UNSW SYDNEY

OPTIVER

TECHNICAL REPORT

PREPARED BY TEAM 2
Axel Librata
Barry Ye
Marvel Nelwan
William Moog
Wyun Ng

2023 | APRIL

CDEV3000

optiver

Table of Contents

Executive Summary

In this report, we develop an approach to investigate implementation and evaluation of a hybrid model within a realized volatility prediction context. Our approach builds on feature engineering to the commonly used Exponentially Weighted Moving Averages Model and the Light Gradient-Boosting Machine Model, herein referred to as the EWMA and LightGBM Models respectively. This report **evaluates the performance of a hybrid model that uses Bayesian averaging to combine EWMA and LightGBM** as base estimators for predicting stock volatility. The choice of model parametrization is critical to the model's strengths and weaknesses. The results show that the **hybrid model performs well in most cases**, with higher accuracy scores and low risk of overfitting, but struggles to accurately predict volatility in certain market conditions. The consolidated hybrid model outperforms the Naïve model as a benchmark, with a **runtime of 16 seconds, R^2 score of 0.878, and RMSPE of 0.196**. The model can be an effective tool for predicting stock volatility, and we have suggested several tentative areas of improvements that relate to further feature engineering.

Introduction

As a market maker, Optiver would be interested in how to effectively provide liquidity to global markets. Hence, our objective was to **address the need for a more accurate and efficient approach** to predicting short-term stock volatility, which is a critical component of effective quantitative trading strategies. The key output of the hybrid model are predictions of the realized volatility that are evaluated through Bayesian Averaging. We apply Ensemble Learning to a widely used Bayesian method with the stock dataset as the key input to demonstrate its predictive capacity. The information provided by our strategy could provide value to you through the following:

1) **Diagnose whether some model parameters** have impact on the underlying predictive powers of the Hybrid Model.
2) Provide information to **reformulate the model structure** and improve its performance.
3) Evaluate its **reliability, predictive capacity, and accuracy** in predicting volatility.

The Hybrid Model Strategy Journey

We have consolidated the development of this project and the added improvements to the model with the feedback given to us by Optiver and other external parties through the following outlines:

Client Meeting 2

Our initial strategy was to use GARCH and ARIMA models which are sophisticated machine learning models used for predicting volatility.

Adrian and Virginia's Feedback:

- Δ Easy-to-understand results or **model interpretability** are more valuable than complex yet accurate results.
- Δ Virginia **validated our hybrid model approach** with her ensemble suggestion.

Client Meeting 3

We developed the models but the runtime for each model was ~184 minutes which is too long for the predictive window of 30 minutes.

Greg and Virginia's Feedback:

- Δ Recommendation to pursue an **EWMA model**, which is faster, simpler, and similar in accuracy to the ARIMA.
- Δ We were recommended to **explore other potential evaluation metrics** instead of just the R2 and RMSPE value.

Client Meeting 4

We presented a hybrid model with EWMA and random forest as base estimators. We also evaluated the models with the root mean squared error (RMSE) and the mean absolute error (MAE) as alternative metrics.

Adrian and Virginia's Feedback:

- Δ Recommendation to **visualize results and metrics better**.
- Δ **Mitigate the risk of heteroskedasticity** and what conditions would the model work.

External Consultant

We consulted with an external consultant, a mathematics and statistics PhD student at UNSW, on the model parameters and mathematical justifications.
Brock Sherlock's Feedback:

- Δ Replace Random Forest with the **LightGBM model**.
- Δ **Revised the Bayesian averaging method** and its limitations. The model previously had an R2 of 0.944 which after revising the hybrid model was tuned to 0.878 with better RMSPE scores.

After numerous stakeholder feedback given to us, we have iterated and developed the completed model. The hybrid model uses EWMA and LightGBM models as base estimators, since they both have **a small runtime and high accuracy**. We use the Bayesian averaging technique highlighted by Brock and Virginia to combine the results of these two models for a more accurate prediction.

Base Estimator: Exponentially Weighted Moving Averages

The EWMA model is a simple and efficient method for predicting stock market volatility. The term "exponentially" refers to how the weight given to each observation diminishes exponentially as it moves further into the past. It accumulates current information as well as previous information and detects any variance in the process. It is also easy to understand and interpret. Unlike other models, it does not rely on specific assumptions about the distribution of the data, which makes it **well-suited for analyzing financial data that may not follow a normal distribution**. Additionally, because it **requires minimal computational resources**, it can be processed quickly, making it a valuable tool for traders and investors who need to make decisions quickly.

EWMA Methodology

The code in the file attached calculates the volatility of financial market data using the EWMA model.

```
def calc_volatility(group):
    if 'log_return' not in group.columns:
        return np.nan
    ewm_vol = group['log_return'].ewm(span=10).std() * np.sqrt(len(group))
    return ewm_vol.mean()
volatility = book.groupby(['time_id', 'stock_id']). apply(calc_volatility).reset_index(name='prediction_ewma')
train = train.merge(volatility, on=['time_id', 'stock_id'], how='left')
```

The code reads data on trading activity and order book information, calculates the log returns of the weighted average price (WAP) for each stock, and then applies the EWMA method to the log returns using a span of 10 secs. Following the EWMA calculation, we used the **annualized volatility formula**, which involves multiplying the standard deviation of the EWMA by the square root of the length of the data, to

obtain the final volatility estimate. Then the mean volatility estimate is updated based on the new observations in the time series which follows the formula:

New Mean Estimate = alpha * New Observation + (1 - alpha) * Old Mean Estimate

The updated estimate is then used to predict the next observation in the time series. The process is then **repeated for each new observation in the time series**. The performance of the model is then evaluated using various metrics, such as R2 score, RMSPE, MAE, and RMSE, to determine how accurately it predicts the actual volatility of the market data.

Parameter Decisions

The EWMA model has a primary parameter which serves as a smoothing factor and is depicted as alpha (α). This smoothing factor determines the weight assigned to observations in the dataset from a range of 0 to 1 in which **a larger alpha value places more weight on recent observations** and less on older observations. Moreover, the optimum alpha value to be chosen for the EWMA depends on the dataset and the context and goal of the analysis.

$\alpha = 2/ (N+1)$
$\mathbf{0.1818} = 2/ (10+1)$

You can derive the alpha value from the equation above, where N is the time window chosen for the EWMA model. This equation is derived from the exponential smoothing method, which is a **common technique used in time-series analysis**. Moreover, the equation aligns with the aforementioned rules where the alpha value increases as N (time period) decreases, hence putting more weight to recent observations.

In consequence, our team had undergone trial and error to find an alpha value that yields the best R-squared and RMSPE values. Overall, we have chosen **a time window of 10 seconds**, which corresponds to an alpha value of 0.1818 as it yields the best performance results. This value allows the model to emphasize recent observations, hence accounting for more variability which mitigates the risk of heteroskedasticity.

Performance Evaluation Benchmark

	EWMA Model Performance	Naïve Model Performance
R^2 Score	0.872	0.845
RMSPE	0.188	0.21
MAE	0.00059	0.00066
RMSE	0.0009	-

Overall, the performance of the EWMA model validates its predictive short-term stock volatility capacity based on relatively higher R^2 score with an 8 second runtime and lower RMSPE, MAE, and RMSE Values.

Model Component: Light Gradient Boosting Machine

The LightGBM model is a decision tree-based machine learning model optimized for speed, accuracy, and memory conservation. Decision tree-based evaluation models like random forest, LightGBM and XGBoost, all differ in how they create their decision tree hierarchies. The power of the Light GBM model is that tree structures do not grow row by row (increasing tree depth), instead they grow leaf-wise. It does this by choosing the leaf it believes will yield the largest decrease in loss. This leaf-wise growth structure means that in large datasets, the Light GBM model is **more efficient in runtime and memory** compared to other tree structures. Hence, making it a great fit for short term stock volatility predictions.

Light GBM Methodology

The data is first sorted by stock and prepared by calculating the squared log return sum, squared log return count and the square root sum of the log returns squared.

```
book_stock = book[book['stock_id'] == stock_id]
book_stock['log_return_squared'] = book_stock['log_return']**2
book_stock_agg = book_stock.groupby(['time_id']).agg({'log_return_squared': ['sum', 'count']}).reset_index()
book_stock_agg.columns = ['time_id', 'log_return_squared_sum', 'log_return_count']
book_stock_agg['realized_volatility'] = np.sqrt(book_stock_agg['log_return_squared_sum'])
```

The squared log return sum of the WAP (weighted average price), measures the dispersion of returns which is vital to predicting stock volatility. Thus, the variability of the stock's price within the given time. Additionally, the squared log return count within each time period suggests a proportional relationship with trading activity or stock liquidity, where a higher count leads to higher trading activity. Finally, the square root sum of the log returns squared corresponds to the realized stock volatility. This value will act as the target value for the Light GBM model's testing and training phase, while the squared log return sum and squared log return count will act as the input variables.

```
X = book_stock_agg[['log_return_squared_sum', 'log_return_count']].values
y = book_stock_agg['realized_volatility'].values
```

To train and test the model, the data was split into 80% training and 20% testing data sets. 20 boosting rounds and early stopping after 5 rounds was used if no changes were made which optimizes for speed and accuracy. The model was then evaluated using the operations below.

Parameter Decisions

Below lie the parameters used in the Light GBM model.

```
params = {'boosting_type': 'gbdt', 'objective': 'regression', 'metric': 'rmse', 'num_leaves': 31, 'learning_rate': 0.05,
'feature_fraction': 0.9, 'bagging_fraction': 0.8, 'bagging_freq': 5, 'verbose': -1, 'force_col_wise': True
```

- Δ **Boosting type** – Boosting algorithm is the gbdt (gradient boosting decision tree), fastest method.
- Δ **Objective** – Objective as a regression to minimize RMSE as the metric, suited for accuracy.
- Δ **Num_leaves** - Maximum number of leaves in a tree is 31, a balance of speed and accuracy.
- Δ **Learning rate** – Step size is 0.05 to update the model's weight with each iteration.
- Δ **Feature fraction** – Reduces overfitting by representing 90% of features used for boosting rounds.
- Δ **Bagging fraction** – 0.8 represents the train-test data split percentage of 80%.
- Δ **Bagging frequency** - Every 5 iterations of boosting a new sample of data will be used for training.
- Δ **Verbose** – Negates the redundant information conveyed with '-1'.
- Δ **Force_col_wise** – Sets histogram building to be true since dataset has multiple columns.

Consequently, all the parameters used for the Light GBM model were chosen to ensure a fast runtime, to minimize overfitting and to create a high predictive accuracy.

Performance Evaluation Benchmark

	Light GBM Model Performance	Naïve Model Performance
R^2 Score	0.843	0.845
RMSPE	0.21	0.21
MAE	0.00066	0.00066
RMSE	0.001	-

Overall, the Light GBM model with an 8 second runtime had similar accuracy to that of the naive model, only having a slightly lower R^2 value, however, despite that little difference, the Light GBM model is still a valuable tool for predicting short-term stock volatility.

Hybrid Model – Bayesian Averaging

Reason for Developing a Hybrid Model

Individually, both the EWMA and Light GBM models perform somewhat accurately, however, the use of the Bayesian averaging technique combined with these two models can **further improve the model's accuracy**. The Bayesian average is also extremely efficient in terms of runtime, it essentially combines each output from the two models with a simple operation. The importance of the Bayesian averaging technique lies in the fact that it can highlight certain predictions and their relative accuracy, to make a further accurate model.

Bayesian Averaging Methodology

The Bayesian averaging formula simply uses a weighted average formula.

$$\frac{C_1P_1 + C_2P_2}{C_1 + C_2}$$

Now for our data values we will be using the predictions made by the two models, indicated by the P_1 (EWMA prediction) and P_2 (Light GBM prediction) variables. The weightings used in the formula are the RMSE values for that model's prediction. The choice for using the RMSE value is because it looks at the actual errors of the model's predictions and is typically **a better measurement for comparing different model's accuracies than the R^2 and MAE values**. Furthermore, for this evaluation to work effectively the current stock being predicted must also be considered, since the accuracy of different models changes for different stocks. To do this, the model must generate the RMSE values for each model's predictions of a certain stock. On the right it shows an example table of this.

stock_id	RMSE_ewma	RMSE_lgb
9323	0.000752	0.000709
22675	0.000793	0.000878
22951	0.000812	0.000845
22729	0.001092	0.001154
48219	0.001195	0.001130
22753	0.000647	0.000604
22771	0.001080	0.001113
104919	0.000453	0.000518
50200	0.000361	0.000417
8382	0.001321	0.001618

Before finally completing the evaluation, one more factor must be considered. Mathematically with the Bayesian averaging formula the larger the coefficients value (C), the larger its weighting in the result, however, statistically speaking a smaller RMSE corresponds to a more accurate model. This means if the RMSE values were directly used as the coefficients, the less accurate model would have a larger weighting. To avoid this, the coefficient values used instead are (1 – RMSE). This essentially ensures that the smaller RMSE values will correspond to a larger coefficient, meaning that the more accurate model will have a larger weighting in the evaluation. Finally, we arrive at our values for C where the C_1 and C_2 variables are correspondingly (1 – RMSE_ewma) and (1 – RMSE_lgb) at the stock currently being predicted. Resulting in the simple operation below for our Bayesian averaging.

```
merged_df['prediction_hybrid']
= ((1 - merged_df['RMSE_ewma']) * merged_df['prediction_ewma'] + (1 - merged_df['RMSE_lgb']) * merged_df['prediction_lgb']
   / ((1 - merged_df['RMSE_ewma']) + (1 - merged_df['RMSE_lgb']))
```

Lastly the model's performance was then evaluated using different metrics for accuracy, such as the r^2, RMSPE, RMSE and MAE values.

Reliability of the Hybrid Model

It is possible that our hybrid model **performs better for short-term predictions** (e.g., 30 minutes) than for longer time horizons, as the performance of a model depends on the features and the underlying structure of the data, which may be more predictable in the short term. This hybrid model, which is effective in predicting short-term volatility, might not capture and consider any black swan events or market sentiments, nuances, and complexities of longer time horizons.

Additionally, **as the time horizon increases, the influence of noise and uncertainties in the data may grow**, reducing the model's ability to make accurate predictions. Financial markets are often more challenging to predict over longer periods due to the influence of various factors such as macroeconomic events, geopolitical situations, and other unpredictable events that can cause significant fluctuations in the market.

Performance Evaluation Methodology

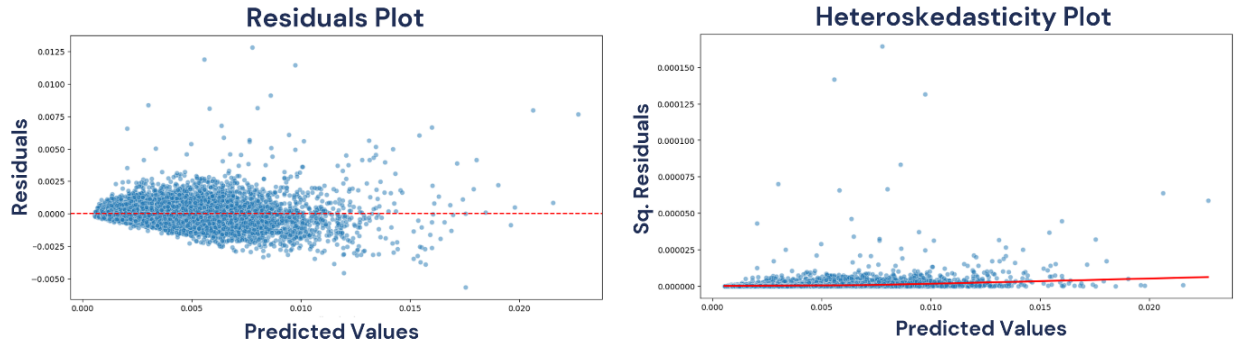
In this technical report, we have chosen **a combination of diagnostic and evaluation plots** to assess the performance of our hybrid model. These methods were chosen for their ability to effectively analyze the model's accuracy, identify potential biases, and provide insights into the model's strengths and weaknesses.

- **Residual Plot**:
  Residual plots are employed to diagnose our hybrid models by displaying the differences between actual and predicted values. A well-fitted model should exhibit residuals that are randomly scattered around zero with no obvious patterns. The use of residual plots allows us to **determine whether the model has effectively captured the underlying dynamics of the data**.
- **Heteroskedasticity Plot**:
  Heteroskedasticity plots are used to identify unequal variances in the error term of a time series model over time. By visually inspecting these plots, we can **determine if the model has successfully accounted for changes in data volatility**. A good fit is indicated by the absence of any noticeable patterns or trends.
- **Scatter Plots with Regression Line**:
  Scatter plots accompanied by a regression line enable us to **assess the overall fit and accuracy** of the hybrid model. These plots help us identify any systematic biases or potential outliers, ensuring a more robust evaluation of the model's performance.
- **Data Visualization**:
  Data visualization plays a crucial role in evaluating the performance of the hybrid model, it enables us to **assess how well the model captures sudden changes in volatility** or other events that may impact stock prices. Through this method, we can make more informed decisions about the model's suitability for specific applications or determine necessary improvements.

By employing these performance evaluation methodologies, we **gain a comprehensive understanding** of the hybrid model's strengths and weaknesses, enabling us to optimize its performance.

Evaluating the Hybrid Model

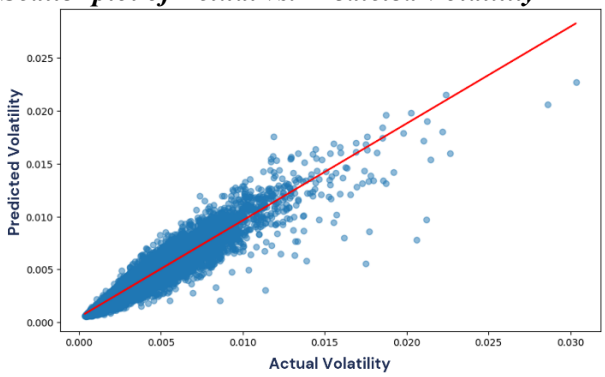
Residuals and Heteroskedasticity Analysis



Below is an outline of key assumptions that a residual and heteroskedasticity plot should meet, along with their performance in our hybrid model:

Assumptions	Residuals	Heteroskedasticity
Linearity: The relationship between the independent and dependent variables should be linear.	✓	✓
Independence: The residuals should be independent of each other, i.e., there should be no correlation between them.	✓	✓
Homoscedasticity: The variance of the residuals should be constant across all levels of the independent variable(s).	✓	✓
Normality: The residuals should follow a normal distribution and symmetrically distributed around zero, with most residuals being close to zero.		✓
No influential points/ outliers: No extreme values/ outliers in the dataset that disproportionately influence the model's performance.	✗	
Normally distributed errors: The error term should be normally distributed, with a mean of zero.		✓

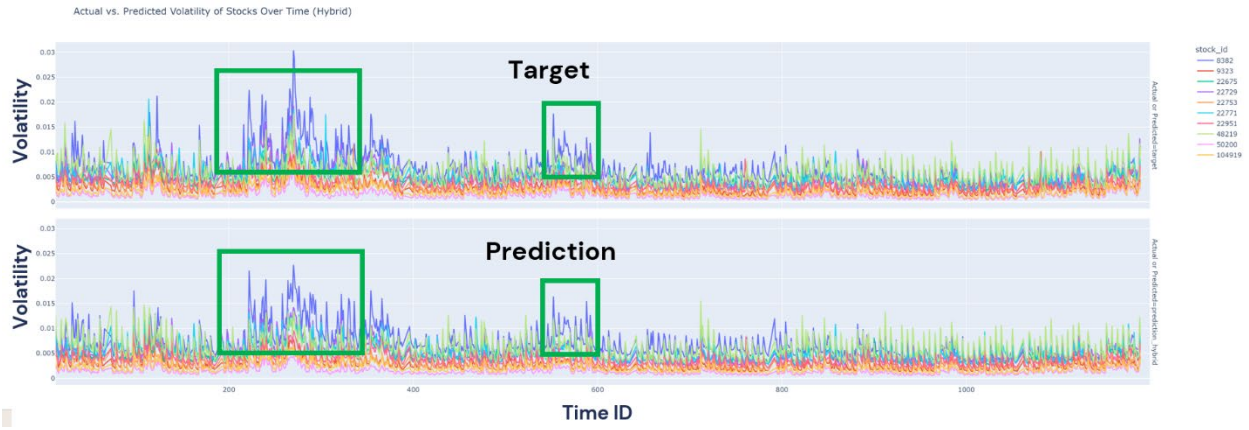
Scatter plot of Actual vs. Predicted Volatility



The scatter plot shows a linear pattern with a positive slope, **indicating a strong positive correlation** between the predicted values and the actual values. The closer the data points are to the line, the better the predictions. There are **only a few outliers** as well.

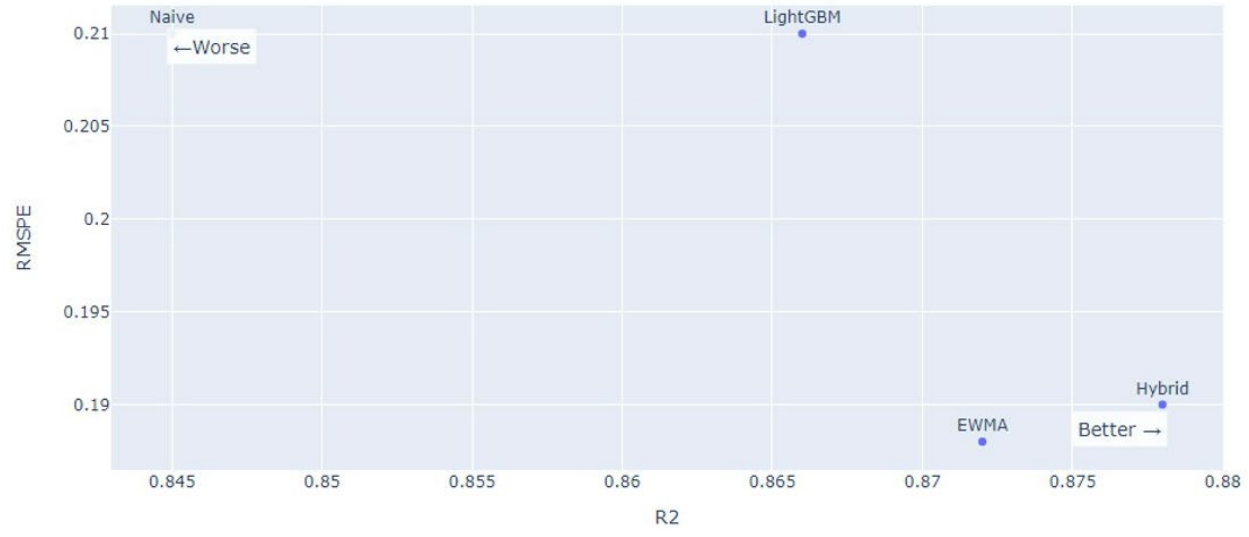
Data Visualization

Here is a plot for the targeted volatility versus our hybrid model predicted volatility, given that time ID from 0 to 1200.



	Hybrid Model Performance	Naïve Model Performance
R^2 Score	0.878	0.845
RMSPE	0.19	0.21
MAE	0.000587	0.00066
RMSE	0.000881	-

Overview of Model Performance



Here is an overview of our hybrid model using RMSPE against R^2. First, we implemented the EWMA and LightGBM models. Then by combining them into a hybrid model using Bayesian averaging, we achieved **a higher R^2 and a lower RMSPE** than the Naïve Model. The runtime of the hybrid model is 16.008 seconds accounting for Bayesian averaging's fast computation of 0.008 seconds.

Risks and Mitigation

In developing our final model, we have identified several risks and methods to mitigate them depicted in the chart.

First, **data quality** is a risk as the accuracy of the model's prediction relies on the quality of the dataset. A dataset full of errors and biased data would result in biased predictions. As such, we have mitigated the risk by pre-processing the data beforehand.

Second, the **computational resources** required are a risk as a hybrid model tends to be complex and thus requires a long time to generate results. We have mitigated this risk by reiterating and optimizing our code to significantly reduce this processing time. For example, we have reduced the time taken to run the EWMA component of our model from 9 minutes to 10 seconds.

Third, **heteroskedasticity** is a risk as constant variability in errors is desirable. We have mitigated this risk through the Bayesian Averaging process which incorporates different model structures and parametrizations to better capture variability. The averaging process thus reduces the impact of heteroskedasticity as it cancels out the effects. Further, we have also applied a smaller time window to our EWMA model to better capture variability by the model.

Limitations and Recommendations

First, there is limitation in the model's capability in **interpreting datasets**. The current codes limit the model to interpreting datasets with a format that is the same as the book and train datasets provided. Hence, an improved model would easily interpret datasets of different formats and contents. This improvement can be done by coding for a model that can pre-process and clean data, as well as undergoing model architecture so that the model is more adaptable and flexible to different datasets.

Second, the computation method used for the **Bayesian Weightings** uses comparisons of the model's predictions and the target dataset, which will not be the case in the real world since actual values will not be available. An approach to this limitation is to use the first 15 minutes of the training dataset being processed to predict 15 minutes of data after. Accordingly, use the predictions of the model to create the required weightings. The model can then predict the 30-minute interval using the weighting calculated in the Bayesian Averaging process.

Third, the model's predictions are **dependent on the dataset used**. An improved model would consider external information such as news sources or market sentiments, further, additional financial indicators such as market breadth or fundamental indicators (e.g., revenue growth, debt levels). However, an improved model would also go further in depth on the provided dataset and consider additional factors. For example, the addition of a time aspect as stated by Virginia could improve model performance.

Therefore, there are various steps to follow to improve these limitations in the future. First, external information should be gathered which includes collecting additional information such as news articles or other financial indicators to improve predictive power. Second, the model should be iterated to include the improvements or to modify the baseline model to boost performance. Finally, the model should then be trained with varying datasets to judge its performance and reiterated to ensure predictive accuracy.

Conclusion

Overall, the report provides a hybrid model for predicting realized short-term volatility that uses Bayesian averaging to combine the strengths of the EWMA and LightGBM models. In terms of R^2 and RMSPE, the findings suggest that the hybrid model surpasses the Nave Model. However, several risks are associated with the development of the hybrid model including data quality, computational resources, and heteroskedasticity. These risks were mitigated by pre-processing the data, optimizing the code, and adding various model structures and parametrizations via Bayesian averaging.

Furthermore, these risks consequently highlight the model's limitations. It is restricted in capacity to interpret datasets, the computational method used for Bayesian averaging, and the model's dependency on the historical datasets used limits the predictive power of the hybrid model. We have identified areas of improvements such as gathering external information, iterating the model to include improvements, and training it with varying datasets to ensure predictive accuracy.

Overall, the hybrid model developed serves as a guide and framework to further improve volatility prediction in Optiver. Although there are inherent risks and limitations associated with its development, the model **can be further optimized and trained to achieve even greater predictive accuracy**.