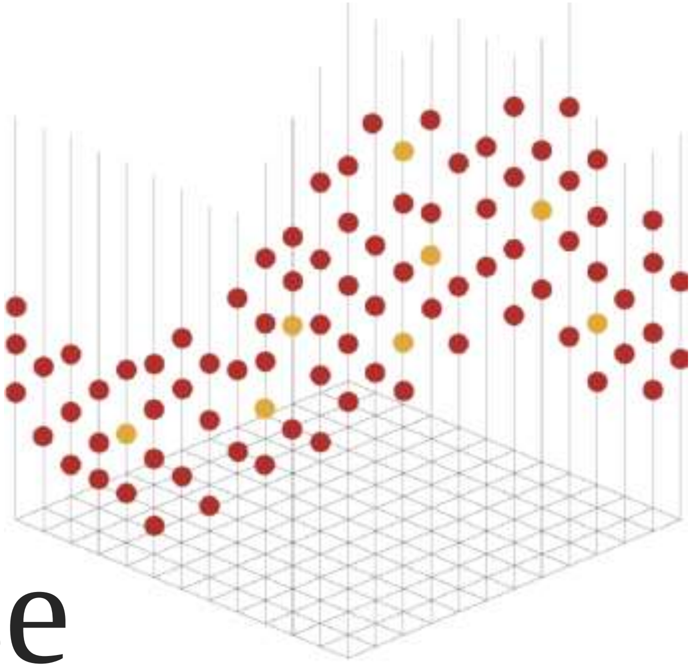


Trading At the Close
-- Predict US stocks closing movements

DATA1030 FINAL PRESENTATION: YU, LETIAN

BROWN UNIVERSITY

GitHub: https://github.com/LetianY/data1030-optiver-trading-at-close/

Introduction - Recap

NASDAQ Stock Market:

- Rapid price change in last 10 min
  (10% of average daily volume!)

- **Dataset:** historic data for the daily ten minute closing auction

- **Data Source:** Kaggle by Optiver

- **Data Collection:** order books and the closing auctions of the stocks

- **Goal:** predict closing price movements for hundreds of listed stocks

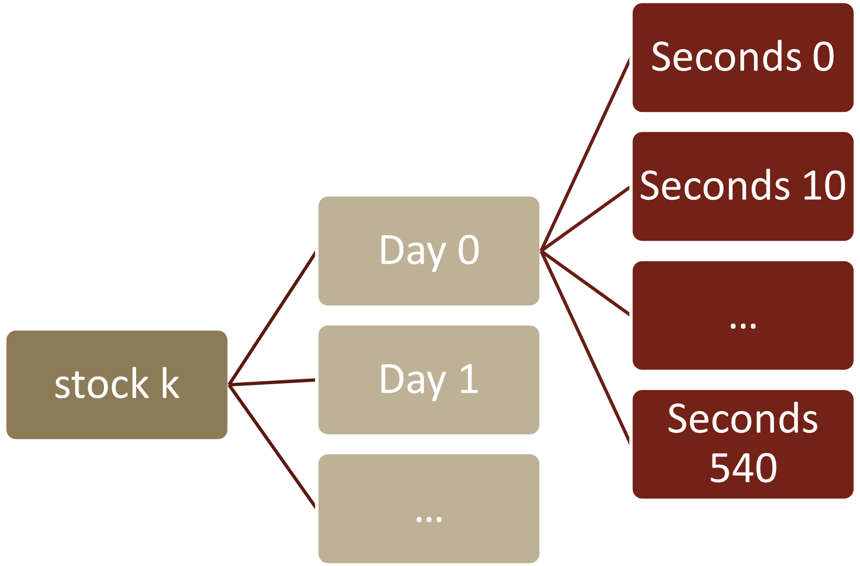
- **Problem Type:** Regression

- **Target:** synthetic index
  (closing price movement)

- **Importance:**
  - prices adjustment
  - supply and demand dynamics
  - trading opportunities

Introduction - Recap

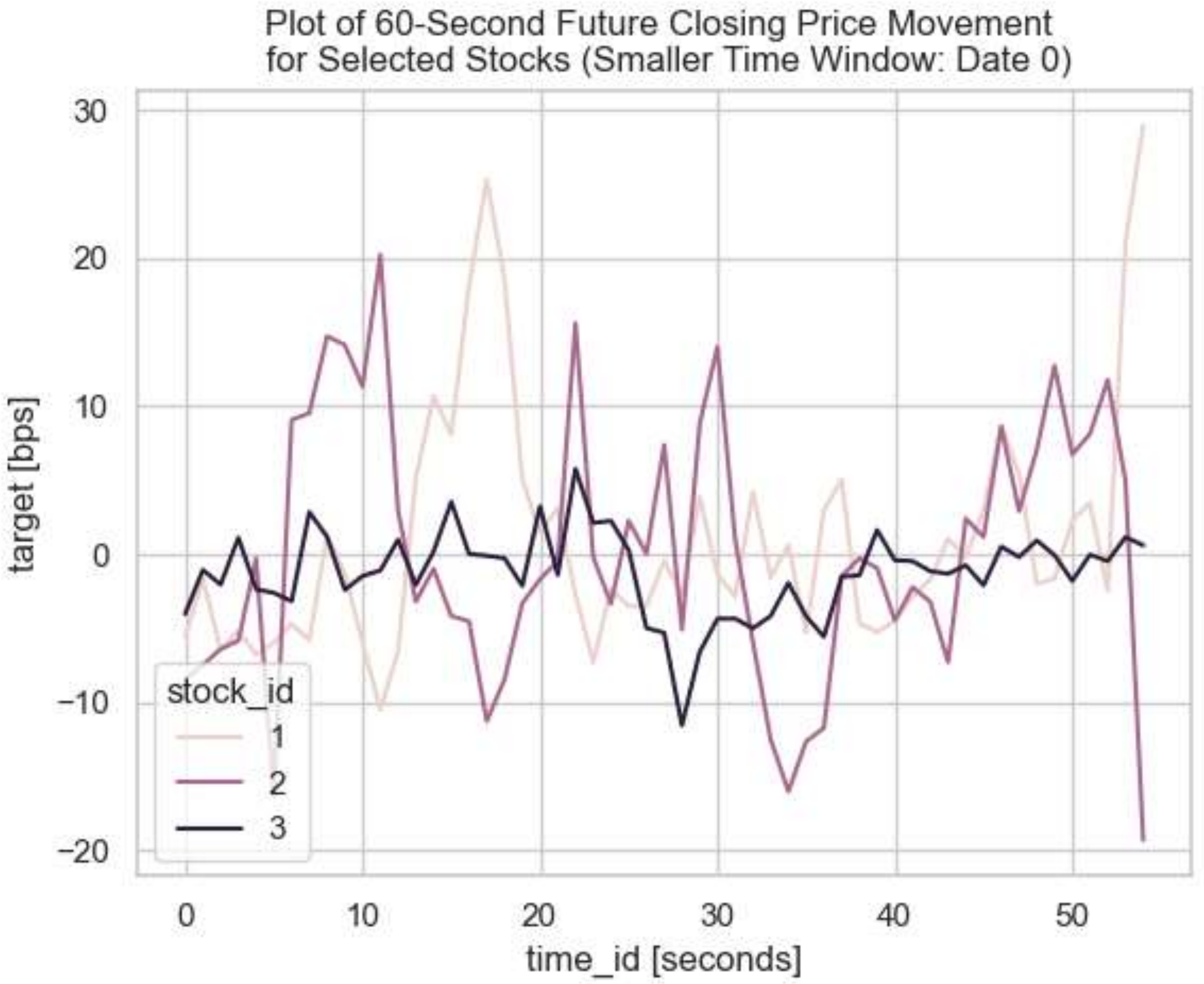
- **Missing data:** time structure & features
- **Time series data:** non-iid
- **Large dataset:** data points in millions
- **Domain Knowledge**



Letian Yu
Brown DSI

EDA Recap

- Volatility
- Extreme Values
- Mean Reversion



Letian Yu
Brown DSI

EDA Recap

- Volatility
- Extreme Values
- Mean Reversion

The plot shows the time series plot of the target 60-second future closing price movement index for selected stocks. We see that different stocks shows different volatilities and there exists extreme values. But in general, mean reversion towards zero is perceived.

Feature Engineering
& Time Series Lagged Features

Manually Constructed Features:

- **lagged targets:** for 1 day (55 features)

- **pairwise price imbalances:** (a-b)/(a+b)

- **pairwise size imbalances:** (a-b)/(a+b)

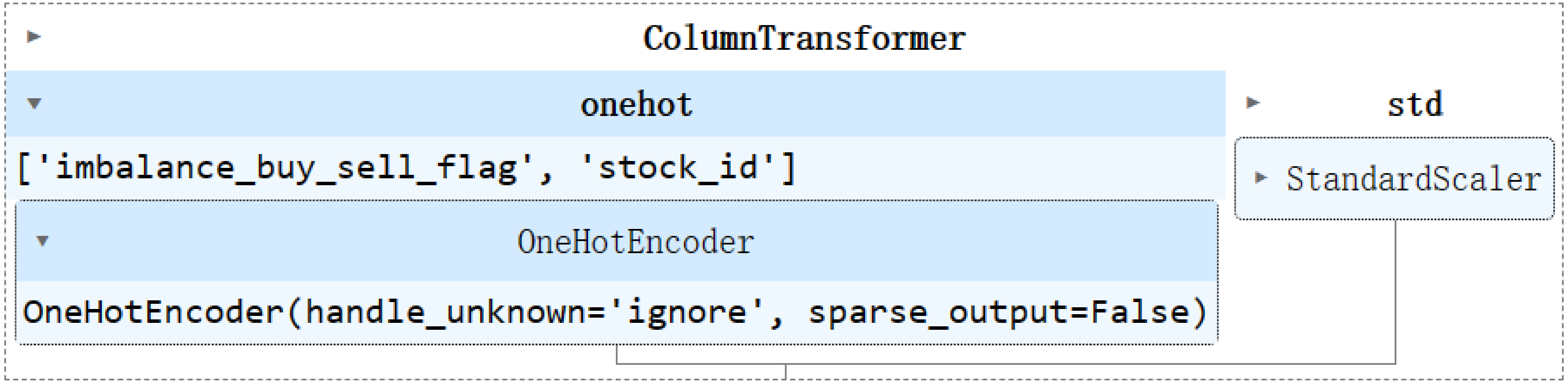
- **statistics:** mean, std, skewness, kurtosis

- **other features with financial meaning:** e.g., price pressure, market urgency

Preprocessing Recap:

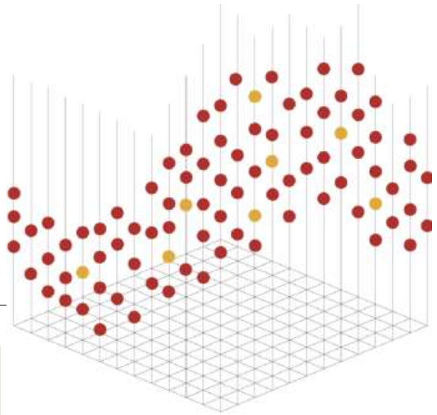
Column Transformer:

- **One-hot Encoder (categorical features):** stock id, buy and sell imbalance flag
- **Standard Scaler:** Other continuous features



Letian Yu
Brown DSI

Data Splitting



GroupTimeSeriesSplit: date_id chosen as group
- Deterministic splitting
- Test group size = Val group size = 0.2 * total group size

Fixed test set: (see example below)
- In real life and competition, test set is always fixed.
- 4 Folds Train-Val split to measure randomness.

date index	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Fold 1	train	train	train	val	val				test	test
Fold 2		train	train	train	val	val			test	test
Fold 3			train	train	train	val	val		test	test
Fold 4				train	train	train	val	val	test	test

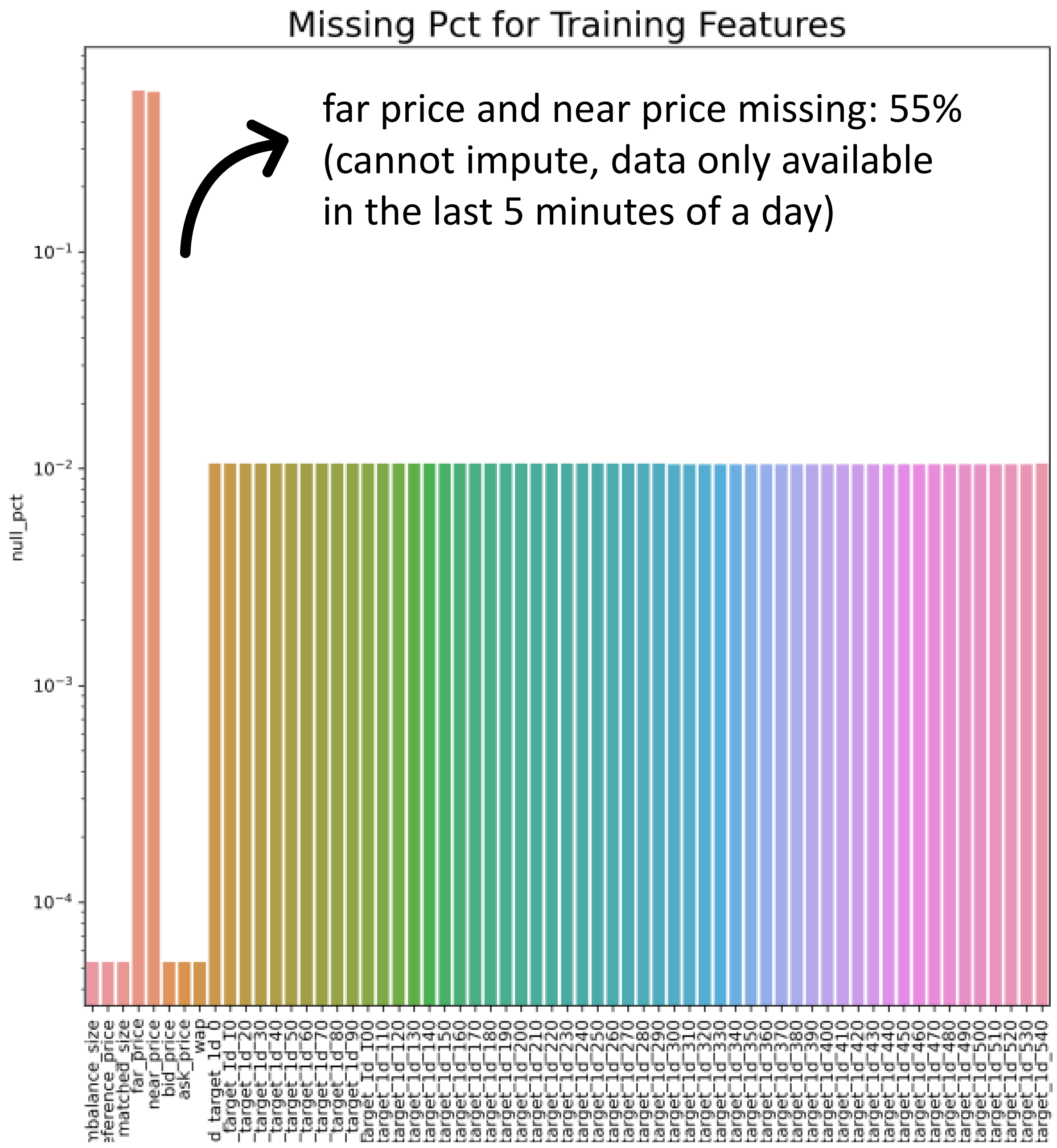
ML Models

- Drop records with target missing:
extremely small proportion of data

- Memory optimization:
halved memory usage by converting dtypes

- Reconstructed dataset:
1. original date 0-481
2. now only using date 0-120
3. Otherwise unable to run reduced feature method with limited memory.

- 3 patterns of missing value in feature



Letian Yu
Brown DSI

ML Models: Handle Missing Values

XGBoost directly handles missing values!

Reduced Feature: all ML algorithms
Train 3 sub-models for 3 missing pattern

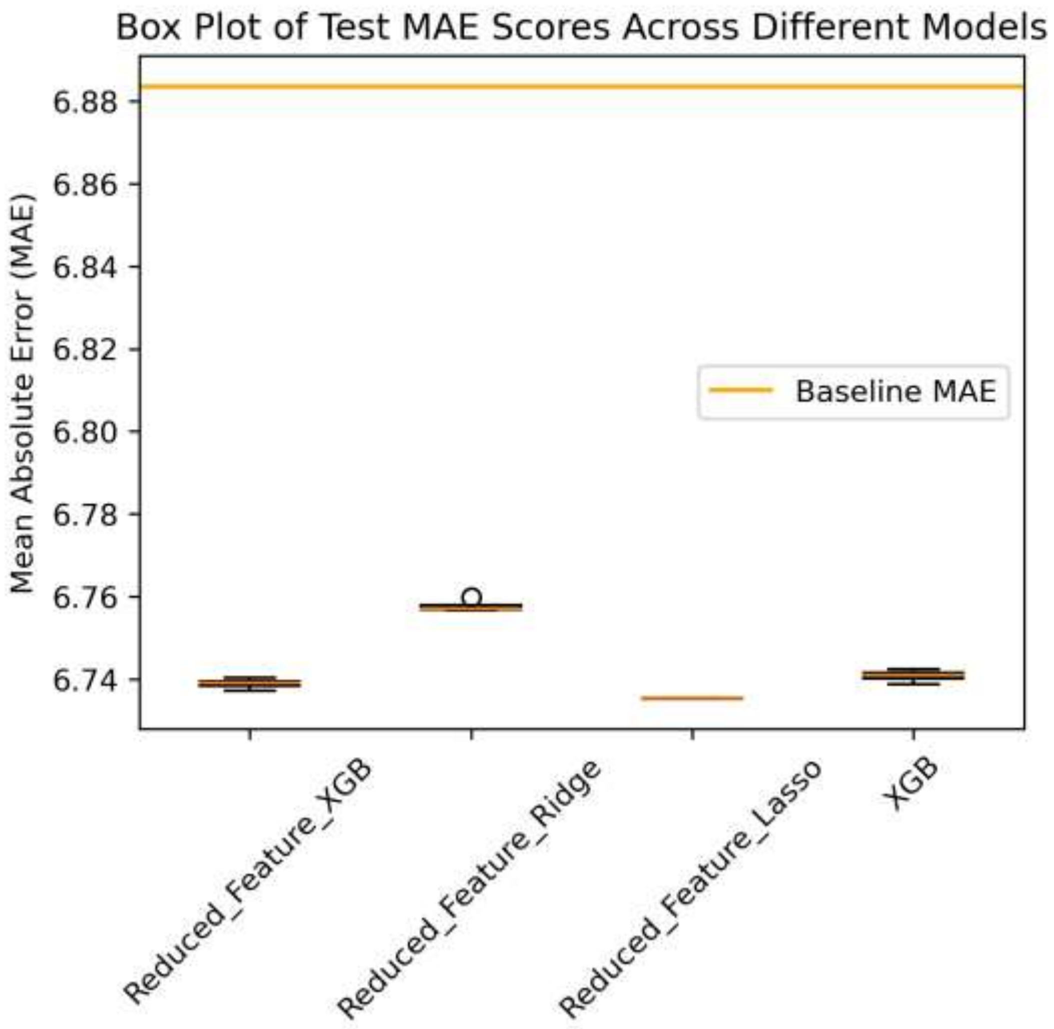
Tried models: Lasso, Ridge, SVR, RF, XGBoost

ML Models & Hyper Parameters

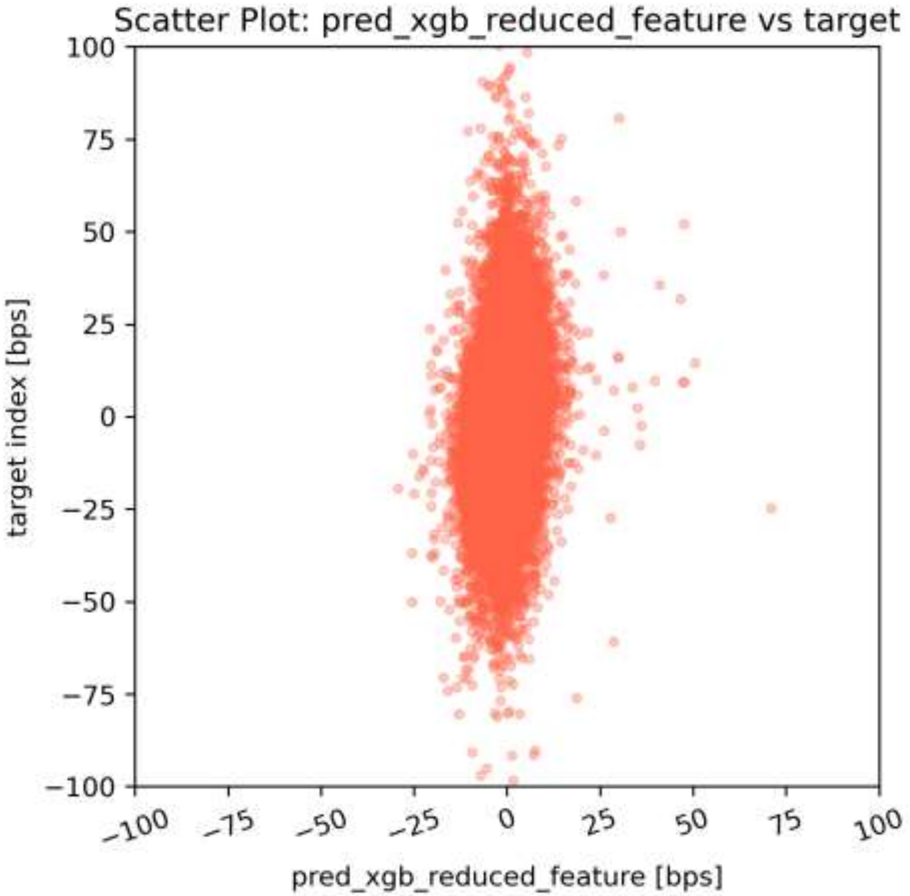
Model Name	**Final Parameters Tunned**	**4-Fold Running Time**	**Device**
XGBoost	reg_alpha: [0, 0.01, 0.1, 1], reg_lambda: [0.1, 0.5, 1, 2], max_depth: [1, 3, 7, 11, 13]	15444.11 seconds (> 4 hours)	GPU P100
Reduced Feature Method: Lasso	alpha: np.logspace(-2, 1, 15)	3073.89 seconds (< 1 hour)	CPU
Reduced Feature Method: Ridge	alpha: np.logspace(-2, 1, 15)	896.27 seconds (about 15min)	CPU
Reduced Feature Method: Random Forest	max_features: [0.5, 0.75, 1.0, None], max_depth: [1, 5, 7, 11, 13, None]	>12 hours per fold, time limit exceeded, abandoned	CPU
Reduced Feature Method: Support Vector Machine	gamma: [0.001, 0.1, 10, 1000], C: [0.01, 1, 10]	time limit exceeded, abandoned	CPU
Reduced Feature Method: XGBoost	reg_alpha: [0, 0.01, 0.1, 1], reg_lambda: [0.1, 0.5, 1, 2], max_depth: [1, 3, 7, 11, 13]	5821.88 seconds (1.61 hours)	GPU P100

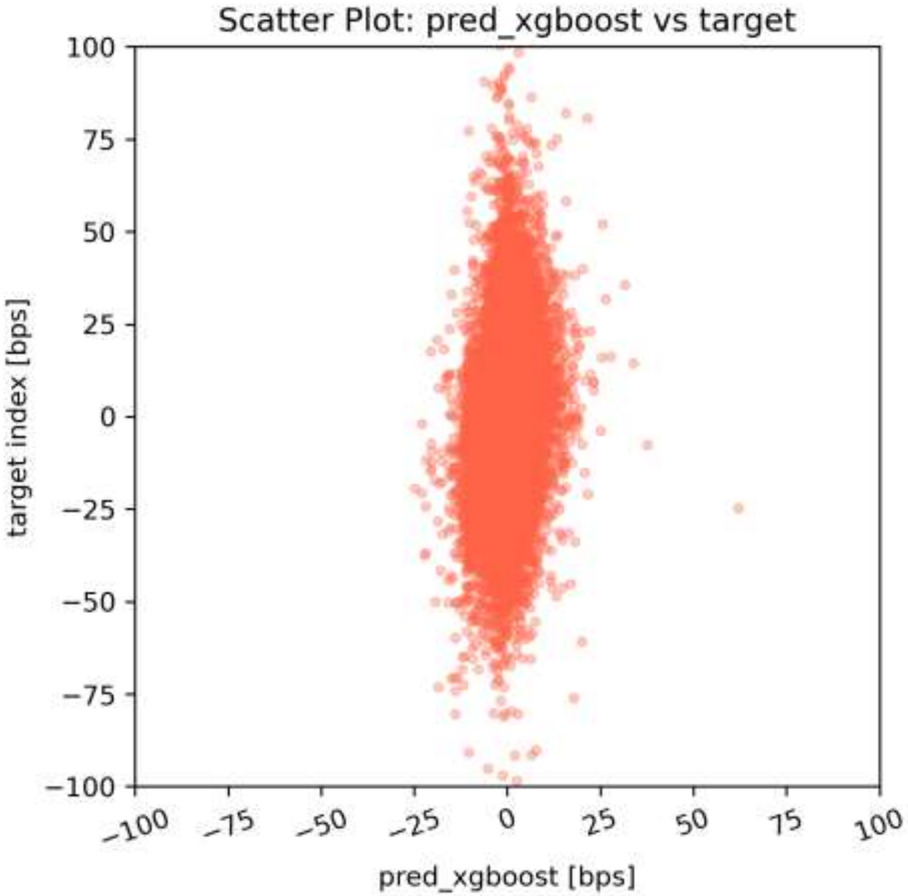
Model Evaluation

Model Name	Test Scores
Baseline: prediction = mean of target index in test set by stocks	MAE: 6.8835
XGBoost	[6.7388, 6.7423, 6.7413, 6.7407] mean MAE: 6.7407, std: 0.0013
Reduced Feature Method: Lasso	[6.7355, 6.7354, 6.7354, 6.7353] mean MAE: 6.7354, std: 0.00005
Reduced Feature Method: Ridge	[6.7569 6.7568 6.7599 6.7573] mean MAE: 6.7577, std: 0.00126
Reduced Feature Method: XGBoost	[6.7403, 6.7393, 6.7388, 6.7371] mean MAE: 6.7383, std: 0.0011



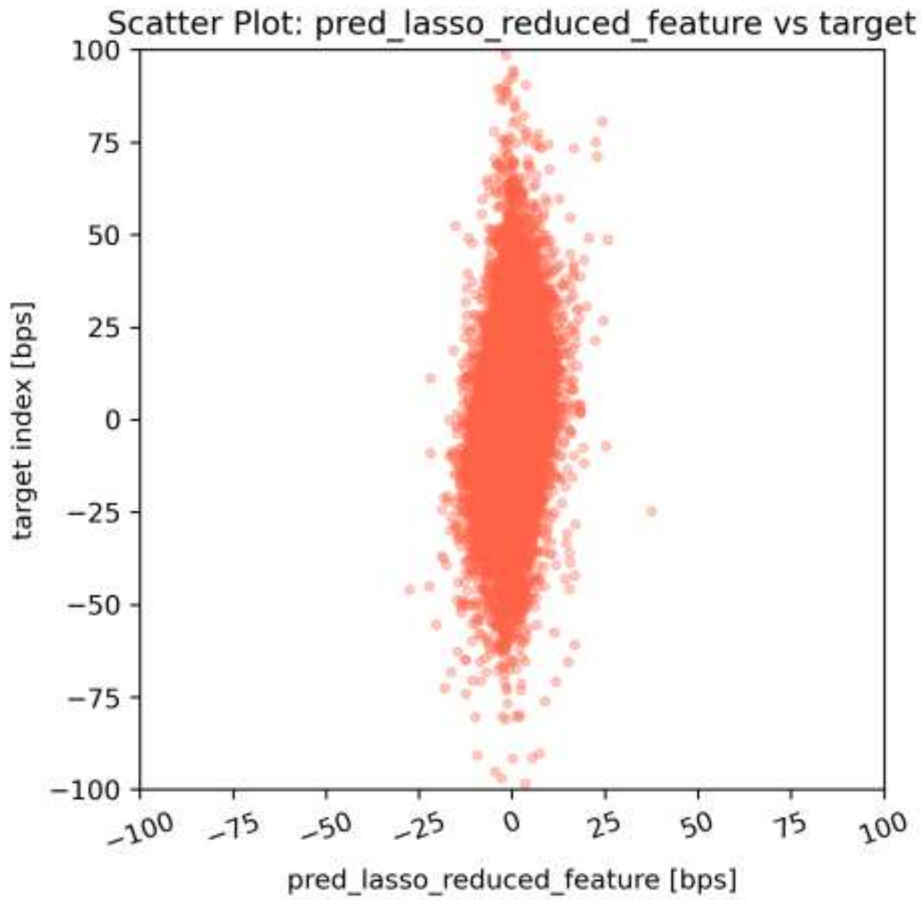
Scatter Plots ☹

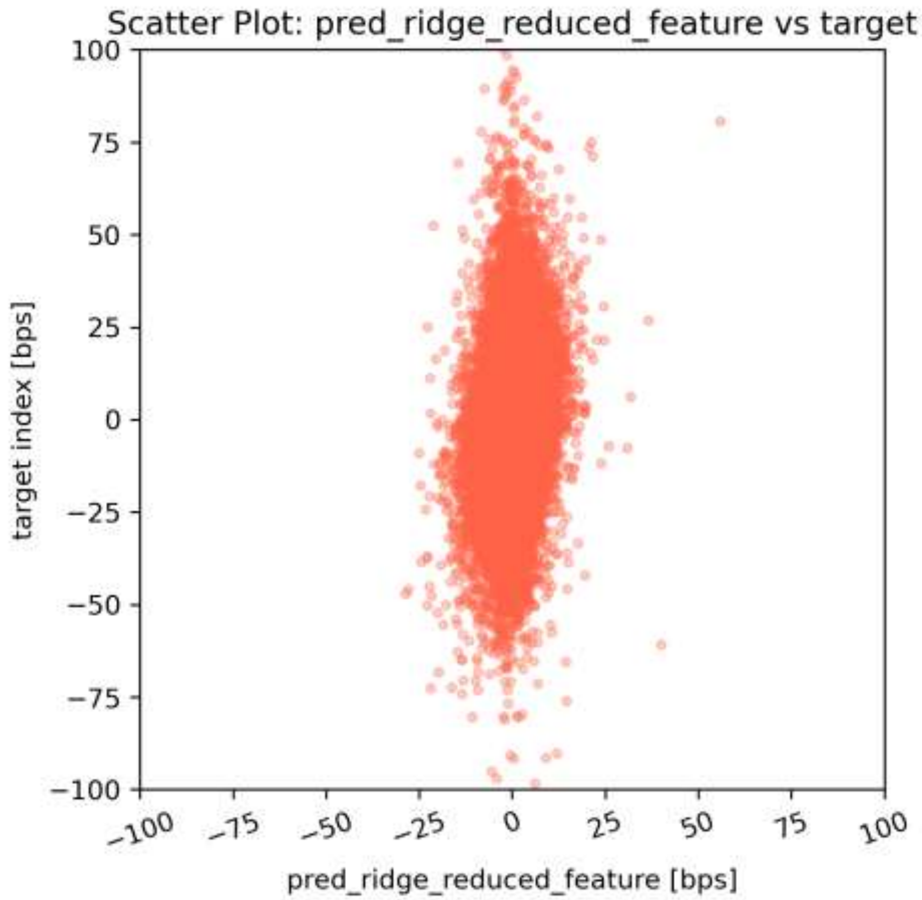




Letian Yu
Brown DSI

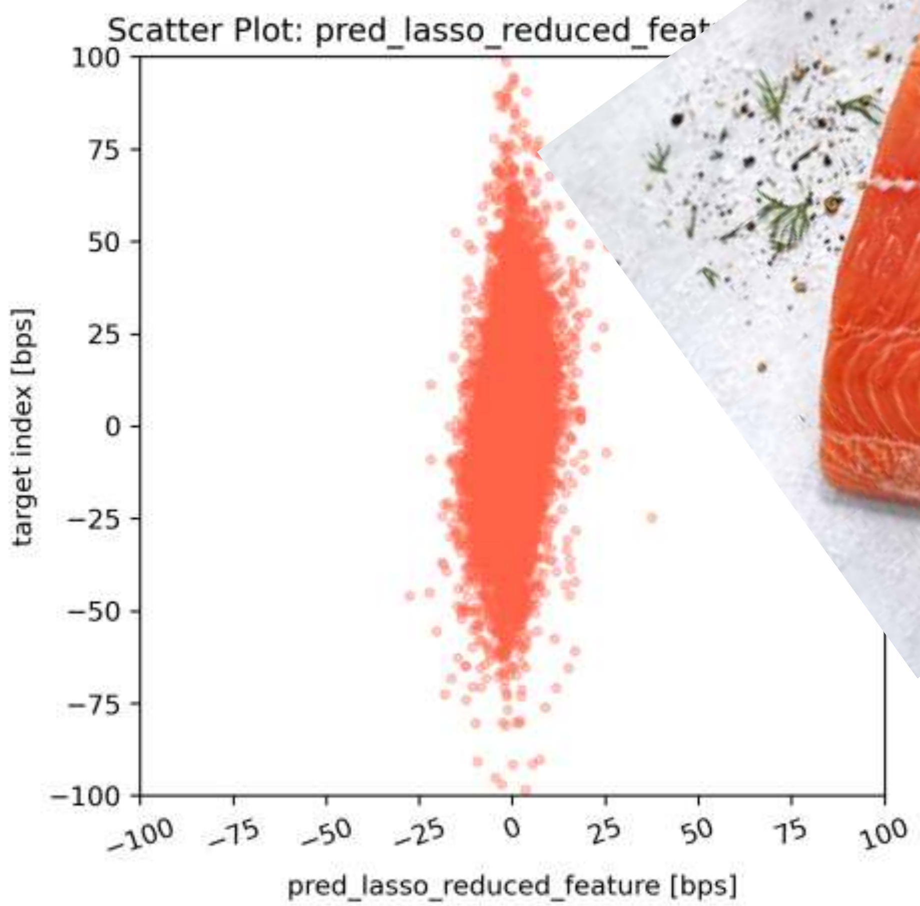
Scatter Plots ☹

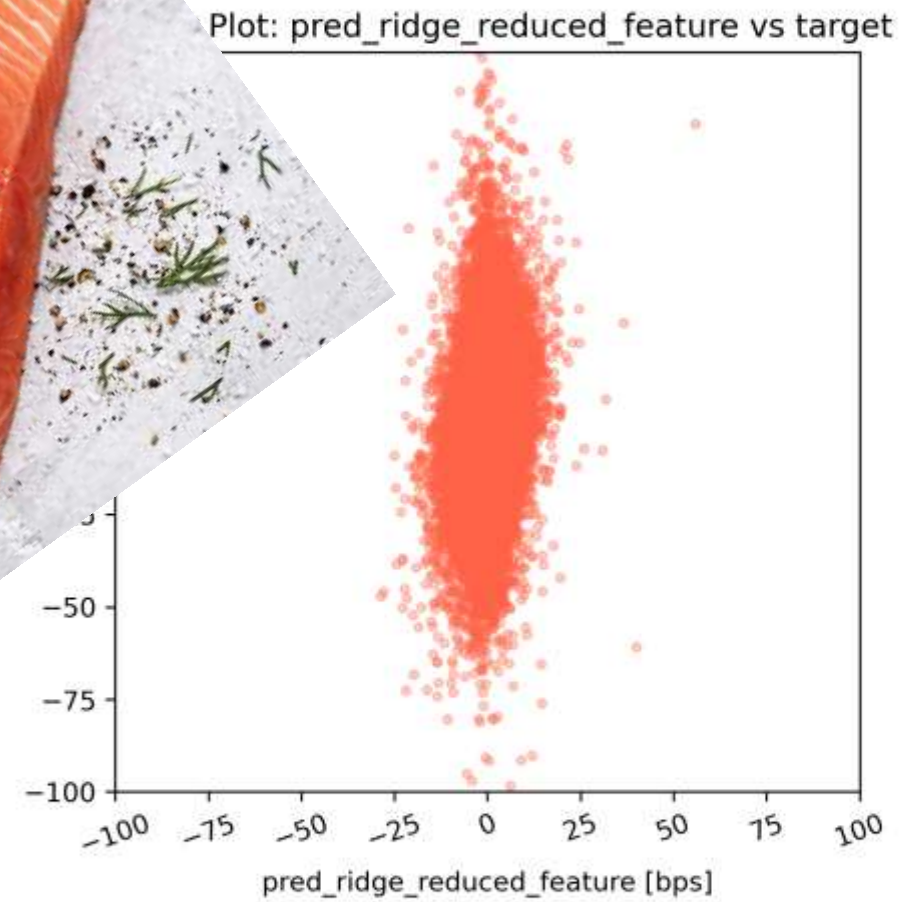




Letian Yu
Brown DSI

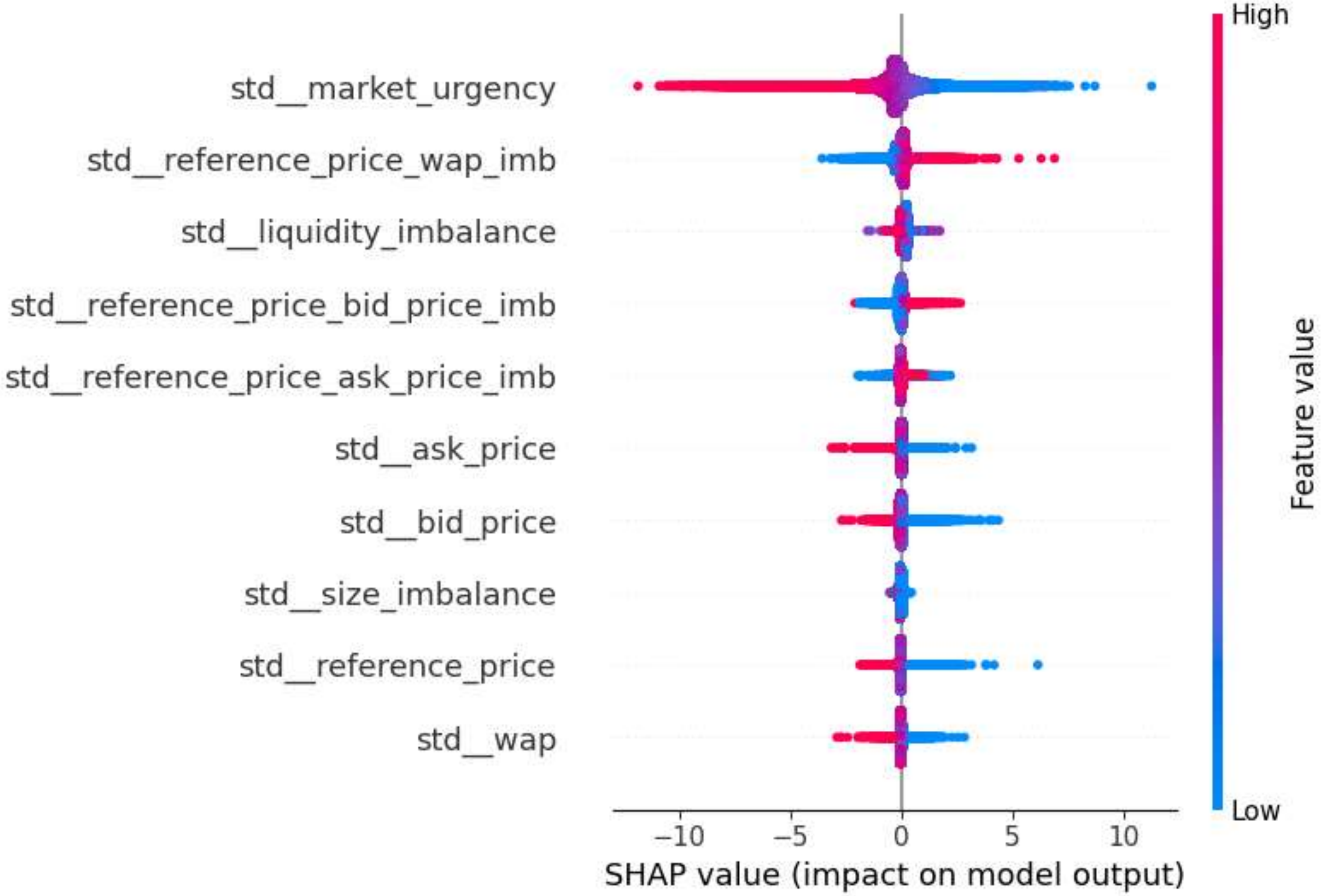
Scatter Plots ☺





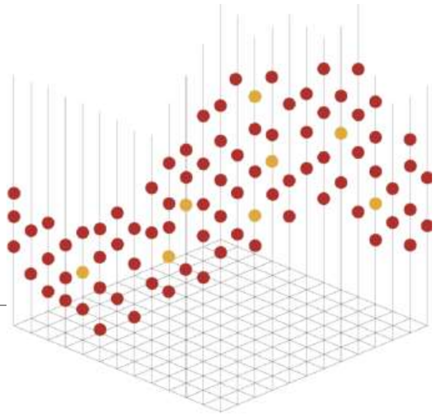
Letian Yu
Brown DSI

Global Shap: XGBoost



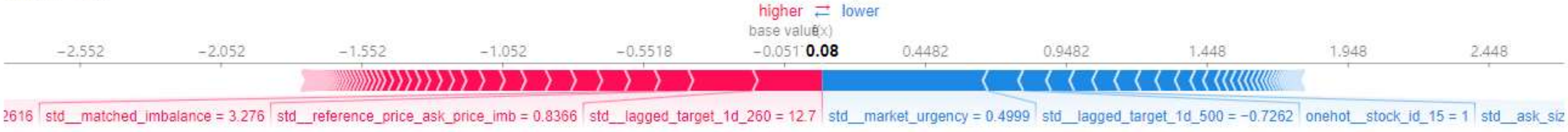
Letian Yu
Brown DSI

Local Shap: XGBoost

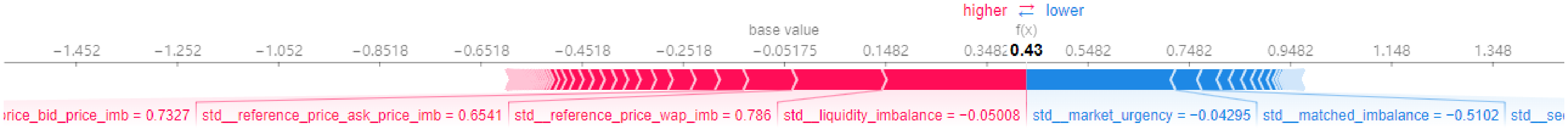


index: 0

index: 1000



index: 2134



Letian Yu
Brown DSI

Outlook

Predictive Power Enhancement:

1. Feature Engineering: construct more features to capture the market factors
2. Fine-tuning of hyperparameters: include larger parameter grids
3. Upgrade device: handle the whole dataset with more resources

Model Interpretability:

1. Reduced Feature Method: multiple models
2. Highly correlated features: permutation importance can fail
3. Comparison between models

1. Introduction: Recap EDA, Preprocessor, Feature Engineering

2. Model Training Pipeline:

- Data splitting
- Hyper-parameters tuning

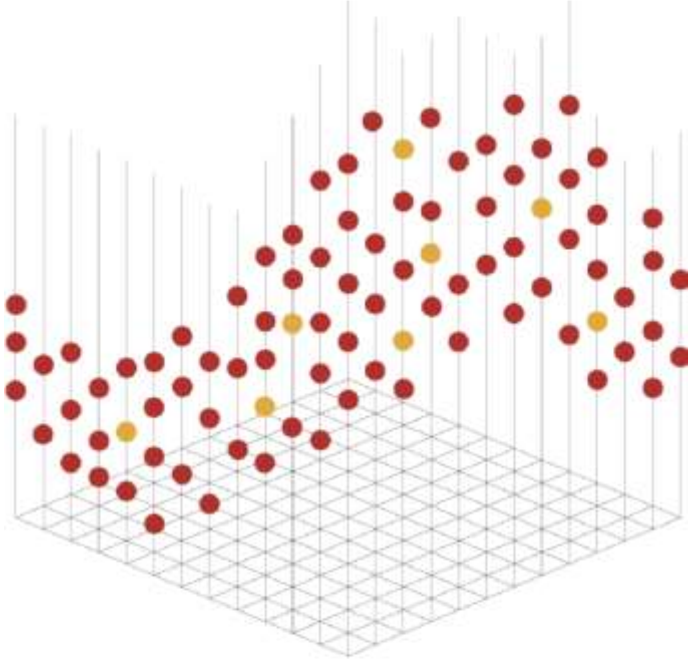
3. Results

- Baseline model and test scores
- Scatter Plot: true vs predicted
- Feature Importance and Interpretability

4. Outlook

Summary

Letian Yu
Brown DSI



Q & A

THANK YOU!

GitHub: https://github.com/LetianY/data1030-optiver-trading-at-close/

Appendix: Feature Table

Features	Description
stock_id	A unique identifier for the stock. Not all stock IDs exist in every time bucket.
date_id	A unique identifier for the date. Date IDs are sequential & consistent across all stocks.
imbalance_size	The amount unmatched at the current reference price (in USD).
imbalance_buy_sell_flag	buy-side imbalance: 1; sell-side imbalance: -1; no imbalance: 0
reference_price	The price at which paired shares are maximized, the imbalance is minimized and the distance from the bid-ask midpoint is minimized, in that order. Can also be thought of as being equal to the near price bounded between the best bid and ask price.
matched_size	The amount that can be matched at the current reference price (in USD).

Appendix: Feature Table

Features	Description
Far_price	The crossing price that will maximize the number of shares matched based on auction interest only. This calculation excludes continuous market orders.
Near_price	The crossing price that will maximize the number of shares matched based auction and continuous market orders.
Bid and ask price	Price of the most competitive buy/sell level in the non-auction book.
Bid and ask size	The dollar notional amount on the most competitive buy/sell level in the non-auction book.
wap	The weighted average price in the non-auction book.
seconds_in_bucket	The number of seconds elapsed since the beginning of the day's closing auction, always starting from 0.

Appendix: Target

Target

- The 60 second future move in the wap of the stock, less the 60 second future move of the synthetic index. Only provided for the train set.
  1. The synthetic index is a custom weighted index of Nasdaq-listed stocks constructed by Optiver for this competition.
  2. The unit of the target is basis points (bps), which is a common unit of measurement in financial markets. A 1 basis point price move is equivalent to a 0.01% price move.
  3. Where t is the time at the current observation, we can define the target:

$$Target = \left(\frac{StockWAP_{t+60}}{StockWAP_t} - \frac{IndexWAP_{t+60}}{IndexWAP_t}\right) * 10000$$

Letian Yu
Brown DSI