Predicting Stock Market Signals

Tree-based Classifiers Using Technical Indicators

Team 12B
Vy Nguyen, Dennis Wu, Kenjee Koh
Hsiang-Han Huang, Becky Wang

Table of contents

I

Project & Dataset Overview

What defines a good investing?

1. Project Overview

What defines good investing?

- Earning $10,000 annual return sounds great!
- 20% return annually on $10M ($20,000) seems impressive!
- 2024 S&P 500 YTD is 26.84%

How to keep up with the market?

- Passive investing is simple and effective for most investors

Humans often want more!

This leads us to two enduring questions:

"How can we outperform the market?"

"As individual investors, can we leverage data analytics on open-source data to achieve market outperformance?"

1. Project Overview

Traditional active investing strategies:

- Stock Selection: Picking outperforming stocks
- Market Timing: Deciding when to buy or sell

Approaches:

- Fundamental Analysis: Evaluating the intrinsic value of a company
- Technical Analysis: Using price/volume patterns to predict trends

Even Wall Street fund managers often underperform benchmarks:

- 85% of active funds underperform the S&P 500 over a decade

Can Machine Learning be the game changer?

1. Project Overview

What Are We Doing?

- Instead of buy-and-hold or investing at fixed intervals, we aim to optimize the timing of our investments
- Developing a machine learning model that can identify the best times to buy a stock

Technical Analysis:

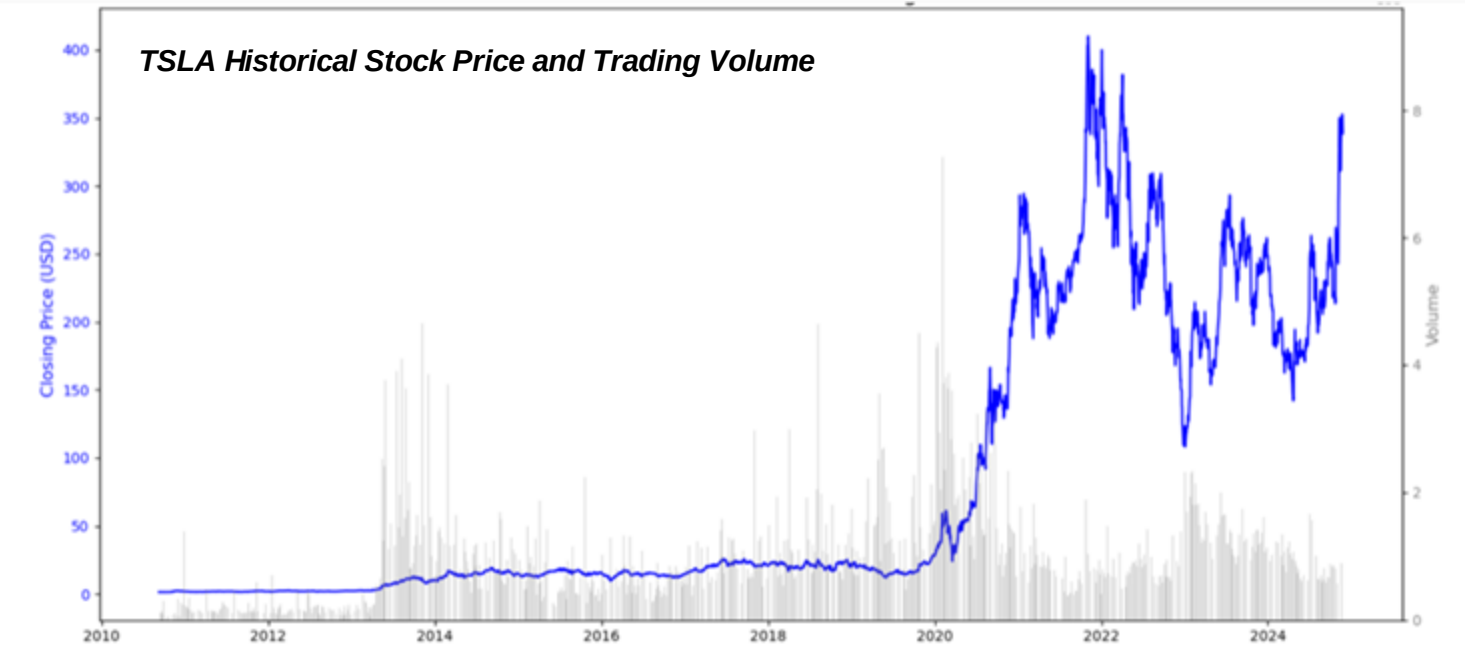
- Accessible price/volume data
- Daily or intraday signals

Market Timing: We use Tesla's historical stock data to train our model and classify each trading day into two categories:

- Up Opportunity: Good time to buy the stock
- No opportunity: Otherwise

1. Project Overview

Tesla(TSLA) as Our Model Training Example



2. Dataset Overview

Data Source	yfinance Python library
Description	Daily historical data
Key Features	Date/Open/High/Low/Close/Volume/Dividends/Stock Splits
Size	3632 data points spanning more than 14 years
Data Type	Numerical/Temporal

II

Variable Engineering

1. Target Variable - Signal

Binary variable (1: Up opportunity, 0: No opportunity)

Step 1: Future Return

$$\text{Future Return}_t = \frac{\text{Close}_{t+5} - \text{Close}_t}{\text{Close}_t} \times 100$$

Step 2: Threshold for "Up Opportunity"

Threshold = Average future return x 1.5

Step 3: Signal

$$\text{Signal}_t = \begin{cases} 1 & \text{if Future Return}_t \geq \text{Threshold}_{\text{up}} \\ 0 & \text{otherwise} \end{cases}$$

Step 4: *Drop rows where we cannot calculate Future_Return due to shifting*

Purposes

- Simplify modeling into a binary classification problem
- Identify **"Up Opportunity"** based on future stock price performance
  - Select 5 future days for Signal due to active short-term trading strategies

2. Features

Purposes of creating 6 features:
Daily_Returns, Volume_Change, Volatility, MA_Ratio, RSI, and Day_of_Week

- Incorporate historical patterns (e.g., moving averages, volatility) for lagging indicators.
- Leverage momentum-based measures like RSI for predicting future movements.
- Add contextual information, such as trading activity (volume) and weekday trends, to improve model performance.

Rationale for choosing from 5 to 14 rolling windows for engineered features:

- Widely used in technical analysis to capture short-term trends while smoothing out daily noise
- Align with the short-term active trading goals
- However, reduced data points due to dropping N/A rows from rolling windows

2. Features

Feature	Definition	Data Type	Formula
1. Daily_Returns	% Daily change in stock price	Continuous (%)	$\text{Daily Returns}_t = \frac{\text{Close}_t - \text{Close}_{t-1}}{\text{Close}_{t-1}} \times 100$
2. Volume_Change	Change in volume as a % of the 10-day rolling average	Continuous (%)	**10-Day Rolling Average of Volume:** $\text{Volume}_{MA10} = \frac{\sum_{i=t-9}^{t} \text{Volume}_i}{10}$ **Volume Change:** $\text{Volume Change}_t = \frac{\text{Volume}_t - \text{Volume}_{MA10}}{\text{Volume}_{MA10}} \times 100$
3. Volatility	Standard deviation of daily returns over a 10-day window	Continuous	$\text{Volatility}_t = \sqrt{\frac{\sum_{i=t-9}^{t}(\text{Daily Returns}_i - \overline{\text{Daily Returns}})^2}{10}}$

2. Features

Feature	Definition	Data Type	Formula
4. MA_Ratio	Ratio of 10-day to 50-day moving averages on price	Continuous	$\text{MA Ratio}_t = \frac{\text{MA}_{\text{Short},t}}{\text{MA}_{\text{Long},t}}$
5. RSI (Relative Strength Index)	Momentum indicator: Measures the magnitude of price gains vs losses over 14-day window	Continuous	**Relative Strength (RS):** $\text{RS}_t = \frac{\text{Gain}_t}{\text{Loss}_t}$ **RSI Formula:** $\text{RSI}_t = 100 - \frac{100}{1 + \text{RS}_t}$
6. Day of Week	Monday,..., Sunday	Categorical	Extracted from the Date index -> Create dummies

Final Data Description

1. Final Dataset

Original dataset
3632 data points

preprocessing
& engineering

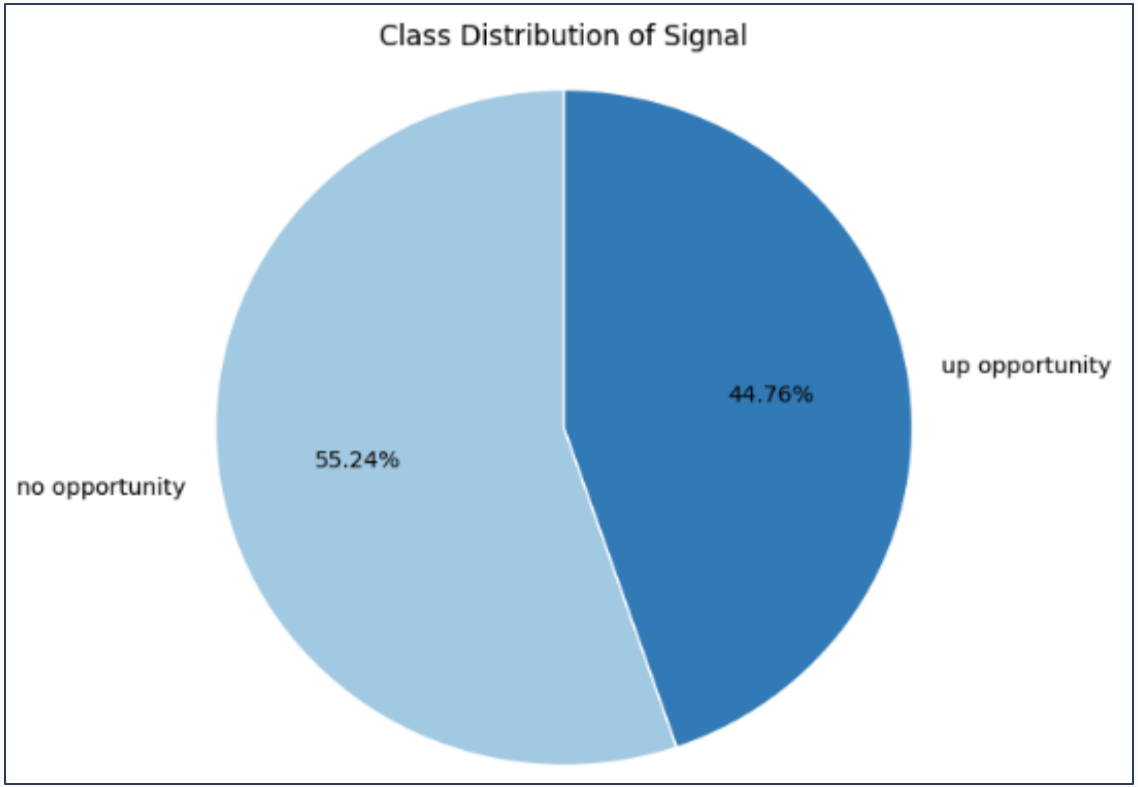
Final dataset
3577 data points

Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
2010-06-29 00:00:00-04:00	1.2667	1.6667	1.1693	1.5927	281494500	0.0000	0.0000
2010-06-30 00:00:00-04:00	1.7193	2.0280	1.5533	1.5887	257806500	0.0000	0.0000
2010-07-01 00:00:00-04:00	1.6667	1.7280	1.3513	1.4640	123282000	0.0000	0.0000

Date	Daily_Returns	Volume_Change	Volatility	RSI	MA_Ratio	Day_of_Week	Signal
2010-09-08 00:00:00-04:00	1.7527	-31.3170	2.4368	67.1497	1.0220	Wednesday	1
2010-09-09 00:00:00-04:00	-0.9090	-7.6107	2.2829	65.0470	1.0294	Thursday	0
2010-09-10 00:00:00-04:00	-2.6074	-3.9432	2.4576	58.0938	1.0353	Friday	0

2. Class Distribution

Relatively balanced class distribution => **Robust for classifier models**



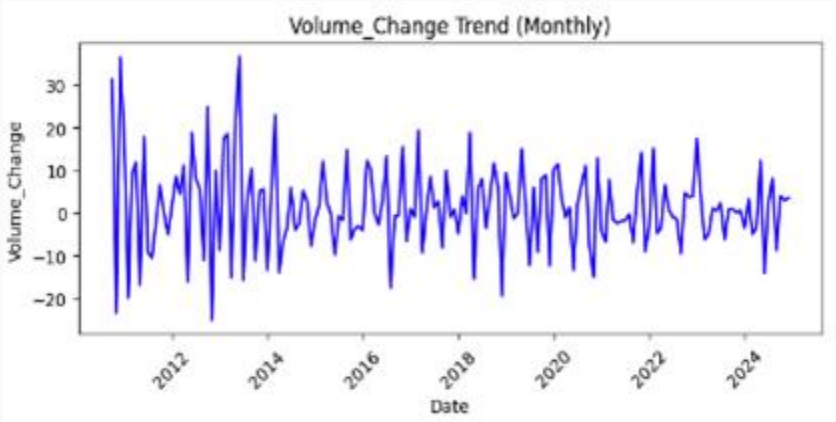
3. Feature Description

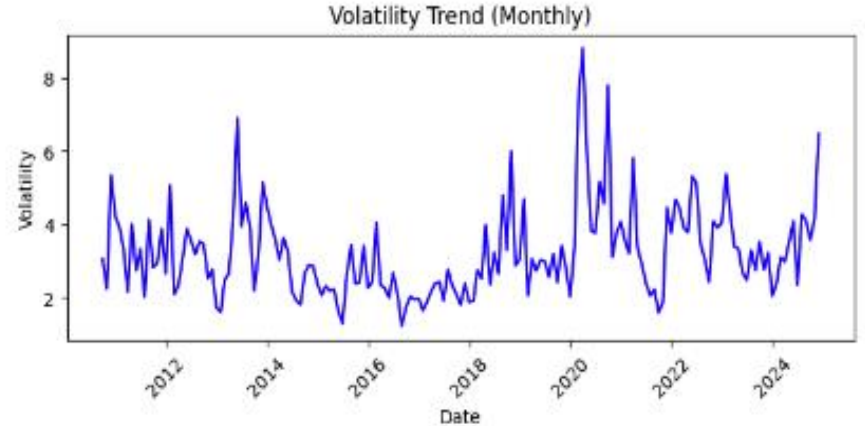
Continuous Features	Mean	Std	Min	Max
1. Daily_Returns (%)	0.2193	3.5918	-21.0628	24.3951
2. Volume_Change (%)	1.4741	45.0526	-83.2239	551.0873
3. Volatility	3.2242	1.5674	0.6531	12.4635
4. MA_Ratio	1.0325	0.1253	0.6763	1.6224
5. RSI	53.2619	17.8167	5.1392	97.5299

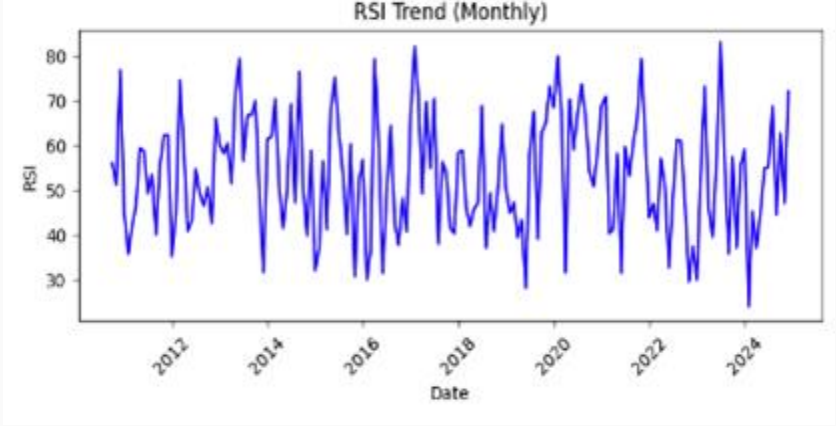
Categorical Feature	Mean
Tuesday	734
Wednesday	734
Thursday	723
Friday	719
Monday	668

Volume_Change has extreme outliers => **Use tree-based models**

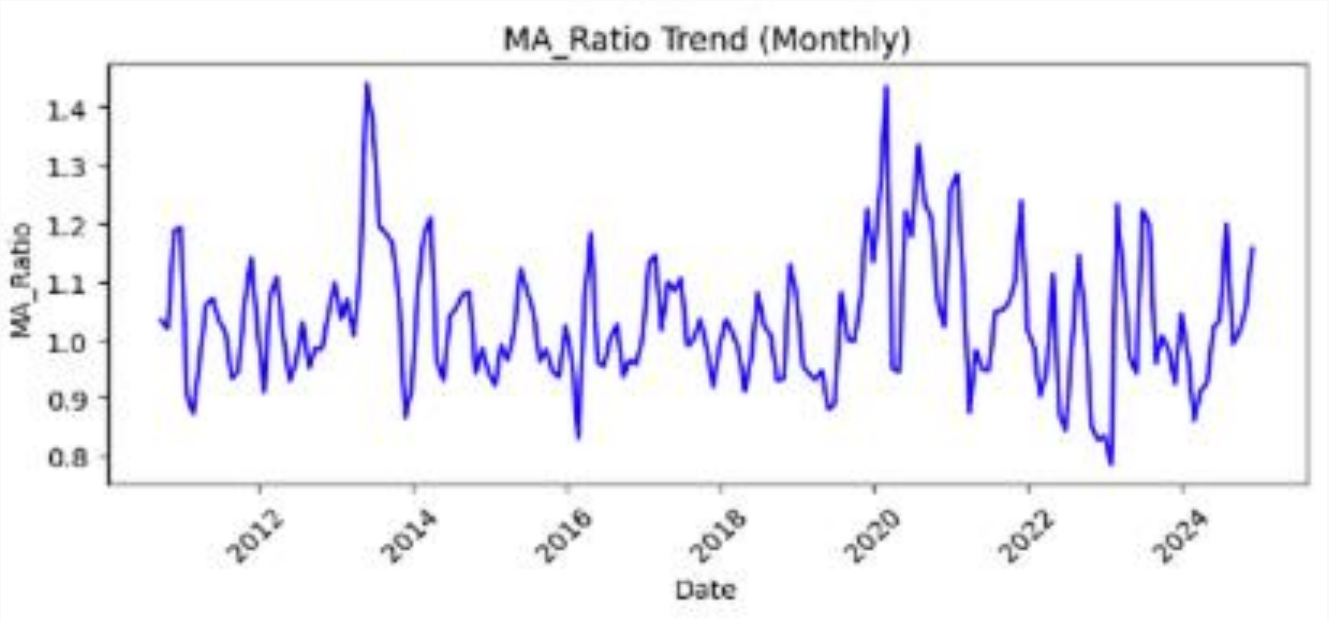

Daily_Returns Trend (Monthly)
Daily_Returns
Date


Volume_Change Trend (Monthly)
Volume_Change
Date

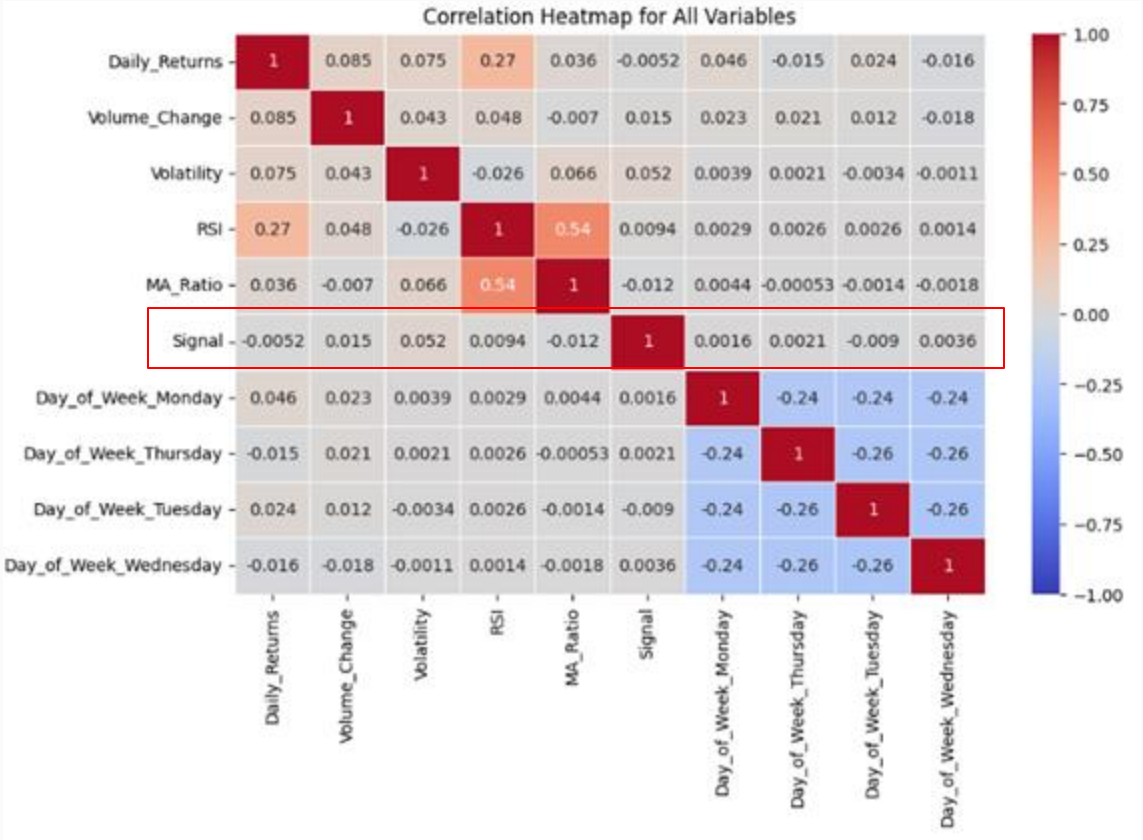

Volatility Trend (Monthly)
Volatility
Date


RSI Trend (Monthly)
RSI
Date

3. Feature Description



4. Correlation Matrix



Low correlation between the target variable and features suggests **non-linear** relationship
=> Use tree-based models

IV

Modeling & Evaluation

1. Model Building

- **Select classifier models**

1. ***Decision Tree***
2. ***Random Forest***
3. ***Gradient Boosting***

To better capture non-linear relationships, handle both numerical and categorical data, and outliers

- **Hyperparameter Tuning**

RandomizedSearchCV
Faster and more efficient than grid search

TimeSeriesSplit(n_splits=5)
Simulate real-world forecasting by preserving time order

- **Split train & test sets (70/30)**

1. ***Random split using train_test_split***
   Class distribution in test: 593/481 => relatively balanced

1. ***Chronological split (older data for train & more recent data for test)***
   Class distribution in test: 617/457 => relatively balanced

How well the model generalizes on both random and time-ordered splits?

2. Performance Summary

Test Accuracy (%)	Decision Tree	Random Forest	Gradient Boosting
Random split	56.33	55.96	59.03
Random split + tuning	57.54	57.26	58.84
Chronological split	49.53	54.38	55.95
Chronological split + tuning	57.17	56.61	57.08

Random split may learn a broader range of patterns -> better generalize to changing dynamics of stocks

2. Performance Summary

Class 1 Accuracy (%)	Decision Tree	Random Forest	Gradient Boosting
Random split	54	42	36
Random split + tuning	25	43	36
Chronological split	56	44	43
Chronological split + tuning	16	52	7

Chronological split is more realistic for forecasting "Up Opportunity" in stock prices.

Which model to choose?

One with highest accuracy or with highest stratified accuracy?

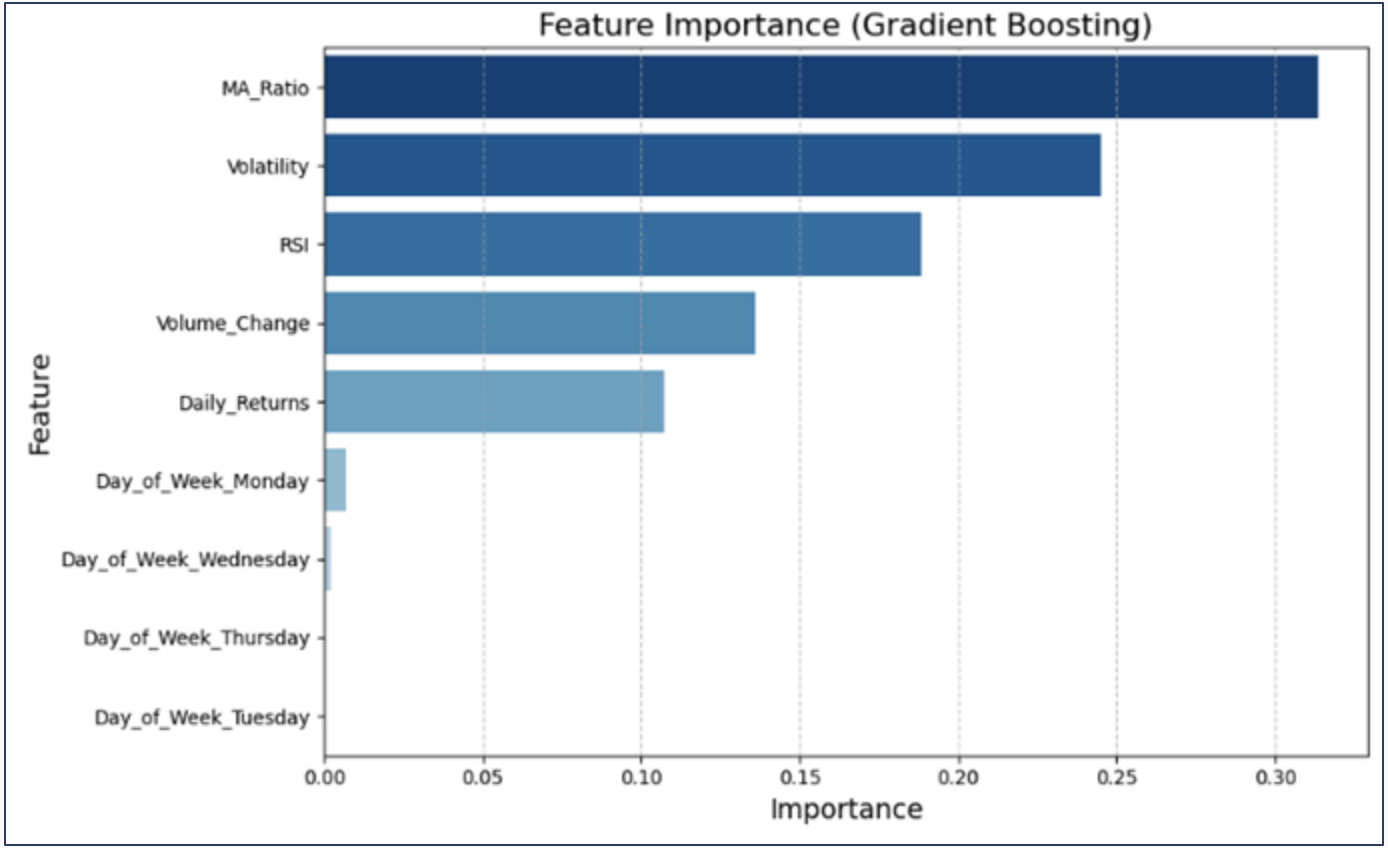
3. Final Model

Financial markets are unpredictable
The cost of investing in opportunities with low or negative returns (Class 0) is much more impactful than missing a potential gain (Class 1)

Gradient Boosting - highest overall accuracy(random split, **59.03%**) with **57%** precision & **36%** stratified accuracy for Class 1

Use to capture opportunities where the price return in the next 5 days **exceeds half** the average future return

4. Feature Importance



V

Limitations & Recommendations

Limitations & Recommendations

All Engineered Variables

Bias in indicator selection and rolling window parameters may limit the model's objectivity and generalizability. Lagging indicators also cause data loss, reducing the dataset size and potentially impacting model performance.

Not Robust for Temporal Pattern

Gradient Boosting Classifier does not capture sequential or time-dependent relationships in the data. Advanced time-series models like Long Short-Term Memory (LSTM) networks or other recurrent neural networks could better identify temporal patterns.

Not Accounting for More Variables

The model lacks macroeconomic indicators, fundamental metrics (e.g., earnings, P/E ratios), news events, and sector-specific trends, limiting its ability to capture broader market dynamics and real-time or industry-specific influences.

VI

Conclusion

Conclusion

Indicator impacts on stock return

- Lagging indicator (MA Ratio) , price fluctuation (Volatility) and leading indicator (RSI) are most influential to short-term price forecasting
- No day-specific effects on stock returns

Role of the model in investment decisions

- The model should not be the sole basis for investment decisions but a complementary tool to identify higher return opportunities.
- No model can perfectly predict stock prices, but incremental improvements, like our 59% accuracy rate, are a step toward informed decision-making.

Scenario Analysis:

1. **Average Future Return (5 days):** 2%
2. **Threshold for "Up Opportunity":** 3% (1.5 × Average Future Return)
1. **Prediction Accuracy:** 59%
2. **Up Opportunity Days:** 45% of 252 trading days (approximately 114 trading days per year)
1. **Capital Allocation Per Trade:** Entire portfolio (compounding gains)

28405%

If All Predictions Are Correct

394%

With 59% Prediction Accuracy

10%

S&P 500 Average Annual Return

Thank You!