REALIZED VOLATITLITY PREDICTION

Correlation with S&P500 index

Abstract

Volatility prediction using a correlation factor to correct the error in Exponential Weighted Moving Average.

Optiver Team 3

Table of Contents

Realised Volatility Prediction

Goal: A model for predicting short term volatility for hundreds of stocks in different sectors.

Solution:

The target volatility can be written as a sum of estimated volatility and an error factor. Error factor can be modelled as a product of correlation factor (β) and the change in volatility of S&P500 index. In this report we will observe the success rate of β and the short-comings of the model.

Addressed Problem: We sought to develop a lower sophistication but low-cost, explainable approach to modelling volatility, in the meantime also investigating the correlation of volatility between stocks and the market. Achieving this outcome can lead to faster adaptability in novel market events and provide Traders with additional signals to inform their trading and risk management.

Model Equation and specification

$$\sigma_{target} = \sigma_{estimated} + \emptyset$$

$$\emptyset = error$$

$$\sigma = standard\ deviation/volatility$$

$$\emptyset = \beta * abs(\sigma_{SPY,curr} - \sigma_{SPY,prev})$$

$$\beta = correlation\ b/w\ stock\ and\ S\&P500\ index$$

$$\beta = \frac{Covariance(stock, S\&P500)}{\sigma_{stock} * \sigma_{SPY}}$$

As the estimated volatility is always greater than the target volatility, we use the correlation (β) term to reduce the error.

$$if\ \beta > 0: (positively\ correlated\ stocks)$$
$$\sigma_{target} = \sigma_{estimated} - \emptyset$$

$$if\ \beta < 0: (negatively\ correlated\ stocks)$$
$$\sigma_{target} = \sigma_{estimated} + \emptyset$$

The main goal is a two way optimization, i.e. finding a better estimator for volatility and reducing the error term in our estimation (using the beta correlation).

Initially, we were using the naïve model for estimating the volatility. To make it even more accurate, we deployed Exponentially Weighted Moving Average (EWMA).

Global Beta

Calculation of Beta:

Initial calculation of volatility (σ) in naïve model is done as the square root of sum of log returns for each time period. Our model optimised this further to use EWMA.

Here we first normalise the time series of the S&P500 index and the stock we are calculating the covariance off.

$$\beta = \frac{\sum(x_i - x^1)(y_i - y^1)}{\sigma_{stock} * \sigma_{SPY}}$$
$$where,$$
$$x^1 = mean\ of\ x$$
$$y^1 = mean\ of\ y$$

Calculation of Estimated Volatility using EWMA:

Volatility for time period 't' can be modelled as

$$\sigma^2_{estimated,t} = \alpha\sigma^2_{estimated,t-1} + (1-\alpha)R_{t-1}$$

$$Where,$$

$$R_{t-1} = log\ returns\ for\ prev\ time\ period$$

$$\alpha = \frac{2}{n+1} = weights$$

As we are using a 10 sec. window, n = 10 therefore, α = 0.182

Global Beta:

Here the correlation factor (β) is taken as the global value, i.e. it is calculated for the entire time series given, and not just the time period we are estimating the volatility for, this is done so that Beta is a more contant value and we want to cover the major trend for the time series.

Further in this report we are using the term global beta, which is calculated for the entire data set i.e., all time periods together. This value is used for all calculations of volatility.

Local Beta:

Beta is recalculated per time period for the entire data set.

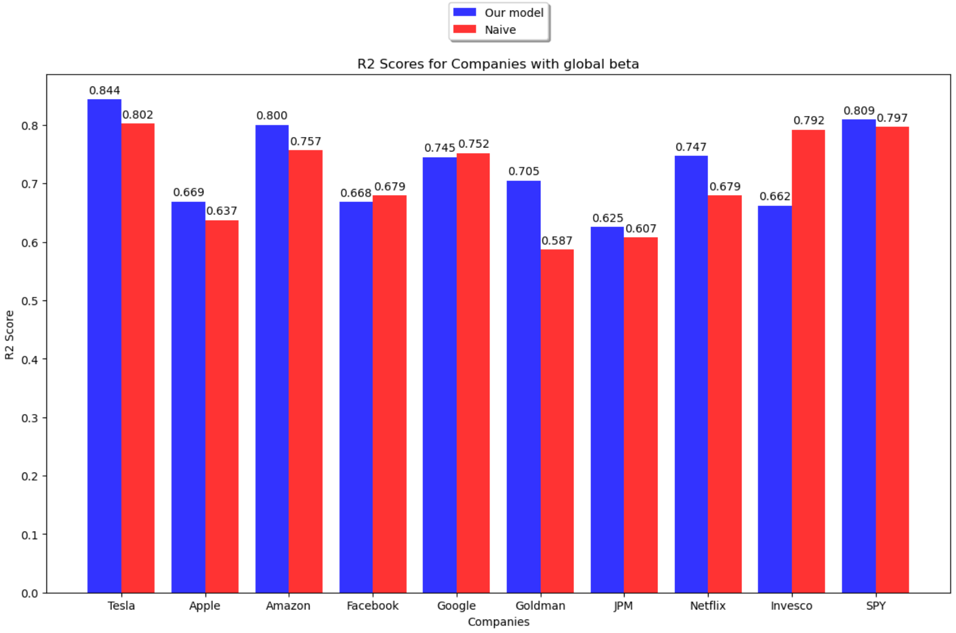
The difference is one considers entirety of past data and the other is completely based on the data of that time period. This was done to test the feasibility of this solution and estimate the power of correlation in volatility calculation without data to model on.

Success Metrics

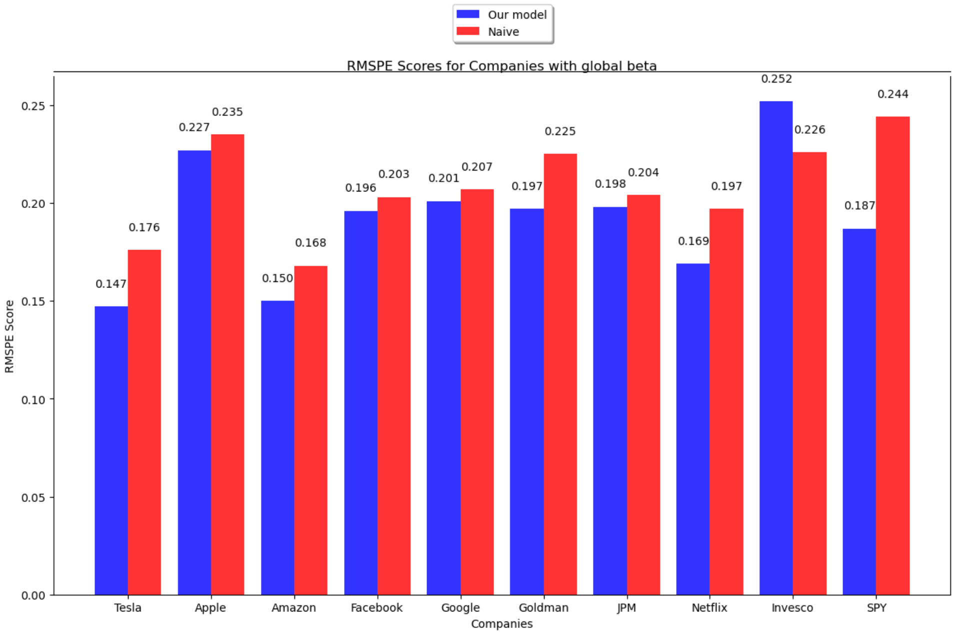
Accuracy Comparison b/w Naïve and Our Model						
Stocks	**Naïve Model**		**Global beta**		**Local Beta**	
	R2	**RMSPE**	**R2**	**RMSPE**	**R2**	**RMSPE**
TSLA XNAS	0.802	0.176	0.844	0.147	0.843	0.147
AAPL XNAS	0.637	0.235	0.669	0.227	0.653	0.229
AMZN XNAS	0.757	0.168	0.800	0.150	0.773	0.156
FACEBOOK XNAS	0.679	0.203	0.668	0.196	0.693	0.187
GOOGC XNAS	0.752	0..207	0.745	0.201	0.757	0.196
GS XNAS	0.587	0.225	0.705	0.197	0.701	0.196
JPM XNAS	0.607	0.204	0.625	0.198	0.600	0.200

NFLX XNAS	0.679	0.197	0.747	0.169	0.732	0.173
QQQ XNAS	0.792	0.226	0.662	0.252	0.662	0.252
Overall	0.845	0.21	0.864	0.198	0.864	0.198

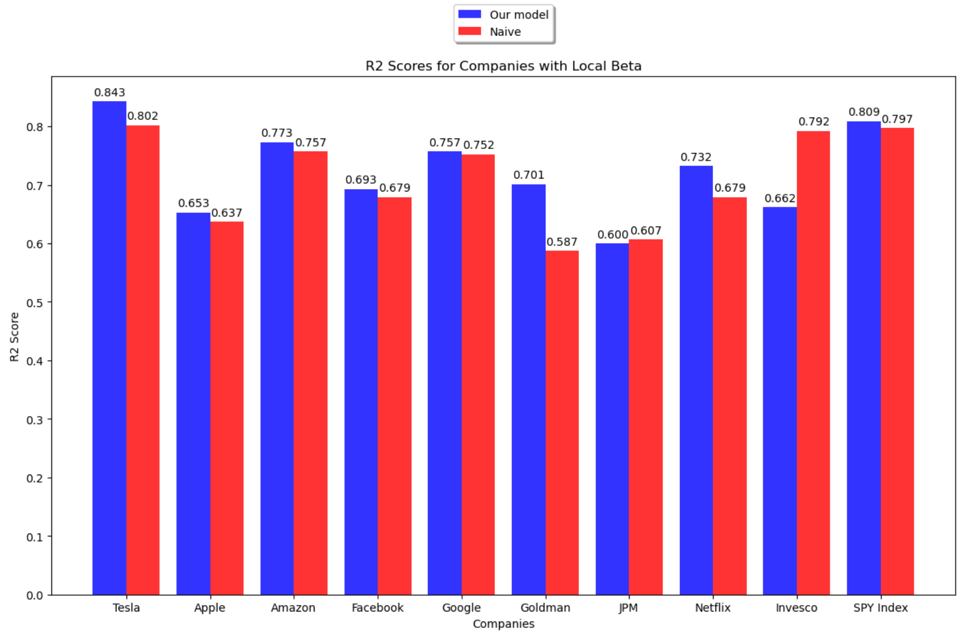
R2 metric graph of Global Beta Model:



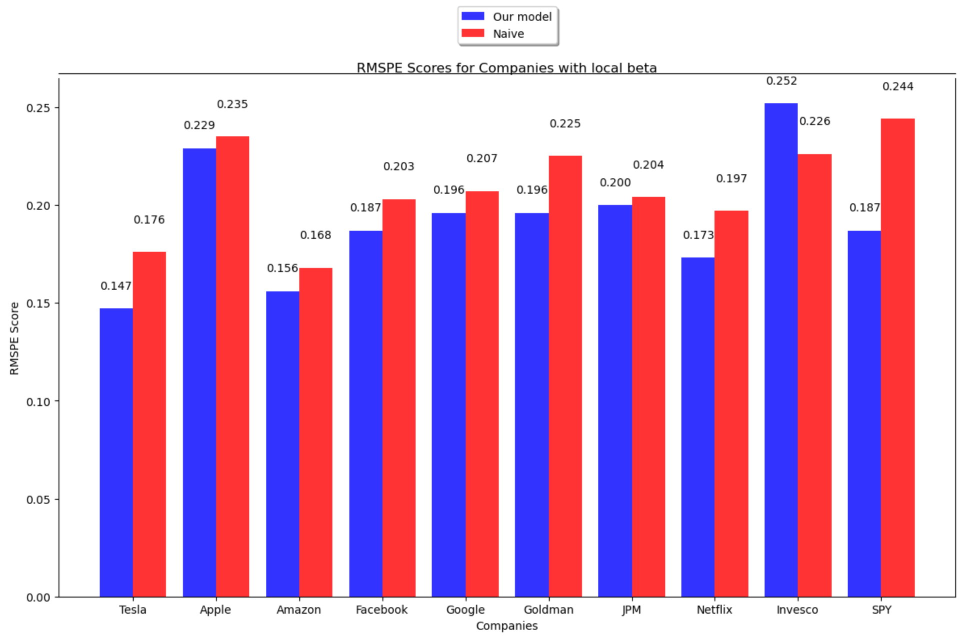
RMSPE graph for Global Beta Model:



R2 metric graph of Local Beta Model:



RMSPE graph for Local Beta Model:



Local beta works best for all stocks, achieving greater performance than naïve model in all scenarios. It is also truly data independent, and its runtime is around 1 sec per stock per time period.

Invesco failure:

The lack of performance is attributed to the following reasons:

1. Sudden large changes in value of asset:

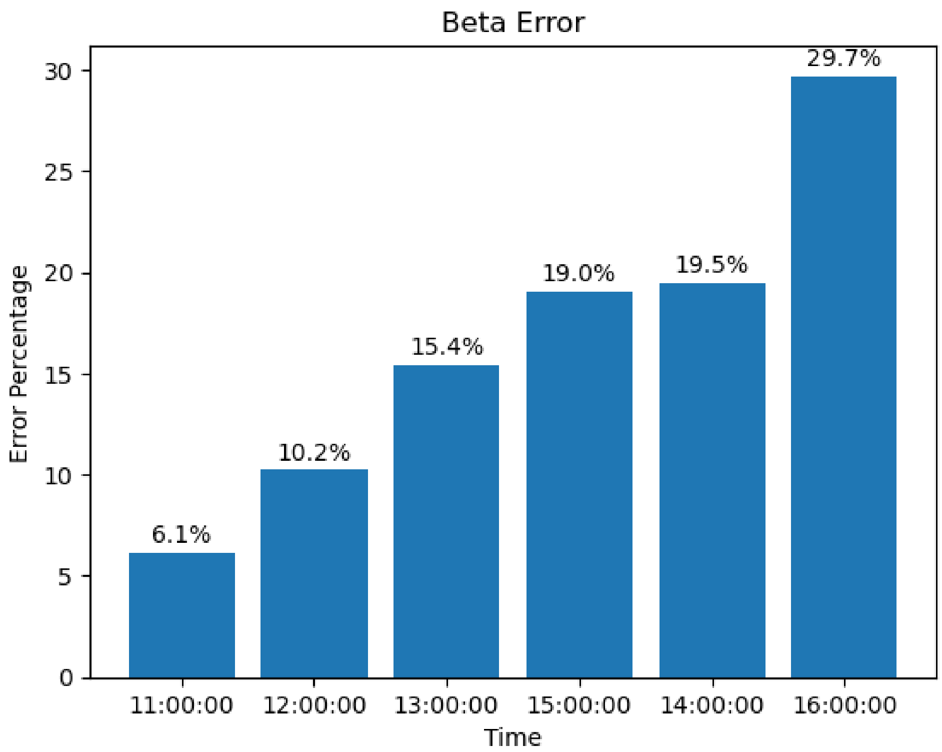
EWMA reacts more slowly to sudden changes as it weights out the previous data, which naïve model takes an equal weighted sum, so reacts well to sudden changes.

2. Higher degree of randomness or noise in the data
   Data smoothing can increase the accuracy of EWMA. It might predict better if the sudden changes are viewed in a smaller capacity on a larger time frame.
3. Invesco (QQQ) is not a part of the S&P500 index so its correlation to this index doesn't make mathematical sense. This might attribute to reduction in accuracy in estimation of volatility.

Beta Error Details

Below is the graph that covers the beta directional errors i.e. the frequency of beta nudging us in the wrong direction for each time period.

Here we see that the period of 16:00 hrs everyday has massive inaccuracies, adding a flag to invert it based on certain underlying conditions will greatly increase the accuracy.



Future improvements

In order to enhance the accuracy of the volatility prediction models, we could explore and incorporate more advanced techniques and methodologies. This includes:

- Data-Driven EWMA model: Data-Driven Exponential weighted moving average model can help improve accuracy as it obtains more optimal alpha values based on historical data and helps us better capture the sudden changes in volatility.

- Neural networks and Machine learning: We can use neural networks and machine learning to refine our volatility model. This could be done by using decision trees for decision making, utilising a one-dimensional convolutional neural network (1D CNNs) for time series analysis, and natural language processing (NLP) techniques for analysing news articles as additional inputs for volatility prediction.
- Ensemble methods: Ensemble methods enable us to make our model more robust and improve our accuracy. In the ensemble model, we could try combining the strengths of the above methods for us to achieve better performance.
- Comparison over other stock indexes: By comparing with other stock indexes like QQQ, instead of just the SPY index, we could gain a better grasp as to the correlation between the log returns and the current market returns.

Technical details of Python Notebook

Function Name	**Utility**
gen_vol_calc	Computes the volatility using log returns
vol_calc	Weighted Average Volatility by splitting time period into 10 mins and 20 mins.
bestBeta	Computes the Beta of the stock with other 9 stocks and returns the most correlated stock
betaCalc	Computes the Beta of the stock for entire time series wrt S&P500 index. Returns Beta
stockIntersect	Function used within betaCalc to normalise the time series of stock and S&P500 index.
ewma	Returns the EWMA volatility calculation for the required time period.

Libraries used:

- NumPy
- Pandas
- skLearn (SciKit Learn)
- matplotlib
- date time

References

Beta Correlation: https://www.investopedia.com/terms/c/correlation.asp

EWMA: https://www.investopedia.com/articles/07/ewma.asp

EWMA: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.ewm.html