POLITECNICO
MILANO 1863

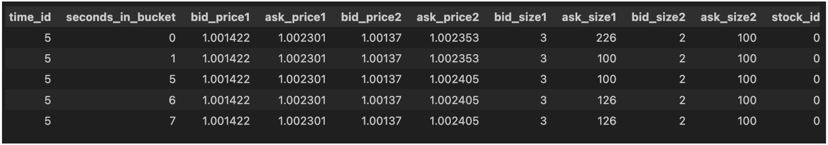
Forecasting Realized Volatility

Evidence from the Optiver Volatility Challenge

Matteo Campagnoli, Riccardo Girgenti
Giacomo Kirn, Francesco Ligorio

Introduction

- **Time Bucket:** interval of 600 seconds (10 minutes)
- Orderbook and trade data for 126 stocks



time_id	seconds_in_bucket	bid_price1	ask_price1	bid_price2	ask_price2	bid_size1	ask_size1	bid_size2	ask_size2	stock_id
5	0	1.001422	1.002301	1.00137	1.002353	3	226	2	100	0
5	1	1.001422	1.002301	1.00137	1.002353	3	100	2	100	0
5	5	1.001422	1.002301	1.00137	1.002405	3	100	2	100	0
5	6	1.001422	1.002301	1.00137	1.002405	3	126	2	100	0
5	7	1.001422	1.002301	1.00137	1.002405	3	126	2	100	0

Figure: Orderbook data for $stock_{id} = 0$ and $time_{id} = 5$.

Time ids are **not sequential!**

- We are asked to predict the realized vol in the next 10 minutes for 112 stock in 3830 time buckets.

Forecasting Realized Vol

- **Cross-sectional approach:** each row is treated independently, ignoring time structure
  - Fit models using only contemporaneous features (e.g., prices, spreads, volumes)

- **Time-ordering approach:** we attempt to reconstruct a meaningful temporal sequence across `time_ids`
  - Use Spectral Embedding, PCA, or t-SNE to recover time order
  - Enables time series models with lagged volatility

Benchmark

It's known that volatility in financial data is strongly auto-correlated. Especially on short time frames, we can use the past realized volatility as a good benchmark for the future.

$$\hat{\sigma}_{t+1} = \sigma_t \qquad \mathit{RMSPE} = 0.341$$

Linear Model: features

- P_t: average de-normalized price
- σ_t: past realized volatility (target in previous bucket)
- σ_t^{GK}: Garman-Klass estimator of volatility
- VI_t: volume imbalance (ask size - bid size), proxy for market pressure
- BA_t: bid-ask spread, proxy for liquidity

$$\hat{\sigma}_{t+1} = \beta_0 + \beta_1 P_t + \beta_2 \sigma_t + \beta_3 \sigma_t^{GK} + \beta_4 VI_t + \beta_5 BA_t$$

Linear Model: results

- Performance is only slightly better than the naive benchmark ($RMSPE > 0.30$)
- Most predictive feature: σ_t, due to strong autocorrelation
- Some features are significant (e.g., price, spread), others less consistent (e.g., imbalance)

Feature	**Coefficient**	**Std. Error**	**P-value**
β_0	1.259×10^{-5}	6.55×10^{-5}	0.847
P_t	1.431×10^{-7}	1.89×10^{-8}	0.000
σ_t	0.6205	0.025	0.000
σ_t^{GK}	0.0771	0.019	0.000
VI_t	-2.496×10^{-5}	2.11×10^{-5}	0.237
BA_t	1.0880	0.234	0.000

Linear Model: Diagnostics

Jarque-Bera: $< 2.2e - 16$
Residuals are significantly non-Gaussian, violating OLS assumptions.

Condition Number: 6.74×10^7 Indicates strong multicollinearity among the predictors.

Conclusion: The model has decent predictive power, but results are not reliable for statistical inference.

Linear Model: a trick

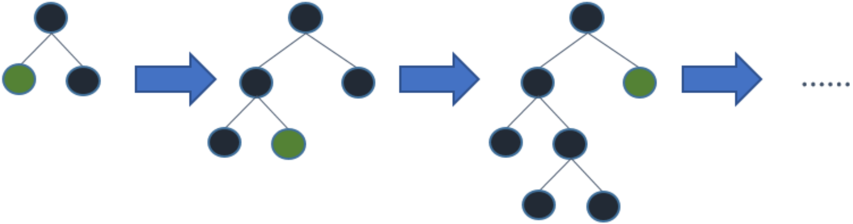
The current linear model is neither accurate nor solid enough to make inference. However, we can improve the predictive power by introducing some weights

$$\hat{\beta}_{WLS} = \arg\min_{\beta} \sum_{i=1}^{d} w_i \big(y_i - X_i^T \beta\big)^2 \qquad w_i = \frac{1}{y_i^2}$$

$$\implies RMSPE = 0.24679$$

Common trick when predicting a percentage error.

LigthGBM: Light Gradient Boosting Machine

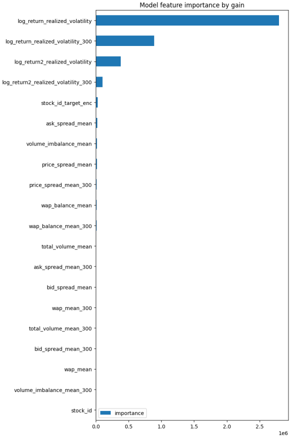


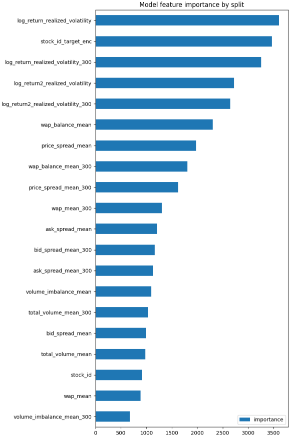
Leaf-wise tree growth

LightGBM: an introduction

- LightGBM is a decision tree-based ensemble method using gradient boosting.
- Instead of fitting a single tree, it builds trees sequentially, each one improving on the residuals of the previous.
- Efficient: uses histogram-based binning and leaf-wise tree growth.
- It handles large-scale, high-dimensional datasets well.

LightGBM: feature importance





LightGBM: feature importance

- **Split importance** (left): counts how many times each feature is used to split the data. Shows model structure.
- **Gain importance** (right): measures the total reduction in loss each feature provides. Captures predictive power.
- The past realized volatility is by far the most important feature under both metrics.
- Recent values (last 300 seconds) also contribute significantly, confirming short-term memory in volatility.

Time Embedding: Why and How

- The time_id is not chronologically ordered – we can't use time-series models directly.
- But price vectors across all stocks at a given time_id contain implicit temporal information.
- We treat each time_id as a point in a high-dimensional space (1 point = vector of prices for all stocks).
- We apply dimensionality reduction to uncover a 1D manifold: a smooth trajectory over time.
- This embedding enables us to reintroduce temporal structure and apply autoregressive-like models.

Temporal Ordering: PCA and tSNE

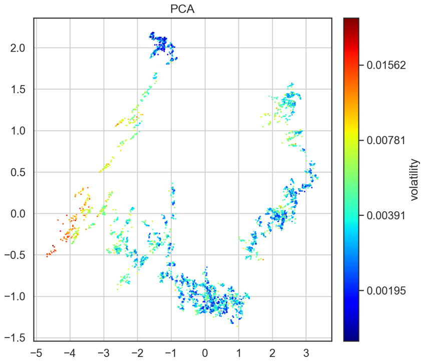


Figure: PCA

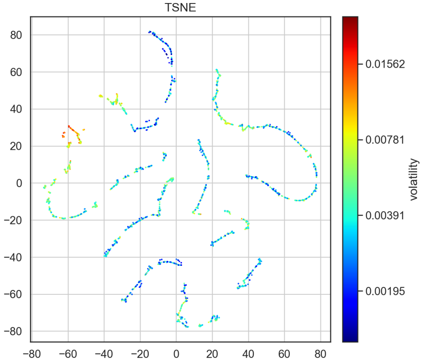
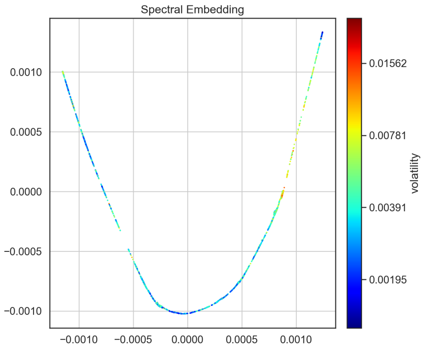


Figure: Gain Importance

Temporal Ordering: Spectral Embedding



- **Smooth line:** the embedding reveals low-dimensional structure — consistent with the idea of an underlying time evolution.
- **Meaningful variation:** color volatility varies smoothly along the curve, indicating that the embedding captures relevant market dynamics.
- **Not enough:** visual coherence is encouraging, but only improves forecasting if the ordering enhances model performance.

Temporal Ordering: AMZN

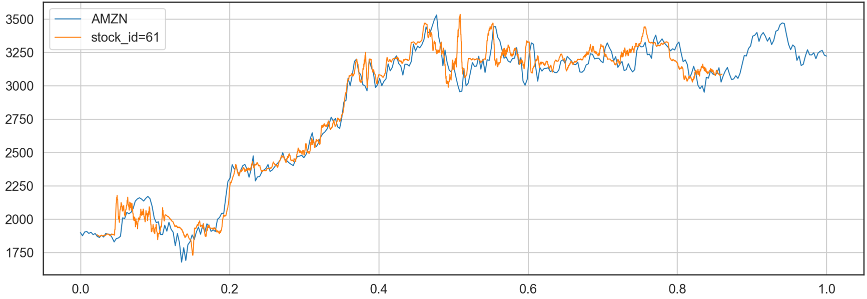


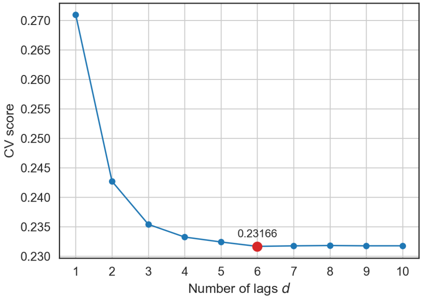
Figure: stock_id = 61

Temporal Ordering: fitting a model

$$\hat{\sigma}_{t+1} = \beta_0 + \beta_1 \sigma_t + \beta_2 \sigma_{t-1} + \cdots + \beta_d \sigma_{t-d+1}.$$

Warning: this is not a AR(d), we are just predicting. No inference of any kind.

Temporal Ordering: results



- $RMSPE = 0.23166$
- We used the same normalization trick

Discussant Group: Summary of Results

- **Linear models:**
  - Log-log regression with fixed effects
  - Best variant (Model III): RMSPE = **0.2487**, $R^2 = 0.8253$
- **LSTM:**
  - Trained on order book time series
  - Achieves best performance: **RMSPE = 0.2349**
  - Gain over linear model is small
- **Conclusion:** Most predictive power comes from past realized volatility. Complex models offer limited marginal gains.

Discussant Group: a question

- How does your LSTM architecture work? What features are you using? Have you employed any regularization technique?