Stock Direction Predictor

We built this project around a simple question: can information about a stock and the wider market help us guess whether the stock will go up or down the next day? We use past market data and a machine-learning model to test this idea.

The model gives a chance that the stock will go UP. It then turns that chance into an UP or DOWN prediction. Most importantly, we test the model on a later year that it has never seen before. The Quant Society website shows these results in an interactive dashboard.

Reference experiment used in this report

Input	Value	Why it matters
Data source	Yahoo Finance	Where the daily market data comes from
Historical window	2016-01-01 to 2026-01-03	Gives several years for training and testing
Training period	2016-2022	Used to teach the models
Validation period	2023-2024	Used to choose the model and settings
Final test	2025	The main year the model had not seen before
Test observations	1,750	Number of 2025 observations in the seven-stock group
Random state	42	Keeps the experiment repeatable

1. The question behind the model

The goal is kept simple. For each stock and each day, the model looks only at information available by the end of that day. The answer is 1 if the stock's adjusted closing price is higher the next day, and 0 if it is not. We remove the final day because we do not know what happens after it.

This setup is important because it is easy to accidentally give a model information from the future. We also do not randomly mix old and new data. The model learns from earlier years, is checked on the next two years, and is finally tested on a later year.

What the current run actually shows

In the 2025 test, logistic regression reached 54.3% accuracy. The simple always-UP guess reached 53.7%. So the model was only 0.6 percentage points better. The project report says this difference is not statistically clear. Random forest did worse, with 47.2% accuracy.

2. How the project is put together

The project is split into a few clear parts: getting the data, creating useful measurements, training the models, checking the results, and showing everything on the website. This makes the project easier to test and repeat.

File / layer	What it does	Why it is separate
data.py	Downloads and checks daily stock data.	Keeps the data step clear and testable.
features.py	Creates returns, ranges, moving averages, RSI, volatility, volume and market features.	Turns raw prices into the model inputs.
train.py	Splits the data by time, compares models, trains them and chooses the probability cut-off.	Keeps model selection separate from reporting.
evaluate.py	Creates predictions, scores, probability checks and feature tables.	Turns model results into useful evidence.
backtest.py	Uses the UP probability cut-off in a simple one-day trading example and includes a trading cost.	Shows how predictions could be used in a basic strategy.
pipeline.py	Runs the full process and saves the files used by the dashboard.	Makes the experiment easy to repeat.
dashboard/app.py	Runs the Streamlit and Plotly website.	Keeps the website separate from the research code.

The experiment flow

- 1. Download daily stock data for seven companies and SPY, then check that the data is clean.
- 2. Create features using only information known by the end of each day, then create the next-day UP/DOWN answer.
- 3. Train on 2016-2022 and use 2023-2024 to choose the model settings and the probability cut-off.
- 4. Train the final models again using all data before 2025, then test them once on 2025.
- 5. Save the predictions, scores, probability checks, and feature information for the dashboard.

Why the timeline matters

We compare the models year by year during testing so that one unusual market period does not decide everything. The probability cut-off is also chosen using only the 2023-2024 validation period, not the final 2025 test.

The trading example

The project also includes a simple example of how the predictions could be used. A stock is held for one day only when its predicted chance of going UP is high enough. When the position changes, the model assumes a small trading cost of 0.05%. This is only an example, not a full trading system.

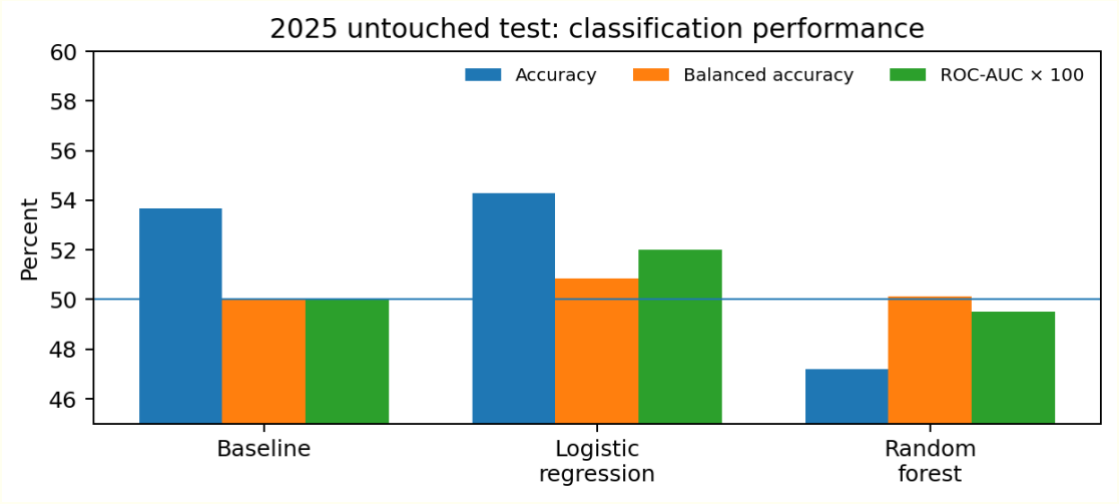

Figure 1. Comparison of the models on the 2025 test data. Percentages are shown as normal percentages.

3. What information does the model use?

The model looks at several types of information instead of just one momentum signal. These include recent price changes, moving averages, volatility, trading volume, the overall market, how the stock compares with the market, and a small amount of calendar information.

Feature family	Examples	What it looks at
Price / return	Overnight return, intraday return, 1d / 2d / 5d / 10d / 20d returns	Recent price movement
Range / volatility	Daily range, 5d volatility, 20d volatility, volatility ratio	How much the stock has been moving
Trend	Price / MA(5), price / MA(20), MA(5) / MA(20), RSI(14)	Recent trend and momentum
Volume	Volume / 20d average, 1d volume change	Unusually high or low trading activity
Market context	SPY 1d / 5d return, SPY 20d volatility, SPY price / MA(200)	What the overall market is doing
Relative / cross-sectional	Excess return versus market, daily return / RSI / volatility ranks	How the stock compares with the market and other stocks
Calendar	Day-of-week sine and cosine	Small weekly patterns

The model choices

Model	Role	Implementation detail
Majority baseline	Simple comparison	Always predicts the most common class.
Logistic regression	Main simple model	StandardScaler + LogisticRegression, C = 1.0, max_iter = 2000.
Random forest	More complex tree model	Uses several settings and gives more weight to the less common class.
Extra Trees	Another tree model tested during development	Compared during development; the final test table reports Random Forest.

We do not treat a feature's importance as proof that it causes stock prices to move. In the Random Forest results, market-related measurements were among the most important, especially the market's one-day return, the stock's price compared with its 200-day average, market volatility, and the market's five-day return. This suggests the model paid a lot of attention to what the wider market was doing.

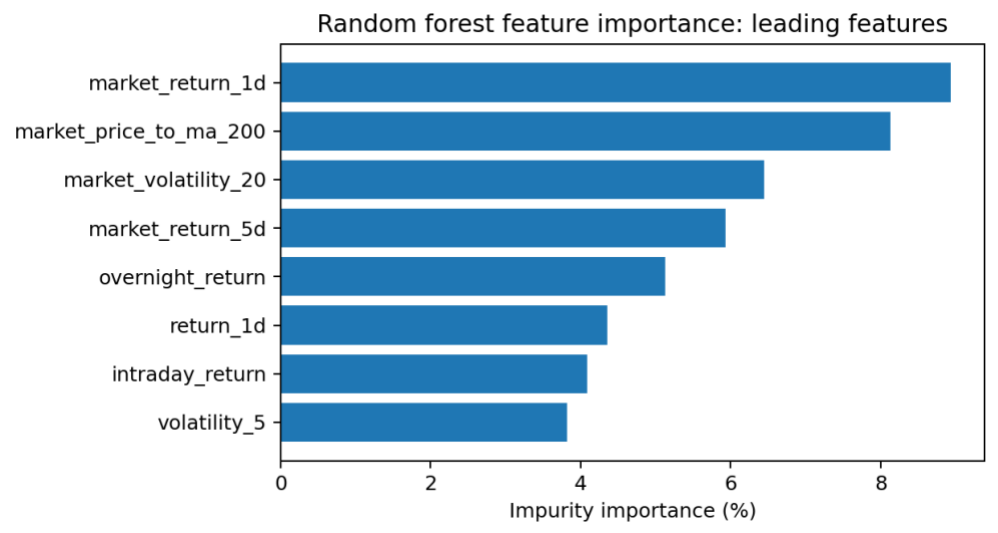

Figure 2. The most important Random Forest features. These show what the model used most, not what causes the market to move.

What the logistic regression model suggests

For logistic regression, some of the strongest model relationships involved five-day volatility, the change in volatility, the stock price compared with its five-day average, the two-day return, the market's one-day return, and the stock's position compared with the market's 200-day average. These results describe this particular experiment. They should not be treated as universal trading rules.

4. How we tried to make the test fair

Stock prediction can look much better than it really is if future information slips into the data or if the test is set up poorly. We therefore make the timing of the data a key part of the project.

Check	Why it matters	Why it matters
Chronological split	Keeps future data out of earlier stages.	Prevents future observations from entering model development.
Feature timing	Makes sure each feature uses only information known that day.	Avoids using the future outcome to construct the predictors.
Target alignment	Makes sure the answer is the next trading-day direction.	Makes the prediction horizon explicit and testable.
Threshold selection	Chooses the probability cut-off without using 2025 results.	Prevents using 2025 labels to tune the final decision rule.
Final refit	Uses all allowed data before 2025 before the final test.	Uses all legitimate development data before the untouched test year.

5. Reference run: what happened

The final test has 1,750 observations from AAPL, MSFT, AMZN, META, NVDA, JNJ and WMT during 2025. These seven stocks were chosen after looking at the original 2025 results. Because the choice was made after seeing the test period, this group should be treated as a description of the results, not as a brand-new fair test.

Model	Accuracy	Balanced acc.	Precision	Recall	F1	ROC-AUC
Majority baseline	53.7%	50.0%	53.7%	100.0%	69.8%	50.0%
Logistic regression	54.3%	50.9%	54.1%	97.8%	69.7%	52.0%
Random forest	47.2%	50.1%	54.3%	10.1%	17.1%	49.5%

How I read the result

The main point is not that the model can accurately predict stocks. It cannot do that with high accuracy in this test. Logistic regression reached 54.3% accuracy compared with 53.7% for the simple always-UP guess. Its ROC-AUC was about 0.52, which is only slightly above a random guess. The reliability report also found that the small improvement was not statistically clear.

Random Forest gives another useful lesson. Its 47.2% accuracy and 49.5% ROC-AUC were worse than logistic regression. In this experiment, making the model more complicated did not make it better.

A useful check: what does “better than the baseline” mean?

Because slightly more than half of the 2025 observations were UP days, an always-UP guess already gets 53.7% accuracy. So a useful model has to beat that simple guess by more than a small amount. That is why we also look at other scores, not just accuracy.

6. How we checked that the project was working

A model can produce numbers even when the code has a mistake. The project therefore includes checks for timing, target alignment, data order, and other common problems.

Check	What it protects
Next-day target alignment	Makes sure the label is the next trading-day direction.
Per-ticker rolling isolation	Stops moving calculations from mixing different stocks.
Feature timing	Checks that features only use information available on that day.
Chronological splits	Keeps training, validation and test in the correct time order.
Pipeline artifacts	Saves the features, predictions and reports used by the dashboard.

Choices that make the result more reliable

- The project downloads adjusted daily price and volume data from Yahoo Finance, then checks and organizes it before making the features.
- The 2025 test year is kept out of model selection and probability cut-off decisions.
- Validation results are averaged across calendar years so that one market period does not have too much influence.
- The model keeps its UP probabilities, so we can check whether those probabilities are sensible instead of only looking at UP/DOWN answers.
- The trading example includes a small transaction cost whenever the position changes, rather than assuming trading is free.

What the reliability report says

Model	Accuracy delta vs baseline	95% bootstrap interval	Stocks beating baseline	Months beating baseline	Statistically clear?
Logistic regression	+0.63 pp	-0.46 to +1.71 pp	3 / 7	7 / 12	No
Random forest	-6.46 pp	-12.23 to -0.51 pp	0 / 7	2 / 12	No

Overall, the project does something important: it shows the small improvement, the large uncertainty around that improvement, and the weaker result from Random Forest. It does not hide the disappointing results.

One important limitation of the test

The website's seven-stock group was chosen after looking at the 2025 results. That creates a selection problem. A stronger test would choose the stocks before seeing the final test results and then test them on a completely new period.

7. Limitations, next steps, and what we learned

This is a research project, not a live trading system or an investment recommendation. The most useful lessons come from understanding what the model can and cannot tell us.

The biggest assumptions

- Yahoo Finance is useful for a project like this, but it is not the same as using professional market data from a financial institution.
- The way adjusted prices are rebuilt is an approximation because the same adjustment is applied across the price data.
- There are only so many daily observations, and market conditions change. A relationship that appears in one period may disappear later.
- Many of the technical measurements are related to each other. A feature being important does not mean it causes the stock to move.
- The one-day trading example does not fully include costs such as slippage, taxes, market impact, liquidity limits, or borrowing costs.
- The seven-stock website group was chosen after seeing the 2025 results, so it should not be called a fresh out-of-sample test.

A realistic improvement plan

Priority	Improvement	Why it matters
1	Run a planned rolling / walk-forward test.	Tests the model over time without choosing stocks after seeing the results.
2	Use more stocks and add sector / market controls.	Checks whether the result works beyond the current stock group.
3	Focus more on probability quality.	A good model should give useful probabilities, not just UP/DOWN guesses.
4	Add more realistic trading costs.	Real costs could remove a small statistical advantage.
5	Compare with stronger simple benchmarks and time-series methods.	Checks whether the model beats credible alternatives.
6	Add more uncertainty and market-regime checks.	Helps tell a lasting pattern from a temporary one.

What we learned

The biggest lesson is about being honest with the results. Logistic regression found a small historical improvement, but the improvement is not statistically clear, and the more complex Random Forest model performed worse. So this project is better seen as a starting point for more research than as proof of a trading advantage.

The project is still useful even without a big accuracy number. It brings together public market data, clear features, time-based testing, probability predictions, reliability checks, and a simple trading example in one repeatable system.

Quant Society project page: https://quant-soc-website.vercel.app/analysis-momentum-strategy.html

GitHub repository: https://github.com/ManasVr09/QuantSocietyStockDirection