# Portfolio Optimization using Volatility Prediction

Rian Sengupta

# Abstract

Ex-Ante portfolio optimization is an important aspect of investment and asset management. It employs robust optimization and forecasting methods to achieve best portfolio return under an acceptable level of risk for a future time horizon. A Monte Carlo Simulation [5] based Mean Variance optimization technique is used when constructing investment portfolios. However, being that change in the return and variance of the portfolio within a short time frame can hurt the portfolio performance, and that historical mean and variance are not enough for future predictions since history isn’t repeated all the time, suitable techniques for forecasting the standard deviation and the return are critical. The methodologies that will be tested are autoregression and machine learning based forecast techniques to capture future volatilities. Machine learning based techniques are not heavily used in calculating volatility in the equity space, so the potential will be investigated in this work. To prevent the aforementioned issues involving high volatilities over future time horizon, forecasting will be made for the next day. Ensuring that the volatility is being forecasted every day and if the forecasting for the next day is significantly different, the mean variance optimization can be run again to rebalance the portfolio. The same thing must be repeated as has been done with volatility with historical means which are being used as opposed to non-historical means since volatility is the key component for equity price movement. For the final analysis, the volatility forecasting model with the best Sharpe Ratio after running the mean variance optimization for a given time period will be selected as the best contender for producing a properly balanced portfolio. The simplicity and transparency of this approach will encourage retail investors to adopt modern portfolio theory-based optimizations for personal portfolios. Yahoo Finance is used as the data source to retrieve daily return of various equities of choice e.g. GOOG,AAPL,MSFT,AMZN,FB and XLU. The daily return time series is used to produce a comparison report across different volatility models and mean variance optimized portfolios. Statistical and Machine Learning libraries available in Python distributions are used to implement the optimization and forecasting functionalities.

# Introduction

The approach to construct a portfolio with the most viable equities/ETFs is centered around Mean Variance based portfolio optimization [2] and the Sharpe Ratios[2], a measure of risk-adjusted returns, of the optimized portfolio. The Mean Variance optimization technique proposed by Harry M[arkowitz](https://www.math.ust.hk/~maykwok/courses/ma362/07F/markowitz_JF.pdf) assumes that the return of each individual volatile asset is normally distributed with their mean and covariance being time-stationary. The historical covariance matrix is used to optimize the mean/ variance combination of the portfolio. However, neither assumptions hold true in real life scenarios.

The rolling mean and volatility based on the weighted combination of daily returns of the risky assets [equities] that constituting the portfolio will be used for the Mean Variance optimization. The optimal combination of rolling means and volatilities are searched out of a large set of portfolios generated using randomly generated weights. In turn, Sharpe Ratios are calculated using the rolling mean of returns divided by the rolling standard deviation.

Traditionally, Sharpe Ratios were calculated using the historic mean of excess returns divided by the historical standard deviation, but it is preferable to work with the Sharpe Ratio representing the future. Using historical rolling mean and standard deviations of the time series to forecast future rolling mean and standard deviation respectively becomes crucial in that context. This paper investigates various volatility forecasting models in terms of accuracy e.g. log likelihood, overfitting (AIC/BIC*[ Please refer to the appendix B for the definitions] )*. These models are applied to forecast rolling volatilities. The spectrum of models includes Linear Regression, ARIMA+GARCH [3], Support Vector Regression [1], and Gradient Boosting [8]. Volatility is the single most important driver of equity risk and most of the work is done to identify a better choice of volatility model. However, only the Linear Regression models are used to forecast the rolling mean which is hoped to be extended in future work. The Ex-Ante Sharpe Ratio presented in the paper is calculated using the Linear Regression-based forecasting of rolling mean and the volatility, since the model is simple, and the training time is reasonable.

# Portfolio Construction

This paper adopts Monte Carlo Simulation [5] based Mean Variance optimization technique for portfolio construction.

When picking a portfolio of equities, different levels of risks can be taken depending on the goals, with risk simply meaning standard deviation of returns. But regardless of the willingness to accept risk, returns can be maximized by picking the combination of equities that has the highest returns for this given level of risk or volatility. It only makes sense to invest in the portfolios i.e. combinations of equities, that have the highest returns for a given level of risk. Monte Carlo simulation uses a probability distribution to generate random weights as a parameter for random portfolio return to simulate thousands of portfolios and pick the ones with the highest returns by plotting returns against standard deviation. These portfolios are called “The Efficient Frontier”.

The process begins by generating a set of random numbers between 0 and 1 using Python/Numpy libraries. The library uses a uniform distribution over [0,1).

The generated random numbers act as weights of the equities composing the portfolio. These numbers should all add up to one and the size of the set is however many equities there are in the portfolio. After this, a time series (spaced by 1 day) of the price percentage return of the portfolio should be constructed using the equities from the portfolio. This includes assigning a random number to each equity in the portfolio [ the random number reflects the quantity of the equity], multiplying it with the equity price and summing the values for all the equities 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 is representative of the historical daily percentage change 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 eliminate any outliers from certain periods of time that would skew the data. The choice of the size of the rolling window is arbitrary at this point. A relatively shorter window size is selected to capture more perturbations in the volatility time series. The future work list will include testing of few methodologies for selecting an appropriate size for the rolling window.[4]

# Forecasting the Return and Volatility of the Constructed Portfolio

Since, the short-term forecasting are more accurate than the long-term ones, the paper proposes daily forecasting of the portfolio return for the next day. A large change in the forecasted value against today's value should trigger the need for rebalancing of the portfolio. The rebalancing should be able to bring the volatility down.

Next step is to find the difference series out of the 10 days rolling standard deviation and mean series constructed from the percentage return series of the constructed portfolio[more suitable for linear regression and ARIMA models, since the difference series is close to stationary]. The stationarity property [6] ensures that the regression beta remain valid for the forecasting time period. After having found the percentage difference series, a linear regression model is used 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 our goal is to describe a broad-brush concept for Ex Ante portfolio optimization. Finally, with the information gathered, a prediction of future standard deviations for the next day 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 for the next day.

# Sharpe Ratio Calculation using Predicted Return and the Volatility

Next, the Sharpe Ratio values can be calculated by dividing the means by the standard deviations for each possible random sets of weights that created different mixes of equities 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. Figure:1 represents the Efficient Frontier along with the max Sharpe Ratio found from the portfolio construction exercise.

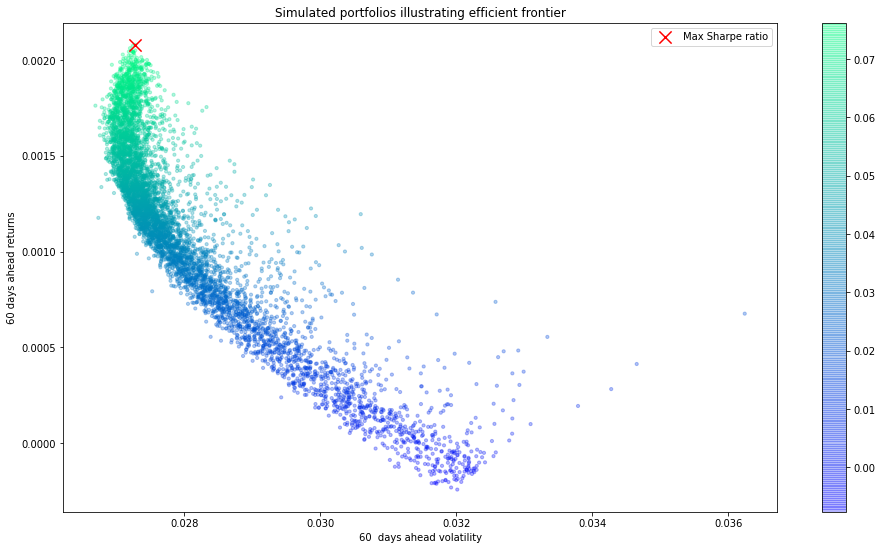


Figure 1: Error distribution is normal

# Comparing Volatility Models

The Linear Regression-based forecasting is never adequate for equity price forecasting. Models e.g. ARMA+GARCH is an industry standard for this. New advancement in Machine Learning has shown better performance in predicting equity prices. A systematic approach has been taken to investigate a handful of models for forecasting the volatility of portfolio return. Each of the selected model is configured to the best of our knowledge to avoid both under and overfitting. The training and forecasting errors along with the confidence interval are recorded for each model. The goal is to make an informed decision while selecting a volatility forecasting model for calculating the Sharpe Ratio for each portfolio composition generated via random trial.

Comparisons of a few volatility models are presented here. The volatility time series used for prediction is the 10 days rolling standard deviation of the percentage return time series of the constructed portfolio

Table:1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model Name** | **In-Sample prediction accuracy** | **Out of Sample prediction accuracy**  **[ 1 day ahead forecast]** | **Worst Confidence Interval for prediction** | **Overfitting Management** | **Verified Model Assumptions** |
| Linear Regression | Not good, Measured in terms of Rsquared | Out of sample prediction was not conducted | N/A | N/A | * Stationary time series for portfolio return * Regression error follows normal distribution * Regression errors are not auto correlated |
| \*ARIMA+ GARCH | Pretty good, Measured in terms of Minimum Square Error | Minimum Square Error is 9.17e-05 | The confidence interval is within the range of 1% to 4.5% where the predicted value ~ 2.8% | Model parameters used are as follows   * Number of autoregressive lag terms: 1 * Number of moving average lag terms: 1 * Number of seasonal lag terms: 2   Since, just a few lag terms are used and there are ~5350 samples in the training dataset, overfitting is not a glaring concern. The AIC [7] value for the for the fitted model is -33251.306 | * Stationary time series for portfolio return * Regression error follows normal distribution |
| Support Vector Regression | Pretty good, Measured in terms of Minimum Square Error | Minimum Square Error  1.88e-07 | The confidence interval is within the range of 0.5% to 2.5% where the predicted value ~ 1% | Model configuration e.g. Kernel used is as RBF.  Model parameter named epsilon specifies the epsilon-tube within which no penalty is associated in the training loss function with points predicted within a distance epsilon from the actual value. A low epsilon value [ 0.001 ] ensures that a penalty is applied during the training for more number of points and helps to avoid overfitting | * No stationarity and normality assumption are needed. |
| Prophet from Facebook | Pretty good, Measured in terms of Minimum Square Error | Minimum Square Error  1.72e-07 | The confidence interval is within the range of 50% to 60% where the predicted value ~ 1% | Overfitting is controlled by changing the model parameters e.g. fourier.order and seasonality.priorscale. Increasing Fourier.order allows the model for fitting the seasonal patterns that change more quickly. Seasonality.priorscale controls the amount of regularization applied to model the seasonality. Regularization is important to avoid overfitting.  Relatively low values are used for fourier.order and seasonality.priorscale to avoid overfitting. | * Normality of the training errors are required |
| Gradient Boosting | Pretty good, Measured in terms of Minimum Square Error | Minimum Square Error 8.63e-05 | The confidence interval is within the range of 1.8% to 3.5% where the predicted value ~3% | The model configuration parameters used are as follows.  max\_tree\_dept:3 learning\_rate:.1min\_samples\_leaf:35 min\_samples\_split:19  The parameters above are key in controlling overfitting. Low learning tree depth along with large number of leaves and splits help in avoiding overfitting | No stationarity and normality assumption is needed. |

## Model Details

**AR - Auto Regressive Process**

The AR model, or the autoregressive model, predicts the observed value against the previous observations as the regressor. To accomplish this, multiple slopes based on historical data are calculated corresponding to previous observations and summed together to produce resultant predictions. The equation also includes the random error represented as

Portfolio returns from previous days are considered as regressors in this case. The current day return is linearly related to a combination of previous days’ returns. The process equation is as follows

Yt is the dependent variable of regression at time t

.

Yt-1 is the dependent variable of regression at time t-1

.

Yt-2 is the dependent variable of regression at time t-2

.

.

.

is the regression coefficient of Yt-1

.

is the regression coefficient of Yt-2

.

.

.

is the independent and normally distributed error

is not auto-correlated

**MA - Moving Average Process**

The AR model, or the autoregressive model, predicts the observed value against the previous error as the regressor. To accomplish this, multiple slopes based on historical data are calculated corresponding to previous errors and summed together to produce resultant predictions. The equation also includes the random error represented as εt

Portfolio returns’ fitting errors from previous days are considered as regressors in this case. The current day return is linearly related to a combination of previous days’ fitting errors. The process equation is as follows

Yt is the dependent variable of regression at time t

.

t-1 is the dependent variable of regression at time t-1

.

t-2 is the dependent variable of regression at time t-2

.

.

.

is the regression coefficient of t-1

.

is the regression coefficient of t-2

.

.

.

is the independent and normally distributed error

is not auto-correlated

**ARIMA**

The ARIMA model, or the autoregressive moving integrated average model, is a linear combination of AR and MA

**GARCH – Generalized Auto Regressive Conditional Heteroscedasticity Model**

The GARCH models the error produced by the auto-regressive moving average model. The motivation is to capture the observed volatility clustering of equity returns.

The GARCH provides an autoregressive relationship between current volatility and the previous days’ volatilities along with ARIMA errors. The process equation is as follows

Yt =

= ARIMA without error terms

=

is conditional variance

is the independent and normally distributed error

is not auto-correlated

*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 30 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 2, Figure 3 and Figure 4 below depict the in-sample fitting, one day ahead forecast and confidence interval respectively.

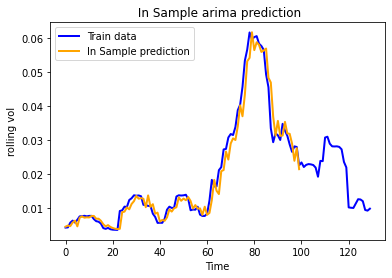


Figure 2: ARIMA in sample fit

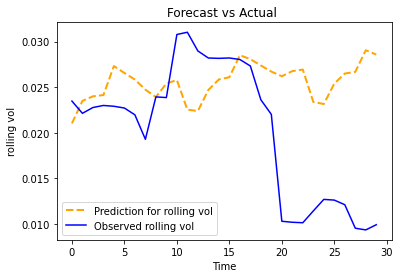


Figure 3: ARIMA GARCH prediction

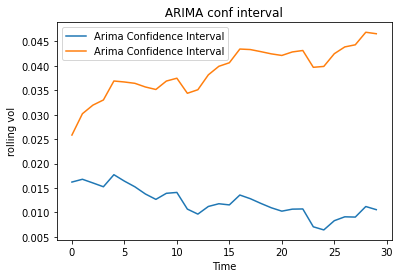


Figure 4: Confidence Interval of Prediction

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)

**SVR - Support Vector Regression**

This approach uses a +kernel to map the lower dimensional input data into an n- dimensional space and identifies a \*\*hyperplanes that captures the best fitting of the observed data. The hyperplane with the maximum number of points represents the regression fitting. Maximum and minimum boundaries are also set to eliminate the outliers

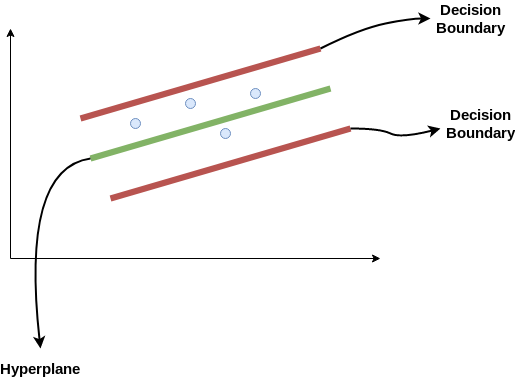


Figure 5: SVR Concept

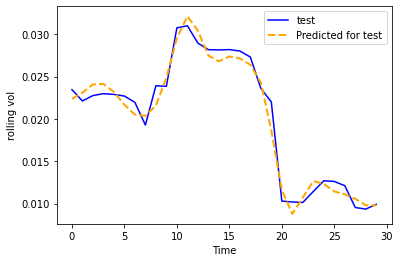


Figure 6: SVR Prediction

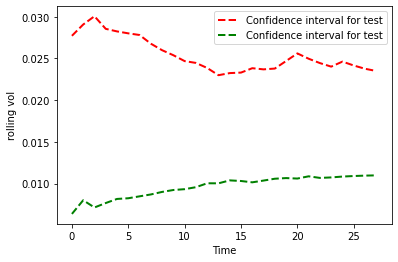


Figure 7: Confidence Interval of Prediction

## Prophet from Facebook

Prophet is similar to a generalized additive model ([GAM](https://en.wikipedia.org/wiki/Generalized_additive_model))[9], with time as a regressor. It 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

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

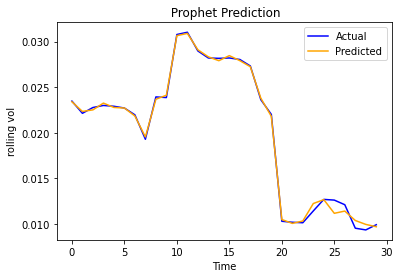


Figure 8: Prophet Prediction

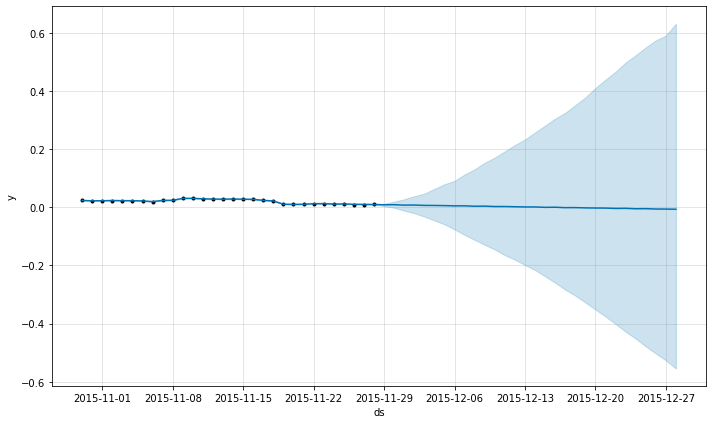


Figure 9: Confidence Interval Of Prediction

## Linear Regression

Linear regression is a [linear](https://en.wikipedia.org/wiki/Linearity) approach to modelling the relationship between a [scalar](https://en.wikipedia.org/wiki/Scalar_(mathematics)) response and one or more explanatory variables (also known as [dependent and independent variables](https://en.wikipedia.org/wiki/Dependent_and_independent_variables)). The case of one explanatory variable is called [*simple linear regression*](https://en.wikipedia.org/wiki/Simple_linear_regression); for more than one, the process is called multiple linear regression

In linear regression, the relationships are modeled using [linear predictor functions](https://en.wikipedia.org/wiki/Linear_predictor_function) whose unknown model [parameters](https://en.wikipedia.org/wiki/Parameters) are [estimated](https://en.wikipedia.org/wiki/Estimation_theory) from the [data](https://en.wikipedia.org/wiki/Data). Such models are called [linear models](https://en.wikipedia.org/wiki/Linear_model). Most commonly, the [conditional mean](https://en.wikipedia.org/wiki/Conditional_expectation) of the response given the values of the explanatory variables (or predictors) is assumed to be an [affine function](https://en.wikipedia.org/wiki/Affine_transformation) of those values; less commonly, the conditional [median](https://en.wikipedia.org/wiki/Median) or some other [quantile](https://en.wikipedia.org/wiki/Quantile) is used. Like all forms of [regression analysis](https://en.wikipedia.org/wiki/Regression_analysis), linear regression focuses on the [conditional probability distribution](https://en.wikipedia.org/wiki/Conditional_probability_distribution) of the response given the values of the predictors, The independent variable used in this work as regressor is time

where:

Timeseries of the portfolio value

t

Time in days

The similar shape of the regression error is used for predicting both rolling mean and rolling volatility

## Gradient Boosting

Gradient boosting-based regression involves three elements:

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

### 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.

### Weak Learner

Decision trees are used as the weak learner in gradient boosting. Specifically, regression trees are used that output real values from splits based on the available features and incrementally builds an ensemble by training each new instance to emphasize the training instances previously mis-modeled. Trees are constructed in a greedy manner.

### 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.

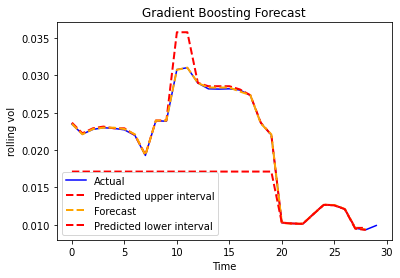


Figure 10: Gradient Boosting Forecast

# Conclusion

This article addresses two key aspects of Ex-Ante portfolio optimization. It employs Mean Variance portfolio optimization technique to find the right balance across various assets constituting the portfolio in terms of risk and return and at the same time and investigates various forecasting techniques to determine the Ex-Ante portfolio value. Monte Carlo based approach for Mean Variance portfolio optimization offers a transparent way to determine the optimal choice of the portfolio mix to attain max Sharpe Ratio. The approach demonstrates a good balance between accuracy and the cost of running the optimization. Additionally, analytical approach of finding Ex Ante max Sharpe Ratio requires forecasting of the correlation coefficients across the returns of the risky assets. The optimization approach articulated in this paper only requires forecasting of the mean return and volatility for a given portfolio, but not the correlation. Aggregation across the return values of the individual assets constituting the portfolio includes the effect of correlation forecasting. Ability to avoid forecasting the correlation coefficients across the risky assets along with the transparency makes the Monte Carlo based approach a well-suited option for Ex Ante portfolio optimization.

The article has conducted a thorough investigation across various forecasting techniques to suite the need of the portfolio in hand. Forecasting portfolio returns constituting risky assets is a daunting task. There is no silver bullet to that problem. Investigation includes basic methods e.g. Linear Regression, current industry standard for forecasting stock volatilities e.g. ARIMA+GARCH and various nonlinear machine learning methods. It has been observed that nonlinear regression methods have outperformed the traditional methods e.g. ARIMA+GARCH in terms of both the forecasting error and tighter confidence interval specifically in the regimes manifesting sharp fluctuation in the volatility. Nonlinear regression methods do not demand strong assumptions like stationarity and normality. This flexibility is crucial for modeling portfolio returns which are non-stationary in nature and do not manifest normality in regression error. There are two classes of non-linear regression methods have been investigated e.g. Tree based and Hyper Plane based. The Gradient Boosting is a Tree based nonlinear regression technique. The limitation in using Tree-based regression method is that the forecast of extreme values is bound by the historical extreme values. The technique does not do a good job in forecasting unforeseen events. However, the overfitting is controlled well through regularization. On the other hand, SVR, a Hyper Plane based nonlinear regression technique demonstrates better handling of forecasting extreme events at the cost of higher degree of overfitting.

Table :1 compares various volatility forecasting methods in terms of the accuracy and confidence of the outcome. Support Vector Regression outperforms other models in terms of both accuracy and confidence. [Given, SVR is susceptible to over fitting, more caution should be exercised.] However, a different model could be appropriate for a different choice of risky assets in the portfolio. The practitioner has to go through a rigorous model selection process for the portfolio in hand.

## Future Work

A few obvious items to be investigated are as follows.

1. Model selection for the mean

The model selection process used for forecasting the volatility should be followed to find the right model for forecasting rolling mean.

1. Method for selecting the rolling window size

This is an area of research. Choice of the window size can impact the volatility number and its trend

1. Daily checking of risk parameters and rebalancing when appropriate

Automatic rebalancing of portfolio is a key component of the portfolio management. A better prediction of volatility can only bring value, if the right action is taken at the right time. Automated rebalancing process fills this gap

# Appendix

A0.

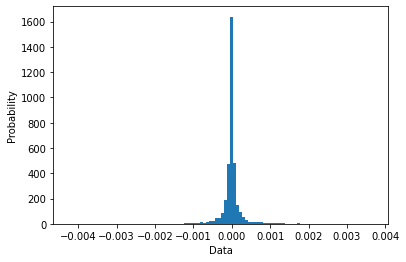


Figure 11: Error distribution for ARIMA is normal

A1.

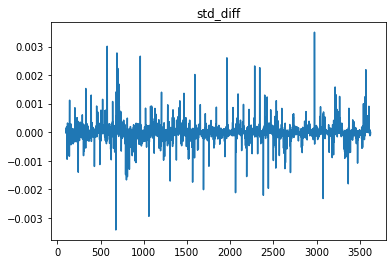


Figure 12: Stationarity check for the 10days rolling standard deviation on the percent difference time series

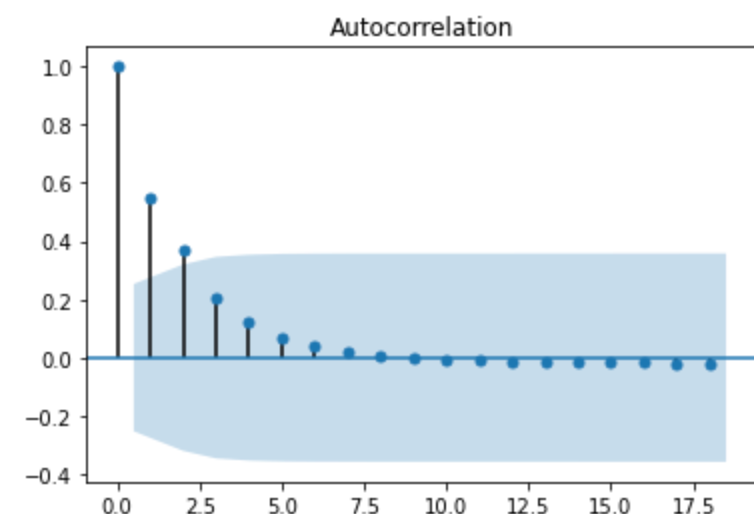
A3.

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 |

Figure 13: Stationarity demonstrated by sharply decaying Autocorrelation for the case of highlighted configuration.

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.

* Volatility Clustering
* **+Kernel**

The function used to map a lower dimensional data into a higher dimensional data.

* **\*\*Hyper Plane**

In SVM this is basically the separation line between the data classes. Although in SVR it is a line that predicts the continuous value or target value

* **Boundary line**

In SVM there are two lines other than Hyper Plane which creates a margin . The support vectors can be on the Boundary lines or outside it. This boundary line separates the two classes. In SVR the concept is same.

* **Support vectors**

This are the data points which are closest to the boundary. The distance of the points is minimum or least.

References

1. Support Vector Regression
   * 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.
2. Mean Variance Portfolio Optimization
   * <https://www.math.ust.hk/~maykwok/courses/ma362/07F/markowitz_JF.pdf>
   * <https://sites.math.washington.edu/~burke/crs/408/fin-proj/mark1.pdf>
3. ARIMA/GARCH
   * <https://people.duke.edu/~rnau/411arim.htm>
   * <http://public.econ.duke.edu/~boller/Published_Papers/joe_86>
4. Rolling Window Size
   * <https://www.sciencedirect.com/science/article/pii/S0167739X19309203>
5. Montecarlo Simulation
   * <https://cse.sc.edu/~terejanu/files/tutorialMC.pdf>
6. Stationarity
   * <https://people.duke.edu/~rnau/411diff.htm>
7. AIC/BIC
   * <https://books.google.ca/books?id=fT1Iu-h6E-oC&printsec=frontcover#v=onepage&q&f=false>
8. Tree based Regression/Gradient Boosting
   * <https://www.dcc.fc.up.pt/~ltorgo/PhD/th3.pdf>
   * <https://blog.dataiku.com/tree-based-models-how-they-work-in-plain-english>
9. GAM
   * https://en.wikipedia.org/wiki/Generalized\_additive\_model