ISyE 6767 Interim Project 2 Report

Vidit Pokharna

November 24, 2025

Data Collection and Cleaning

The project uses historical market data from January 2000 to November 2021. Data was collected using a multi source approach to ensure reliability and consistency across a large universe of tickers.

The primary data source was Yahoo Finance accessed through the yfinance API. Previously downloaded files stored in the cache directory were reused to reduce API calls and ensure consistent data snapshots. Additional fallback CSV files in the data and data/stock_dfs directories were used whenever available.

The data cleaning pipeline standardized column names across inconsistent file formats by mapping variations such as Date, Datetime and date into a unified date field and Open or open into open. The date field was converted to pandas datetime and numeric fields were cast to float with error handling. The series was aligned to a business day frequency using asfreq("B") and missing observations were filled using a forward fill with a backward fill fallback.

Tickers with fewer than 400 observations after cleaning were excluded. The pipeline ensured the required columns existed for feature engineering. In cases where adjusted close was missing, it was recreated from the close price.

Several issues were identified during data preparation. Many tickers contained inconsistent column names which required normalization. Missing values were addressed through business day alignment and fill strategies. A more subtle issue occurred with QuantStats which raised errors when encountering unsupported frequency codes such as "YE". A custom patch was introduced to remap these codes into valid pandas frequency aliases and to enforce daily frequency during report generation. This fix eliminated all frequency related errors and allowed backtests to complete successfully.

Feature Engineering

The feature set combines short, medium and long horizon price and volatility information designed to capture diverse market conditions.

The engineered features include basic return measures such as one day percentage returns and five day log returns, volume z scores computed using a 20 day rolling mean and standard deviation, rolling statistics such as SMA, EMA, momentum and volatility measured across 5, 10, 21 and 63 day windows, and classic technical indicators including RSI with 14 periods, MACD with 12, 26 and 9 period windows, the MACD signal line and Bollinger Bands computed from a 20 day SMA with two standard deviation offsets.

The target variable is a binary indicator set to one if the next day's closing price exceeds the current close and zero otherwise.

These engineered features provided a moderate level of predictive power. Accuracy across most tickers ranged between 50 and 65 percent which indicates that the feature set captures at least a modest amount of return predictability. The variety of momentum, volatility and trend indicators proved useful although some redundancy exists due to correlated rolling window features. Even so, ensemble models are generally robust to such correlations.

Modeling Approach

Two ensemble learning approaches were used: Gradient Boosting and Random Forest. Both methods provide the ability to capture nonlinear relationships and interactions between features.

Gradient Boosting was selected due to its ability to build sequential models that correct earlier errors. To balance flexibility and generalization, the hyperparameter grid restricted tree depth to between two and three, used learning rates between 0.01 and 0.1 and used between 100 and 200 boosting stages. These choices help prevent overfitting given the relatively short histories available for many tickers.

Random Forest was selected for its stability and robustness to noise. The grid included 100 and 200 estimators, tree depths of three, five or unlimited and minimum leaf sizes of one, five or ten. These parameters regulate model complexity and help avoid deep tree overfitting.

Both models were trained using a five fold TimeSeriesSplit to preserve temporal order and prevent lookahead bias. Hyperparameters were selected using GridSearchCV with accuracy as the scoring metric. The final model for each ticker was refit on the combined training and validation set and evaluated on the held out test set.

Initial expectations were that accuracies would fall slightly above random guessing, typically between 52 and 58 percent. Actual performance aligned with this expectation since most models achieved between 50 and 65 percent accuracy. A few outliers exceeded 75 percent accuracy although these models often failed to produce meaningful trading signals which indicates overfitting to the directional label rather than learning a tradable pattern.

Trading Reports and Backtesting

Trading performance was evaluated through Backtrader integrated with QuantStats. For the top 10 tickers in the large universe ranked by accuracy, full trading reports were generated.

Two strategies were tested. ProbabilitySignalStrategy enters long positions when the model predicts an upward move and exits when predictions turn negative. TrendFilteredStrategy adds a 50 day SMA trend filter and a three percent trailing stop which produces more conservative and risk aware signals.

Each report includes a cumulative return curve, Sharpe ratio, maximum drawdown, monthly returns breakdown and trade statistics.

Representative results include the following:

- CORI with Gradient Boosting achieved a Sharpe ratio of 1.05 and a maximum drawdown of 3.44 percent

- AAME with Random Forest achieved a Sharpe ratio of 0.65 with a drawdown of 32.49 percent
- EQS with Random Forest achieved a Sharpe ratio of 0.51 and a drawdown of 9.95 percent
- BELFA with Random Forest achieved a Sharpe ratio of 0.20 with a drawdown of 5.07 percent

Several high accuracy models such as CBA, ABCO, AMTD, CETXP and CAPR produced no trades. These cases highlight an important distinction. High predictive accuracy on the next day direction does not necessarily mean that the predictions translate into profitable trading signals.

Additional reports such as TGT gradient boosting ProbabilitySignalStrategy and HAS random forest ProbabilitySignalStrategy also produced valid trading activity and full QuantStats summaries.

Conclusion

The two models demonstrated moderate effectiveness for the task of daily direction prediction. Most accuracies fell between 50 and 65 percent and a few models produced strong risk adjusted performance during backtesting. The Gradient Boosting model on CORI was the standout example with a Sharpe ratio greater than one and a very small drawdown.

Gradient Boosting tended to produce higher peak performance while Random Forest delivered more stable predictions with better interpretability. However Gradient Boosting was also more sensitive to hyperparameters and occasionally too conservative, which led to no trade scenarios. Random Forest trained quickly and reliably but sometimes experienced larger drawdowns.

The experiments highlight two important insights. First, directional accuracy alone is not a guarantee of trading profitability. Second, model performance varies significantly by ticker which suggests that a portfolio based approach or an ensemble across tickers may improve overall reliability.

Additional Considerations

Although some models achieved accuracy above 60 percent and occasional Sharpe ratios above one, the models in their current state are not ready for live trading. Typical accuracies provide only a modest edge that may be fully erased by transaction costs or slippage. Several top accuracy models produced no trades at all which further indicates prediction issues or overfitting.

To increase realism, future work should incorporate explicit transaction costs, slippage modeling, walk forward retraining, out of sample testing on data from 2022 to 2024, position sizing methods such as Kelly or risk parity, the inclusion of macroeconomic features and sentiment indicators and portfolio level evaluation instead of per ticker analysis.

There is also evidence of overfitting. Some models achieved suspiciously high accuracy but generated no trades. Overfitting was partially mitigated through time series cross validation, regularization through hyperparameters, train validation test splits and minimum data requirements but further steps such as early stopping, feature selection and simpler baseline models would strengthen generalization.

The overall conclusion is that the pipeline successfully demonstrates moderate predictive power, sensible strategy performance for a subset of tickers and a complete end to end machine learning and trading workflow. At the same time, the results highlight the challenges associated with applying machine learning to financial markets and emphasize the importance of thorough validation and realistic trading assumptions before considering deployment.