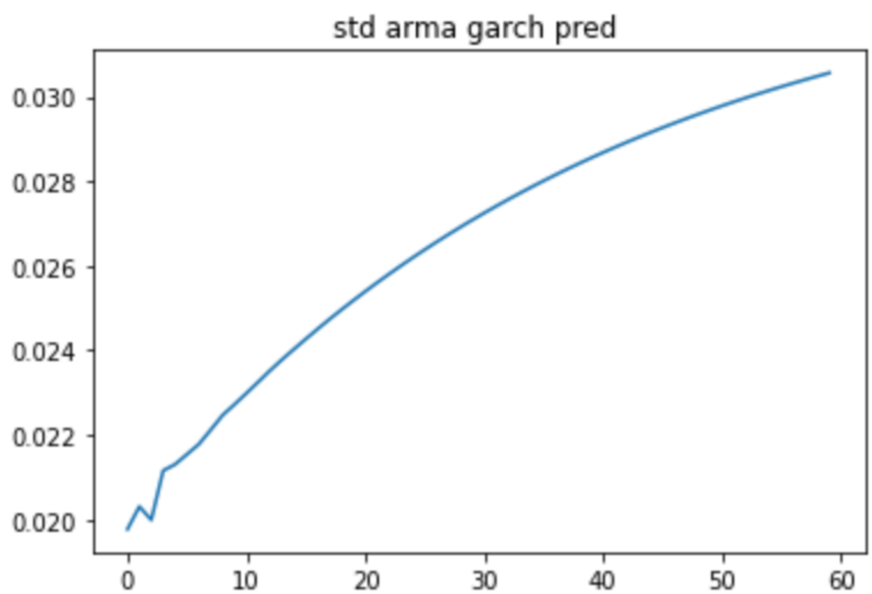
Autocorrelation relatively slowly decaying



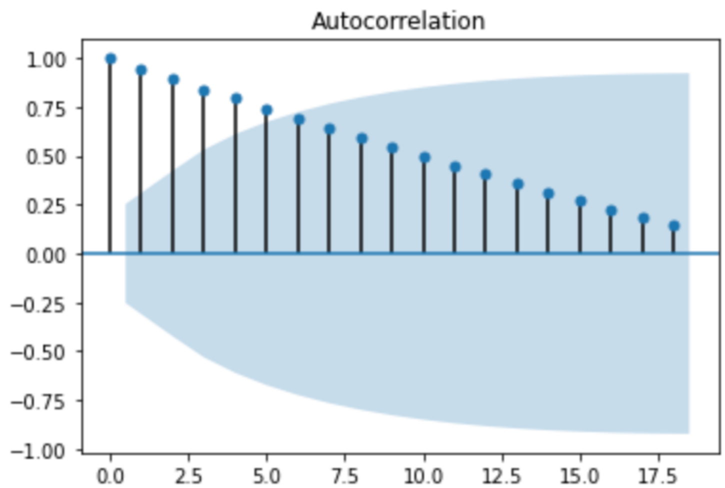
Prediction of
ARIMA-ARCH
(1, 1) - (1, 1, 1)



Autocorrelation slowly decaying



Prediction of
ARIMA-ARCH
(1, 1) - (5, 1, 5)



Autocorrelation slowly decaying

Definitions

- T-value: The t-value measures the size of the difference relative to the variation in your sample data.

- P-value: In null hypothesis significance testing, the p-value is the probability of obtaining test results at least as extreme as the results actually observed
  - The distribution of the coefficients of regression and the ARIMA model drives the P-value

- AIC: The AIC statistic is defined for logistic regression as follows (taken from "The Elements of Statistical Learning"):

  AIC = -2/N * LL + 2 * k/N

  Where N is the number of examples in the training dataset, LL is the log-likelihood of the model on the training dataset, and k is the number of parameters in the model.

  The score, as defined above, is minimized, e.g. the model with the lowest AIC is selected.

- BIC:
  The Bayesian Information Criterion, or BIC for short, is a method for scoring and selecting a model.

  It is named for the field of study from which it was derived: Bayesian probability and inference. Like AIC, it is appropriate for models fit under the maximum likelihood estimation framework.

  The BIC statistic is calculated for logistic regression as follows (taken from “The Elements of Statistical Learning“):

  BIC = -2 * LL + log(N) * k

  Where log() has the base-e called the natural logarithm, LL is the log-likelihood of the model, N is the number of examples in the training dataset, and k is the number of parameters in the model.

  The score as defined above is minimized, e.g. the model with the lowest BIC is selected.

  The quantity calculated is different from AIC, although can be shown to be proportional to the AIC. Unlike the AIC, the BIC penalizes the model more for its complexity, meaning that more complex models will have a worse (larger) score and will, in turn, be less likely to be selected.

References

- SVT
  - Sethi, Alakh. “Support Vector Regression In Machine Learning Analytics.” *Analytics Vidhya*, Analytics Vidhya, 27 Mar. 2020, cdn.analyticsvidhya.com/wp-content/uploads/2020/03/SVR1.png.