

Quant Insti Hackathon Report

Hirithik Yaadhav, ME23B032

Phone:	+91 7708930206	Smail:	me23b032@smail.iitm.ac.in
---------------	----------------	---------------	---------------------------

Explanation

Tech Stacks/ Libraries :

- ⇒ `yfinance` : For downloading daily OHLCV data for the 10 specified stocks.
- ⇒ `pandas` : For all data manipulation, time-series indexing, feature alignment, and signal matrix generation.
- ⇒ `scikit-learn` : For the core machine learning pipeline, including data preprocessing (`StandardScaler`) and modeling (meta model and its base estimators).
- ⇒ `TA-lib` : Used to generate robust technical indicators for feature engineering.
- ⇒ `vectorbt` : For the high-speed, simpler backtesting and performance simulation.

Model :

- ⇒ The core of the strategy is `VotingClassifier` model from `scikit-learn`. This model consists of three distinct base-learner models
 1. `RandomForestClassifier`
 2. `GradientBoostingClassifier`
 3. `AdaBoostClassifier`
- ⇒ The `soft` voting method is used, so that the average probabilities provided by all three models is used to ensure robustness
- ⇒ The model was trained by 15 engineered features which includes Returns, 14-Days RSI, Simple Moving Average, Price-to-MA Ratio, Volatility in varying windows and Bollinger Band Width.
- ⇒ It was trained to solve a Binary Classification problem which is to predict whether a stock's next 5-Day return would be positive (1) or not (0)

Assumptions :

- ⇒ This strategy assumes that the engineered features hold the predictive power for next week's returns
- ⇒ Transaction cost of 10 basis points is accounted for all entries and exits
- ⇒ The strategy rebalances on every week in beginning (usually mondays) and goes long on two stocks with highest predicted probabilities and allocated 50% of the portfolio to each.

Validation Method :

- ⇒ The backtest (2021 - 2025) was split into 5 one-year periods and for each test year, the model was trained once on the preceding 4 years of data (Eg : 2017-2020 data was used to train the model for 2021).
- ⇒ The model which is trained on 4 year window was then used to make weekly predictions for subsequent entire test year and this was repeated 5 times using loop ensuring that all predictions are fully out-of-sample.

Results and Insights

PORFOLIO STATS:	
Start	2021-01-04 00:00:00
End	2025-10-13 00:00:00
Period	1200
Start Value	100000.0
End Value	285070.322876
Total Return [%]	185.070323
Benchmark Return [%]	110.873003
Max Gross Exposure [%]	100.0
Total Fees Paid	47240.443146
Max Drawdown [%]	47.091678
Max Drawdown Duration	609.0
Total Trades	1346
Total Closed Trades	1344
Total Open Trades	2
Open Trade PnL	-481.778804
Win Rate [%]	61.904762
Best Trade [%]	39.469806
Worst Trade [%]	-24.463317
Avg Winning Trade [%]	3.897816
Avg Losing Trade [%]	-2.75101
Avg Winning Trade Duration	8.44351
Avg Losing Trade Duration	6.449219
Profit Factor	1.484213
...	
Annualized Return:	37.52486764427445 %
Volatility:	32.6715109806391 %
Sharpe Ratio:	1.139051383488284
Max Drawdown:	-47.09167778478937 %

Figure 1: Portfolio Performance

- ⇒ The ML-based strategy, after accounting for transaction costs, yielded a positive cumulative return of approximately 185% over the 5-year test period (Jan 2021 - Dec 2025). This corresponds to an annualized return of around 37%
- ⇒ This strategy as suggested by the problem statement has a great source of risk which is holding a highly concentrated two stock portfolio which is reflected in the volatility of around 32%
- ⇒ The strategy is also regime dependent which can be inferred from its cumulative returns plot, that the strategy performed well in the bullish 2021 and 2023-24 recovery period while brought a huge drawdown during the 2022 bear market.
- ⇒ The places to improve may involve diversifying the number of stocks to hold than to concentrate to two, using relevant features - stationary and non stationary or use a regime based approach in modelling.

For code and other results please refer the jupyter NB attached [here](#)