Optiver Trading at the Close:
Predicting Stock Price Movements

COMS-W4995 Applied Machine Learning Project

December 3, 2024

1 Introduction

This project focuses on predicting future price movements of stocks relative to a synthetic index composed of NASDAQ-listed stocks during the daily ten-minute closing auction. The challenge involves analyzing market microstructure data to forecast short-term price movements, which is crucial for market making and trading strategies.

2 Dataset Description

The dataset contains historical data from NASDAQ's closing auctions, including various market metrics:

- Time-based identifiers: stock_id, date_id, seconds_in_bucket
- Auction metrics: imbalance_size, matched_size, imbalance_buy_sell_flag
- Price indicators: reference_price, far_price, near_price, bid/ask prices
- Volume metrics: bid/ask sizes, weighted average price (WAP)
- Target variable: 60-second future move in stock WAP relative to synthetic index

The target is measured in basis points (1bp = 0.01%) and calculated as:

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

3 Feature Engineering

Our feature engineering approach creates a comprehensive set of features capturing different aspects of market behavior. Each feature category is designed to capture specific market dynamics that influence price movements during the closing auction:

3.1 Basic Market Metrics

- Volume and liquidity indicators:
  - Total volume (ask_size + bid_size): Represents overall market participation and potential volatility. Higher volumes often indicate increased activity and likelihood of significant price movements.
  - Mid-price calculation: Estimates fair value by averaging best bid and ask prices, helping detect shifts in market sentiment and supply-demand imbalances.
  - Liquidity imbalance indicators: Measure differences between buy-side and sell-side liquidity, indicating potential directional pressure on prices.

  - Size-based imbalance ratios: Quantify market pressure by comparing bid and ask order sizes, particularly relevant during closing auctions when traders adjust positions.

- Price combination features: Capture relationships between different price points (reference, far, near, ask, bid, WAP), revealing arbitrage opportunities and market inefficiencies.

- Triplet imbalance features: Analyze relationships across three related variables to provide deeper insights into market conditions and trading pressure.

3.2 Market Dynamics Features

- Price and spread metrics:
  - Price spread and intensity: Reflect market liquidity and trading costs, with changes indicating evolving market conditions.
  - Relative spread calculations: Enable consistent comparisons across different price levels, helping identify abnormal trading patterns.
  - Micro-price incorporating volume: Provides refined price estimates weighted by trading volumes, better indicating likely price direction.

- Market pressure indicators:
  - Imbalance momentum: Measures how quickly liquidity imbalances are changing, indicating strengthening market pressure.
  - Market urgency indicators: Capture trader aggressiveness through order submission and cancellation patterns.
  - Depth pressure calculations: Assess cumulative order volume at various price levels, identifying potential support and resistance.

- Statistical features: Calculate mean, standard deviation, skewness, and kurtosis to characterize price and size distributions, revealing unusual market behavior.

3.3 Time Series and Global Features

These features incorporate temporal patterns and stock-specific characteristics:

- Rolling window calculations:
  - Price and size differences: Capture short-term changes and trends, useful for detecting momentum or reversal patterns.
  - Returns and momentum indicators: Measure recent performance and trend strength relative to the index.
  - Temporal pattern features: Identify recurring patterns that may emerge from trading strategies or market regulations.

- Time-based features: Account for variations in trading behavior across different times (day of week, minute, seconds), as activity patterns often vary throughout the auction period.

- Stock-specific global aggregations:
  - Median and standard deviation of sizes: Establish baselines for normal trading activity, helping identify unusual volume patterns.
  - Price statistics and ranges: Provide context for current price movements based on historical behavior.
  - Historical trading patterns: Leverage past behavior to improve prediction accuracy during similar market conditions.

4 Model Implementation

We implemented four different models to capture various aspects of the price movement prediction problem:

4.1 LightGBM Model

A gradient boosting framework optimized for efficiency and performance:

- Objective: Mean Absolute Error (MAE)
- Key parameters:
  - 3000 estimators with 128 leaves per tree
  - 0.6 subsample and colsample ratios
  - Learning rate: 0.05
  - Early stopping with 100-round patience

4.2 GPU-Accelerated Random Forest

Implemented using RAPIDS cuML for parallel processing:

- Parameters:
  - 100 estimators
  - Maximum depth of 15
  - 8 CUDA streams for parallel processing
- Leverages GPU acceleration for both training and inference

4.3 GRU Neural Network

A recurrent neural network designed for sequential data:

- Architecture:
  - 2-layer GRU with 128-dimensional hidden state
  - Dropout rate of 0.2 for regularization
  - Final linear layer for regression
- Training:
  - Batch size of 256
  - Adam optimizer with MAE loss
  - Early stopping with 10-epoch patience

4.4 CNN-LSTM Hybrid

A hybrid architecture combining convolutional and recurrent layers:

- CNN component:
  - Two convolutional layers with ReLU activation
  - Feature extraction from raw input data
- LSTM component:
  - 2-layer LSTM with 128-dimensional hidden state
  - Dropout rate of 0.2 between layers
- Dense projection layer between CNN and LSTM

4.5 Training Approach

Common training methodology across all models:

- Data splitting:
  - 80/20 time-based train/test split
  - Ensures temporal consistency in evaluation
- Preprocessing:
  - Removal of missing values
  - Feature standardization where applicable
  - GPU acceleration for compatible models
- Validation strategy:
  - Out-of-time validation
  - Early stopping to prevent overfitting
  - Model-specific learning rate schedules

5 Results and Analysis

5.1 Model Performance Comparison

Performance metrics across all models on the test set:

- LightGBM: MAE = 4.9874 (best performing)
- Random Forest: MAE = 5.0308
- GRU: MAE = 5.1655
- CNN-LSTM: MAE = 5.1653

5.2 Error Distribution Analysis

- LightGBM shows the tightest error distribution:
  - Median error: 3.54 basis points
  - 75% of predictions within 6.64 basis points
  - Lower maximum error (162.82 basis points)
- Neural models (GRU and CNN-LSTM) show similar patterns:
  - Median errors around 3.67 basis points
  - 75% of predictions within 6.86 basis points
  - Slightly higher maximum errors (169 basis points)
- Random Forest performance:
  - Median error: 3.58 basis points
  - 75% of predictions within 6.70 basis points
  - Maximum error comparable to LightGBM

5.3 Stock-Specific Performance

Analysis across 200 different stocks reveals:

- LightGBM achieves the lowest mean stock-specific MAE (5.12)
- All models show consistent performance patterns:
  - Best-case MAE: 2.62-2.72 basis points
  - Median stock MAE: 4.69-4.77 basis points
  - Worst-case MAE: 12.83-13.43 basis points
- Standard deviation of stock-specific MAE:
  - LightGBM: 1.90 (most consistent)
  - Random Forest: 1.93
  - Neural Models: 1.97 (slightly more variable)

5.4 Key Findings

- Tree-based models (LightGBM, Random Forest) outperform neural architectures
- LightGBM shows superior performance in both accuracy and consistency
- Neural models demonstrate nearly identical performance characteristics
- All models maintain reasonable error bounds across different stocks
- Models show robust performance across various market conditions

The results suggest that while all models provide reasonable predictions, the tree-based approaches, particularly LightGBM, are more effective at capturing the complex relationships in market microstructure data. The similar performance of neural architectures indicates potential redundancy in their modeling approaches for this specific prediction task.