Volatility Prediction Using Machine Learning

Digvijaysinh Gohil* (AU1940199), Tirth Bharatbhai Kanani* (AU1920144), Smit Shah* (AU1940291), Satya Shah* (AU1940288)
School of Engineering and Applied Science, Ahmedabad University
***All Authors have contributed equally**

Abstract—**Volatility is one of the most prominent terms one can hear on any trading front and for good reasons. In financial markets volatility captures the amount of fluctuation in prices. High volatility is associated with periods of market turbulence and to large price swings, while low volatility describes a more stable market. For trading firms to accurately predict volatility is essential for the trading of options, whose price is directly related to the volatility of the underlying product. In the following project we will make use of the data published on Kaggle and we will do EDA on the data to better understand the features and then try to predict the volatility using various Regression models such as Linear, Logistic, Ridge, and Gradient Boosting Model. We will further fine tune the features of the data using various feature engineering techniques to get better accuracy.**

Index Terms—**Machine Learning, Stock Market, Volatility, Regression, Polynomial Regression, Data Analytic, Data Reprocessing, ARIMA Model, LaTeX.**

I. Introduction

Volatility is the backbone of finance because it serves as both an information signal for investors and an input to various financial models. What is the significance of volatility? The response emphasizes the significance of uncertainty, which is a key feature of the financial model. Increased financial market integration has resulted in protracted market uncertainty, emphasizing the relevance of volatility, or the rate at which the value of financial assets changes. Volatility, which is utilized as a proxy for risk in many domains, including asset pricing and risk management, is one of the most significant variables. Modeling is even required because of its substantial presence and delay. Following the Basel Accord, which went into effect in 1996, volatility has become a major risk measure in risk management (Karasan and Gaygisiz 2020). Modeling volatility is the same as modeling uncertainty, and it allows us to better comprehend and approach uncertainty, allowing us to get close enough to the real world. We need to calculate the return volatility, also known as realized volatility, to see how well proposed models account for the real-world situation. The square root of the realized variance, which is the total of squared returns, is realized volatility. The performance of the volatility prediction approach is calculated using realized volatility.

The goal of the project is to forecast short-term volatility for 112 stocks from various industries. The data set includes stock book and trade data for several periods. We're meant to forecast a target value (volatility) for each stock class. In the training data set, 107 stocks have 3830-time buckets, 3 stocks have 3829-time buckets, 1 stock has 3820-time buckets, and 1 stock has 3815-time buckets. As a result, there are 428,932 rows to anticipate. The training set has three columns, whereas the placeholder test set has two.

- stock id - Stock ID
- time id - Time Bucket
- Target - Actual volatility of the following 10 minute window for the same stock id/time id

To approach this particular problem, we have used two model so far:

- Linear Regression
- Polynomial Regression
- Light GBM

II. Literature Review

Increased financial market integration has resulted in sustained market uncertainty, reinforcing the significance of volatility, or the rate at which the value of financial assets changes. Thus, volatility has a major dependency on stock trading risks. One of the first papers on building the relationship between market volatility and returns in the assets was given by Black in 1976. Further efforts in forecasting future volatility led to the development of several statistical models such as Autoregressive Conditional Heteroskedasticity (ARCH) and which was first put forward by Eagle in 1982. This approach does not make use of sample standard deviation but formulates conditional variance of returns via maximum likelihood procedures.

The evolved version of GBDT (Gradient Boosting Decision Tree) is LightGBM. It has good accuracy and has an RMSPE error of 0.211 which is the least among several algorithms like Logistic regression, XGboost, and SVM. This model was proposed by Yue Wu and Qi Wang of China (IEEE International Conference on CSAIEE, 2021). GBDT is a very prominent approach for volatility prediction. It is widely used in Industries. The problem with GBDT is that it needs to pass through the data multiple times. Also, it needs to load the entire data into memory multiple times and so it is very time-consuming. The widely known tool for GBDT was XGBoost. There were disadvantages. Space consumption was large. There was a larger overhead in time in this approach. So the traditional GBDT algorithm is not the solution at an industrial level where the data is very large. So to use GBDT for better and faster results at the industrial level, LightGBM is used.

Additionally, the historical average method on the past data is also used to predict volatility but it is rather more static. On the other hand, Moving Average, Exponential Smoothing Methods weighs more on the recent volatility values. The RiskMetrics model uses the EWMA (Exponentially Weighted Moving Average). The Smooth Transition Exponential Smoothing Model which was proposed by James Taylor(2001) was a very flexible approach of exponential smoothing where the weights depend on size and signs. Different Stochastic Volatility approaches have been made in past such as Quasi maximum Likelihood Estimation(QLME) and Generalized Method of Moments(GMM).

III. Implementation

A. Linear Regression

Regression machine learning models, look at continuous variables to see how they relate to a target variable. Because volatility may be expressed as a range of values rather than discrete categories, regression proved the best choice for modeling the relationship between variables across time. Because volatility is merely a measure of change, there are an endless number of ways that may be used to measure volatility in data presented as a time series in theory. However, because volatility is a measure of change, it must be placed in the context of a base value or pattern. The present volatility of the stock market can be quantified by comparing today's movement to last week's movement, last week's movement to this year, or even this year's movement to the previous 10 years. Because there is no consensus on what ”volatility in the stock market” even entails, quantifying volatility is nearly subjective.

Using daily percent change as a measure of volatility produced faultily and, in some cases, utterly incorrect results. This was because daily percent change was, ironically, extremely variable. Though the data showed spikes for particularly volatile days, the next day's value was classified as non-volatile if there was no variation. Volatility should have been examined from a more nuanced perspective, characterized as existing or not existing over longer periods rather than only in daily surges. Another problem with using daily percent change as a measure of volatility was that it ignored market patterns. Such an incidence was documented as a case of volatility daily.

The only volatility measurement that offered the trend-sensitive, real-time and consistent data needed for this research was modified to match the indicator data.

B. Polynomial Regression

It is a common method in financial mathematics to use continuous-time models like the Black Scholes model to approximate financial markets that operate in discrete time. Due to the discontinuous structure of market data, fitting this model is difficult. Thus, we use the Black Scholes equation to represent the pricing of financial derivatives, where volatility is a function of a finite number of random variables. This illustrates the impact of uncertain factors on volatility determination. The goal is to measure the impact of this uncertainty when calculating derivatives prices. Our underlying method is the generalized Polynomial Chaos (gPC) method, which uses a stochastic Galerkin approach and a finite difference method to numerically compute the solution's uncertainty.

Now as we know polynomial regression performs better on non-linear data and since we had too many parameters we saw that the linear regression performed poorly. In particular degree 3 polynomial seemed to be a better fit for the data and hence after some tuning of parameters we were able to achieve better accuracy than the regression model.

C. Light GBM

LightGBM is a decision tree-based gradient boosting framework that improves model efficiency while reducing memory utilization. It employs two innovative techniques: Gradient-based One Side Sampling and Exclusive Feature Bundling (EFB), which address the drawbacks of the histogram-based approach used in most GBDT (Gradient Boosting Decision Tree) frameworks. The properties of the LightGBM Algorithm are formed by the two methodologies of GOSS and EFB explained below. They work together to make the model run smoothly and give it an advantage over competing for GBDT frameworks.

LightGBM does not grow a tree row by row, unlike most other implementations. Instead, it grows trees leaf-by-leaf. It selects the leaf that it believes will result in the most significant loss reduction. Furthermore, unlike XGBoost and other implementations, LightGBM does not use the sorted-based decision tree learning algorithm, which searches for the optimum split point on sorted feature values. Instead, LightGBM uses a highly optimized histogram-based decision tree learning algorithm, which provides significant performance and memory savings. Gradient-Based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) are two unique techniques used in the LightGBM algorithm to run faster while maintaining excellent accuracy.

D. Experimental Data

The data we have been working has been provided to us by Optiver a firm based in Netherlands which deals with huge number of clients on day to day basis. They deal with ETFs, stocks and options and are committed to continuously improve the financial markets. The data contains relevant and original trades happening in the market everyday. It also includes the passive information happening in market as an order book data. We have book.parquet files partitioned by stock-id. The parameters and their meaning are shown in the table below:

stock_id	ID code for the stock.
time_id	ID code for the time bucket.
seconds_in_bucket	Number of seconds from the start of the bucket, always starting from 0.
bid_price[1/2]	Normalized prices of the most/second most competitive buy level.
ask_price[1/2]	Normalized prices of the most/second most competitive sell level.
bid_size[1/2]	The number of shares on the most/second most competitive buy level.
ask_size[1/2]	The number of shares on the most/second most competitive sell level.

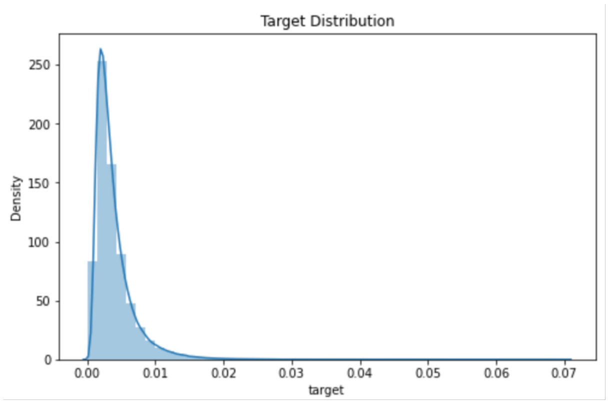
The trade_[train/test] parquet is also partitioned by stock id. Contains information about trades that were really completed. Because there are more passive buy/sell intention updates (book updates) in the market than actual trades, this file should be sparser than the order book. It also has the remaining three fields.

price	The average price of executed transactions happening in one second. Prices have been normalized and the average has been weighted by the number of shares traded in each transaction.
size	The sum number of shares traded.
order_count	The number of unique trade orders taking place.

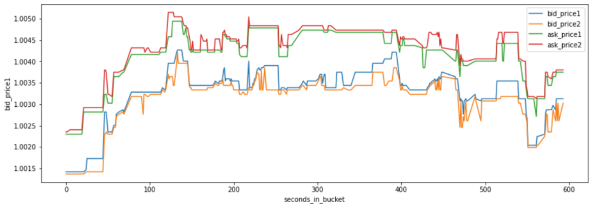
IV. Results

We ran a total of two models, one is linear regression and the other is polynomial regression. While it is pretty evident that polynomial regression performed better it is interesting to look at how linear regression failed to work even after we specifically tuned the parameters for its model. The regression model worked well on the training data but when we tested it on the unseen data it failed to generalize well. On the other hand, for polynomial regression, we had to approach it with a trial and error problem. For degrees 1 and 2 the model wasn't much better than simple linear regression. For degree 3 it performed the best out of all the models we tried up til now. After that increasing the degrees of polynomial the accuracy started decreasing slowly. Final accuracy for polynomial regression of degree 3 we got was 12.199 which is much better than the linear regression and hence with more feature engineering and parameter tuning we can increase the accuracy of the polynomial regression model.

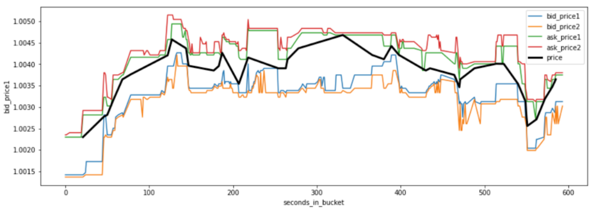
The first step was to understand the data, since there were millions of rows and too many parameters spread across CSV and Parquet files, it was necessary that we get a clear understanding of the data. We began with Explanatory data analysis by plotting graphs of different columns in python. We got the following distribution graph of the target volatility shown below:



As you can see the target volatility is skewed to the left with the mean being 0.003. Further, we tried to see the relationship between different ask and bid prices upon the share available in the market:



We then compared this graph to the actual trades taking place in the trade-book data given to us; Parameter price is added to the following graph:



After the data exploration, it was quite clear to us that the data needs to be preprocessed. Hence we began to look for supporting research articles and papers. We processed the data given to us by calculating the Weighted average price and realized volatility. The formula of the same has been mentioned below:

Weighted Average Price

$$WAP = \frac{BidPrice_1 \times AskSize_1 + AskPrice_1 \times BidSize_1}{BidSize_1 + AskSize_1} \tag{1}$$

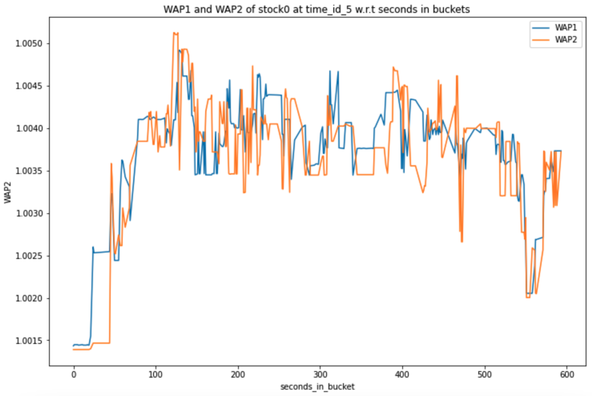
Realized Volatility

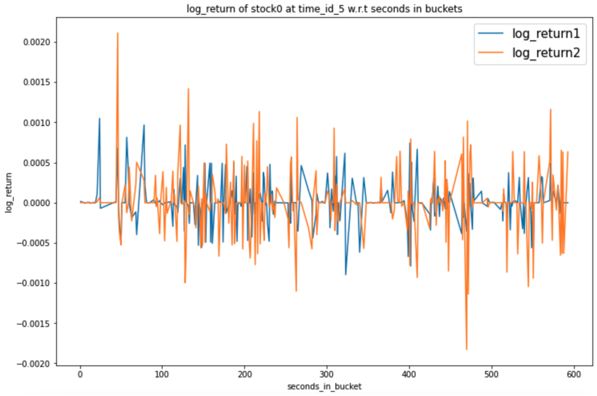
$$\sigma = \sqrt{\sum_t r^2_{t-1,t}} \tag{2}$$

After calculating this in the data set the accuracy of the model started to increase, we first tried linear regression with data preprocessing and the accuracy did increase but it was not exponential. The model did perform better but it still wasn't up to the mark. Hence we decided to move forward with the Polynomial regression. Since the data was non-linear it made more sense to use polynomial regression and for degree fitting, we used the trial and error method. Through this, we were able to achieve an accuracy of 0.24 RMSPE value. The evaluation criteria are decided by the Kaggle and hence we chose to follow them.

$$RMSPE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}((y_i - \hat{y_i})/y_i)^2} \tag{3}$$

The following is the formula for the evaluation criteria.





Here, as shown above, the WAP is the function of stock price, and hence they can vary a lot, as seen in graph 1 (WAP). It could be challenging to compare two stocks whose prices are not comparable. Let's suppose there is stock A, whose fee is 100, and stock B, whose price is ten, and if there are, say, increases in expenditure on both the stocks, by default the, stock A would have a higher mean and effect on the WAP then the store where in return the return on the stock B is higher than the stock A. Hence we take returns in calculation and then take the logarithm function to help us computational speed. The same is shown in graph 2 (Log return).

LightGBM	0.24
Polynomial Regression	12.19
Linear Regression	25.09

V. Conclusion

Realized Volatility is the representation of the changes happening in the stock markets over a given period of time, market volatility and risks associated with it. In this article we have used the data sets provided to us by the Optiver a foreign exchange firm dealing with huge amounts of clients and data. Section 3 provides the information of the two models we built and section 4 explains the results obtained through those models. We conclude with some related work to the target i.e. realized volatility. We obtained the lowest RMSPE value of 12.199 through our polynomial regression model which significantly better than linear regression. The accuracy achieved in LightGBM Model is 0.24.

References

1) Y. Wu and Q. Wang, ”LightGBM Based Optiver Realized Volatility Prediction,” 2021 IEEE International Conference on Computer Science, Artificial Intelligence and Electronic Engineering (CSAIEE), 2021, pp. 227-230, doi: 10.1109/CSAIEE54046.2021.9543438.

2) S. M. Idrees, M. A. Alam and P. Agarwal, ”A Prediction Approach for Stock Market Volatility Based on Time Series Data,” in IEEE Access, vol. 7, pp. 17287-17298, 2019, DOI: 10.1109/ACCESS.2019.2895252.

3) Kaggle.com. 2022. Introduction to financial concepts and data. [online] Available at: https://www.kaggle.com/code/jiashenliu/introduction-to-financial-concepts-and-data [Accessed 20 March 2022].

4) A machine learning approach to volatility ... - pure.au.dk. (n.d.). Retrieved March 19, 2022, from https://pure.au.dk/portal/files/208284743/rp2103.pdf

5) “Optiver realized volatility prediction,” Kaggle. [Online]. Available: https://www.kaggle.com/c/optiver-realized-volatility-prediction. [Accessed: 20-Mar-2022].