JPMC 1
Market Maven

Fall 2024 AI Studio

Simona Isakova, Jasmine Khalil, Carly Kiang,
Zoe Krishtul, Samhitha Sangaraju

BREAK THROUGH TECH

JPMorgan Chase & Co.

Student Team

Samhitha Sangaraju
BS, Computer Science and Data Science – Rutgers University

Jasmine Khalil
BS, Electrical Engineering – Penn State University

Carly Kiang
BS, Computer Science – Columbia University

Simona Isakova
BS, Computer Science – CCNY

Zoe Krishtul
BS, Computer Science and Cognitive Science – University of Central Florida

BREAK THROUGH TECH

JPMorgan Chase & Co.

Challenge Advisors and TA

Seoyoung Kyung - Applied AI/ML Lead at JPMorgan Chase & Co

Jatindeep Singh - Applied AI/ML Lead at JPMorgan Chase & Co

Shrey Verma - Master's Degree Student at Cornell Tech

Challenge Summary

Predicting closing price movements for Nasdaq listed stocks during the critical final ten minutes of trading -- Heightened volatility and rapid price fluctuations, which significantly impact the day's economic outcomes.

Real-World Application: Mirrors real-world challenges faced by financial professionals who must make quick, data-driven decisions under pressure.

BREAK THROUGH TECH

JPMorgan Chase & Co.

Why Use Machine Learning and Business Context

Machine learning algorithms are great at uncovering **intricate patterns** and **relationships** between data -- **enhanced accuracy** which is very useful when forecasting future trends.

- Financial institutions and tech companies are driving AI/ML integration, algorithmic trading, and ESG-focused investing in FinTech.
- As a global finance leader, JP Morgan emphasizes innovation, market leadership, and risk management, guided by integrity, responsibility, and client-centricity.

Market Efficiency, Price accuracy, and market accessibility during volatile periods -- Maintains trust and stability in financial markets.

Project Goals

Supervised learning focused on regression: Specifically, predict the future price movements of stocks relative to a synthetic index, based on historical data from the order book and closing auction.

- **Understanding Data (Exploratory Data Analysis):** Understanding Target Function and Components of Target Function (Stock Return V/S Index Return)

$$Target = \left(\frac{StockWAP_{t+60}}{StockWAP_t} - \frac{IndexWAP_{t+60}}{IndexWAP_t}\right) * 10^4$$

- **Feature Engineering and Correlation Analysis:** Understanding Calculated Features (Rolling Average, Volatility) and their effect on Components of Target Function.
- **Developing Regression Models :** Build and evaluate models to predict the target. Implement models: LightGBM, GRU, Random Forest, XGBOOST, Catboost to highlight the best-performing model using MAE comparison.

Introducing the Dataset

- **Data Background:** The dataset focuses on the high-stakes last 10-minute closing auction on Nasdaq, where volatility peaks as prices settle. These prices are refined using order book and auction data.
- **Data Source and Description:** Provided by Kaggle, this dataset includes historical auction data, with a synthetic index based on Nasdaq-listed stocks, aimed at understanding price movements during the market close
- **Features Included:** Numeric features cover stock identifiers, time markers, price and volume metrics, bid-ask spread, imbalance indicators etc., all essential for analyzing auction dynamics and closing price behavior. We are predicting the “target” column with regression modeling.

	volume_weighted_price	reference_price	wap	near_price	stock_id	date_id	seconds_in_bucket	imbalance_size	imbalance_buy_sell_flag	matched_size	...	row_id	bid_ref_price_diff
0	5.663061e-08	0.999812	1.0	1.000241	0	0	0	3.180603e+06	1	13380277.00	...	0_0_0	0.000000
1	6.951083e-09	0.999896	1.0	1.000241	1	0	0	1.666039e+05	-1	1642214.25	...	0_0_1	0.000000
2	7.698351e-09	0.999561	1.0	1.000241	2	0	0	3.028799e+05	-1	1819368.00	...	0_0_2	-0.000158
3	7.786061e-08	1.000171	1.0	1.000241	3	0	0	1.191768e+07	-1	18389746.00	...	0_0_3	-0.000172
4	7.557200e-08	0.999532	1.0	1.000241	4	0	0	4.475500e+05	-1	17860614.00	...	0_0_4	-0.000138

bid_ref_price_ratio	ref_price_ma_5	price_momentum	bid_size_volume_ratio	imbalance_volume_interaction	day_of_week	hour_of_day	price_volatility
1.000000	0.999794	0.000084	0.004533	4.255735e+13	0	0	0.000263
1.000000	0.999794	0.000084	0.001969	2.735993e+11	0	0	0.000263
0.999842	0.999794	-0.000335	0.020862	5.510500e+11	0	0	0.000263
0.999828	0.999794	0.000610	0.000126	2.191631e+14	0	0	0.000263
0.999862	0.999794	-0.000639	0.000923	7.993517e+12	0	0	0.000263

BREAK THROUGH TECH

JPMorgan Chase & Co.

Cleaning the Data: Interpolation, Imputation

First Method: -1 Imputation

- Replaced all NaN values with **-1** to indicate that the element was missing when data was collected.

	MAE Train	MAE Test
Model with -1 Imputation	XXX	XXX
Model with linear interpolation	XXX	XXX

Second Method: Linear Interpolation

- Used a **rolling average** for all columns but columns with **first element missing** required an alternate approach.
- Correlation between this column and all remaining numeric columns.
- Linear interpolation using column with **highest correlation value.**

BREAK THROUGH TECH

JPMorgan Chase & Co.

Dataset Story/Analysis

- **Predicting Price Trends:** It anticipates price trends during high volatility, helping with strategies like arbitrage and risk management

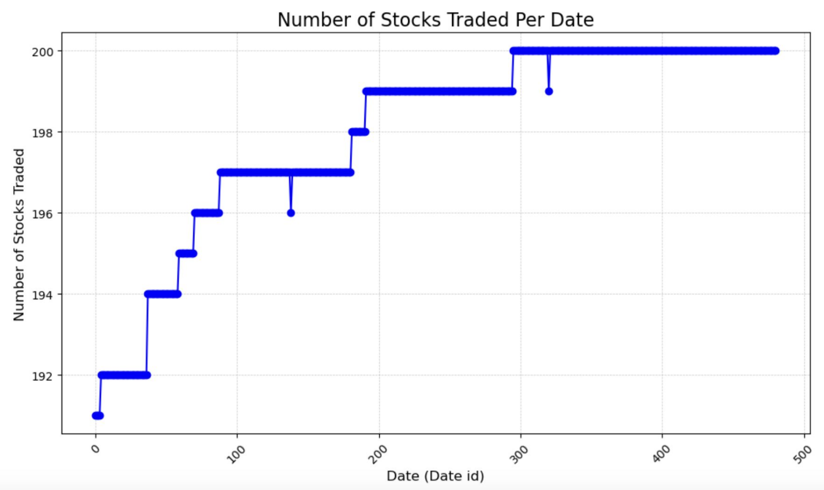
- **Pattern Insights:** The model reveals trends in volume, volatility, and momentum, offering a broader view of market behavior beyond closing prices.
  - Investigating why the MAE values are different for certain days…were there any impactful societal happenings?

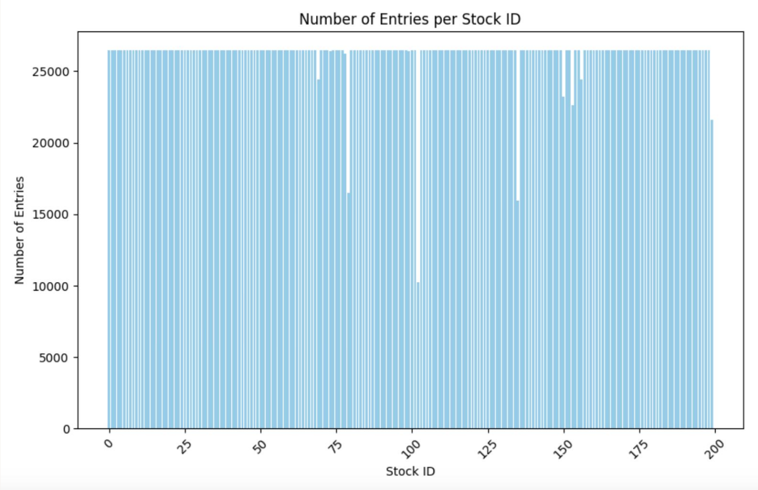
- Our modeling provides greater clarity behind the last-minute price movements, enabling data-driven decisions that can improve market predictions, fairness, and trader confidence

Dataset Story/Analysis

Analyzing Volume

1. How does the number of stocks traded each day vary?
2. Are there differences in the number of trades for each stock id?





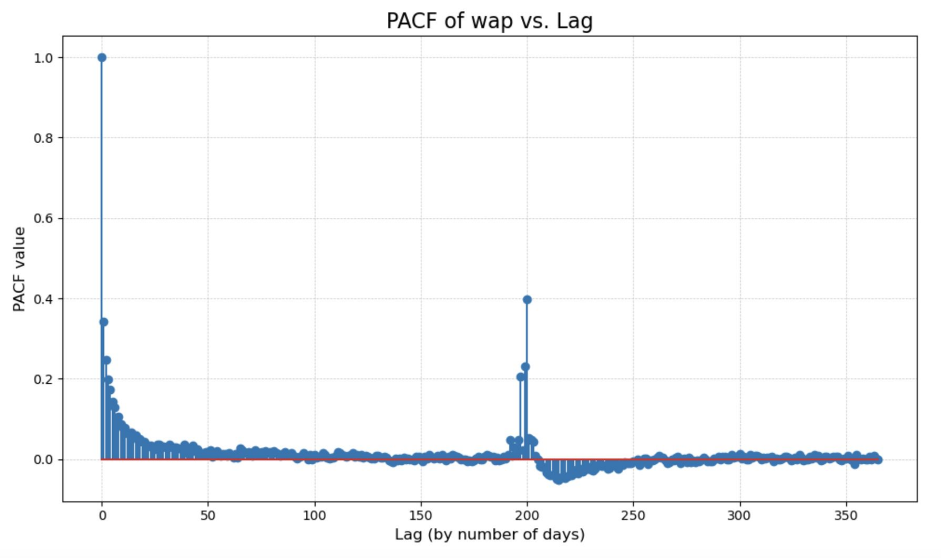
Later date id's had a higher number of stocks traded

Stock ID with the minimum count: 102
Count: 10230

Dataset Story/Analysis

Analyzing Time Series:

- Are there particular lags that are significant?
- Data spans across ~480 days, plotted partial autocorrelation across 365 days (1 year)
- A Partial Autocorrelation Function (PACF) plot highlights significant lags in WAP, indicating potential relevance to WAP.



The most significant lag is 200 days with a PACF value of 0.3981593808380232

Feature Engineering

New Features: Times Series

- **day_of_week**

  As explained later, day_of_week has a relatively significant correlation with the Target function.

- **hour_of_day**

  On the other hand, hour_of_day does not have a relatively significant correlation with the Target function.

```
data['bid_ref_price_diff'] = data['bid_price'] - data['reference_price']
data['bid_ref_price_ratio'] = data['bid_price'] / data['reference_price']
data['ref_price_ma_5'] = data['reference_price'].rolling(window=5).mean()
data['price_momentum'] = data['reference_price'].diff()
data['volume_weighted_price'] = (data['reference_price'] * data['matched_size']) / data['matched_size'].sum()
data['bid_size_volume_ratio'] = data['bid_size'] / data['matched_size']
data['imbalance_volume_interaction'] = data['imbalance_size'] * data['matched_size']
data['day_of_week'] = data['date_id'] % 7
data['hour_of_day'] = (data['seconds_in_bucket'] // 3600) % 24
data['price_volatility'] = data['reference_price'].rolling(window=5).std()
```

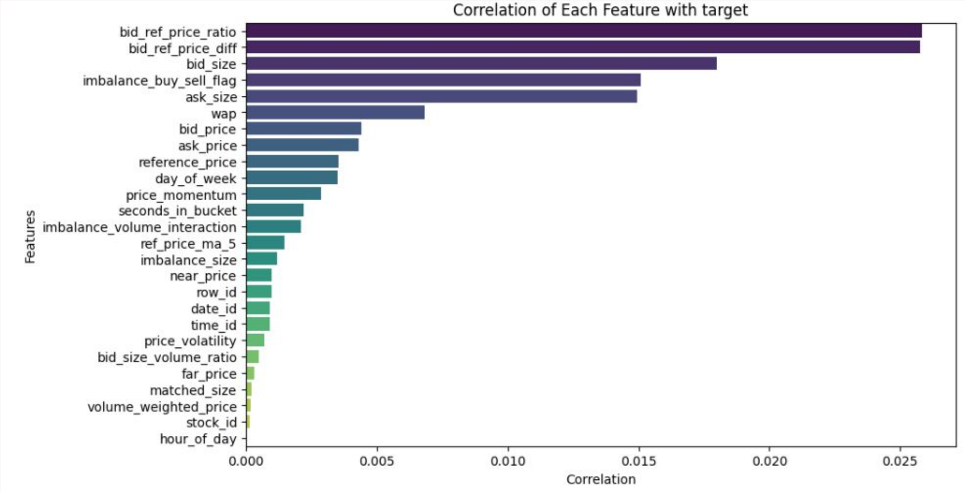
BREAK THROUGH TECH

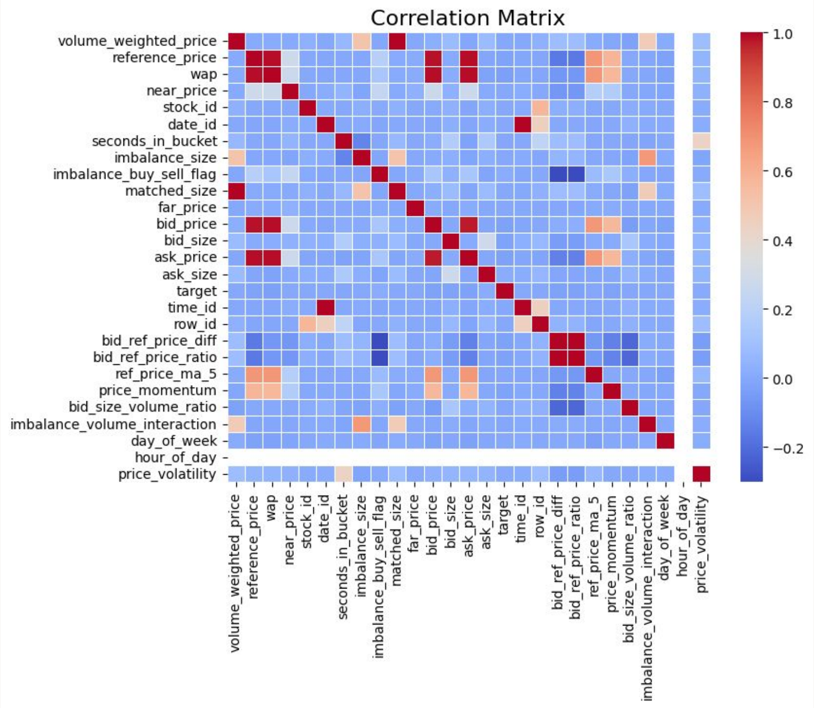
JPMorgan Chase & Co.

Feature Selection & Importance

Currently, we have kept all of our features because we did not add a large number of new features to the dataset.

However, we still explored **feature importance** between all columns and the Target column as well as a correlation matrix to have an idea of the **impact** each feature has on the outcome of the Target as well as each other.





Modeling Results

- LightGBM
- XGboost
- Catboost
- Random Forest
- GRU

	Mean Abs. Error: Train	Mean Abs. Error: Test
LightGBM linear interpolation	6.287039	6.301912
LightGBM -1 Imputation	6.286259	6.300450
XGBoost linear interpolation	6.104001	6.188764
XGBoost -1 imputation	6.111118	6.193579
Catboost linear interpolation	6.503802	6.046937
Catboost -1 Imputation	6.503802	6.046937
Random Forest linear interpolation	6.828563	6.732851
Random Forest -1 imputation	6.842371	6.827611
GRU linear interpolation	6.5529701	6.203946
GRU -1 imputation	6.5508523	6.201512

GRU Experimentation

- Initially used 1 GRU layer for results shown previously (simple model)
- Hypothesis: Would using multiple layers result in a lower MAE score?

GRU -1 imputation	**Mean Abs. Error Train**	**Mean Abs. Error Test**
1 layer	6.5508523	6.201512
5 layers	6.5478053	6.198259

- The reduction in MAE was minimal and did not justify the significant increase in computational resource requirements

Pros and Cons of Best Performing Models

- Catboost and XG Boost

Catboost Pros:

- Effective handling of categorical features
- Handles imbalanced data well
- Avoids overfitting
- Robust against noisy data

Catboost Cons:

- Resource intensive
- Longer training time on CPU
- Feature interaction complexity
- May not be optimal for pure numerical data

XGBoost Pros:

- High accuracy
- Efficient and scalable
- Versatile for various tasks
- Handles missing data effectively

XGBoost Cons:

- Complex tuning
- Resource-intensive
- Prone to overfitting
- Low interpretability

Ensemble Model Results

- **LightGBM**
- **Catboost**
- **XGboost**
- **GRU**
- **Random Forest**

	Mean Abs. Error Train	Mean Abs. Error Test
-1 Imputation	6.0702	6.1962
Linear Interpolation	6.0587	6.1910

Conclusions

- **Best performing models:** XGBoost, Catboost
- The **ensemble model** also performed very well comparatively

BREAK THROUGH TECH

JPMorgan Chase & Co.

Thank You!
Any Questions?

BREAK THROUGH TECH

JPMorgan Chase & Co.