Machine Learning-Based Trading Strategy for Pfizer (PFE)

Ajoy Mathew (CS23B101)

June 13, 2025

1 Objective

This project aims to develop a machine learning-based trading model that predicts the next-day return of Pfizer (PFE) stock based on historical features. The strategy focuses on **daily long or short positions** with a base capital of **$1 million**. The goal is to outperform a long-only benchmark on out-of-sample data.

2 Data Description

I used publicly available daily price and volume data for PFE from January 1, 2010, to April 16, 2025 using alpha vantange library. The dataset includes the following fields:

- `Date`
- `Open, High, Low, Close`
- `Volume`

3 Data Preprocessing

3.1 Extraction and Cleaning

We begin by loading the raw historical data for Pfizer Inc. (PFE) from a CSV file using `pandas.read_csv()`. The dataset includes standard OHLCV (Open, High, Low, Close, Volume) features along with date information.

To ensure clean and consistent formatting:

- We remove numerical prefixes in column names (e.g., `"1. open"` becomes `"open"`) using regular expressions.
- All column names are standardized to lowercase and underscored for uniformity and code readability.
- The `date` column is parsed as a `datetime` object using `pd.to_datetime()` with `errors='coerce'` to handle any invalid dates gracefully.
- Rows with unparseable or missing dates are removed, and the `date` column is set as the index.
- The index is sorted chronologically and duplicate entries (if any) are dropped to ensure temporal consistency.
- The OHLCV columns are explicitly cast to `float` to avoid any downstream type issues.
- Any remaining rows with missing values across any feature are removed using `dropna()`.

3.2 Train-Validation-Test Split

The cleaned dataset is partitioned into three non-overlapping periods to support robust model development:

- **Training Set:** January 1, 2010 to December 31, 2019. This period is used to fit the model and estimate parameters.
- **Validation Set:** January 1, 2020 to December 31, 2021. This set is used for model selection, hyperparameter tuning, and performance evaluation.
- **Test Set (Out-of-Sample):** January 1, 2022 to April 16, 2025. This completely unseen data is reserved for final evaluation and simulates real-world deployment.

Each split is saved into separate CSV files to ensure reproducibility and modular usage in later pipeline stages.

3.3 Handling Non-Numeric Features

The dataset contains only numeric data; no categorical encoding is required.

4 Feature Engineering

The following features were computed with a rolling window of size 10:

- **Log Return:** $\log(\frac{P_t}{P_{t-1}})$
- **EMA:** Exponential moving average
- **Rolling Mean and STD:** Simple moving statistics
- **Price Deviation:** Deviation from rolling mean

Additionally:

- **Rolling Normalization:** Features are normalized using rolling mean and std
- **Squared Terms:** Each normalized feature is squared to capture non-linearities
- **Interaction Terms:** Pairwise products of normalized features

5 Modeling

5.1 Prediction Target

The binary target is defined as:

$$\texttt{target}_t = \begin{cases} 1 & \text{if return}_{t+1} > 0 \\ 0 & \text{otherwise} \end{cases}$$

6 Models Explored

We experimented with the following models:

- **Logistic Regression (base)** – simple, interpretable
- **Logistic Regression with Ridge (L2 penalty)** – reduced overfitting
- **Random Forest** – high in-sample accuracy, poor generalization
- **SVM (RBF kernel)** – sensitive to hyperparameters, expensive
- **LDA / QDA** – not robust under non-normality

7 Model Selection Rationale

We selected **Ridge Logistic Regression** for the final model due to:

- Simplicity and interpretability
- Robustness to overfitting via L2 regularization
- Balanced performance on training and validation data
- Probabilistic output allows threshold tuning

8 Final Model and Thresholding

We used:

- **Logistic Regression (L2)**
- **Thresholds:**

$$\texttt{signal} = \begin{cases} 1 & \text{if } \texttt{prob} > 0.5 \\ -1 & \text{if } \texttt{prob} < 0.5 \\ 1 & \text{otherwise} \end{cases}$$

9 Performance Metrics

Strategy Performance Summary

Validation Data

Metric	Value
Total Return (%)	-0.5233
Sharpe Ratio	-0.0067
Sortino Ratio	-0.0104
Max Drawdown (%)	-43.9341
Average Drawdown (%)	-26.6441
Benchmark Return (%)	-77.5681
Benchmark Max Drawdown (%)	-60.8664

Test Data

Metric	Value
Total Return (%)	78.2890
Sharpe Ratio	1.2802
Sortino Ratio	1.6853
Max Drawdown (%)	-22.4467
Average Drawdown (%)	-6.1328
Benchmark Return (%)	56.2514
Benchmark Max Drawdown (%)	-25.5358

10 Plots

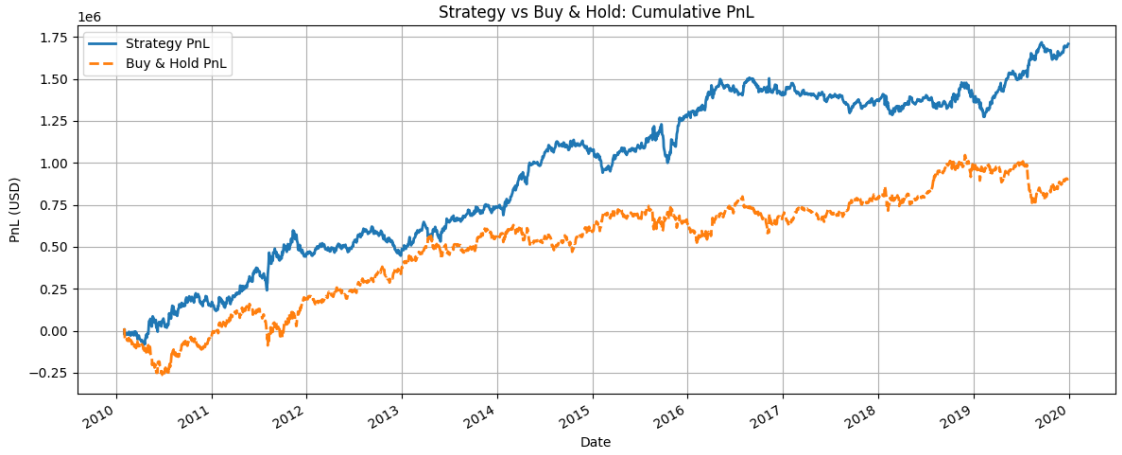

Figure 1: Strategy vs Benchmark Cumulative Returns on Train Data

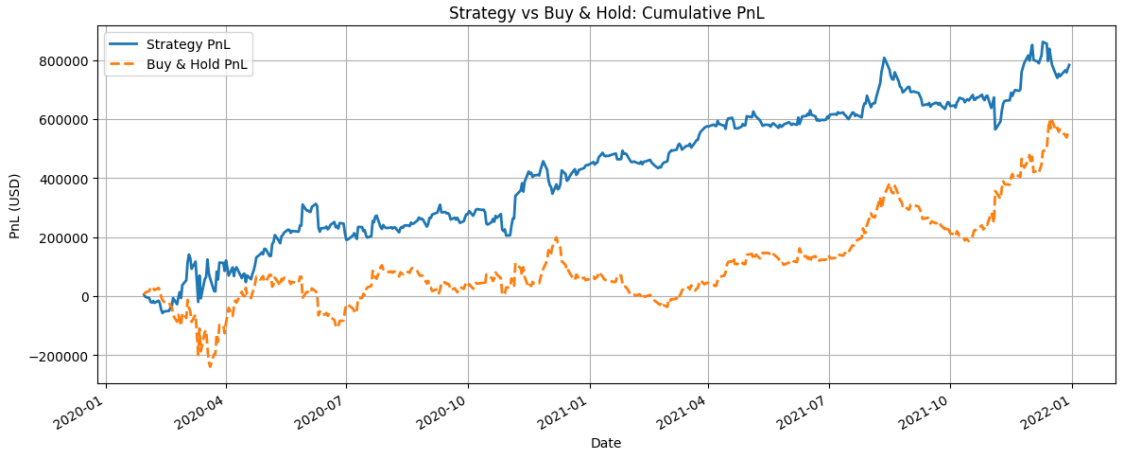

Figure 2: Strategy vs Benchmark Cumulative Returns on Validation Data

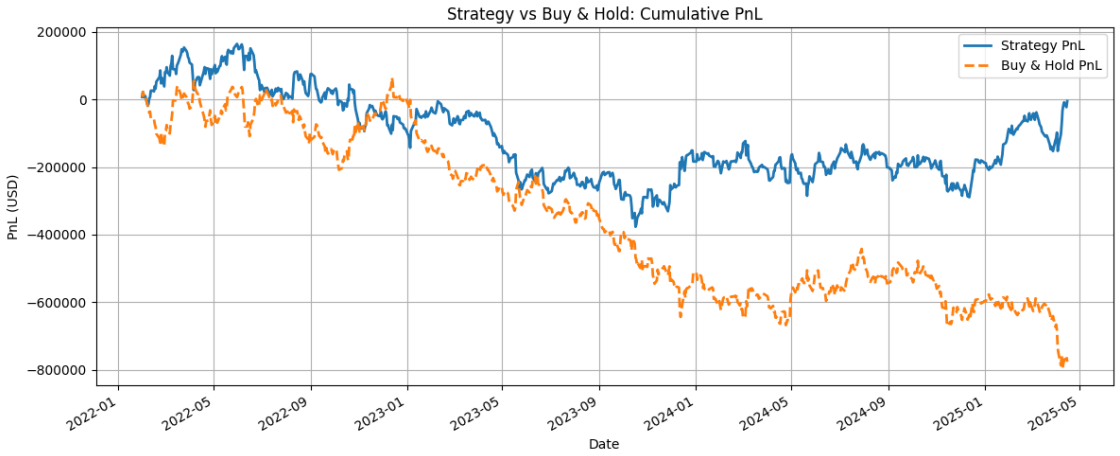

Figure 3: Strategy vs Benchmark Cumulative Returns on Test Data

11 Conclusion

This project demonstrates a rigorous pipeline for feature engineering, model selection, and backtesting. Ridge Logistic Regression provided the best balance of performance and generalization across training and out-of-sample datasets.

Further Possibilites/ Next Steps:

- Try additional external features (e.g., sentiment, macro indicators)
- Ensembling multiple weak models