Applications of Machine Learning to Portfolio Optimization

STAT 4710/5710 Final Project Writeup

Arnab Chaudhuri, Diego Ordonez, Yuxing Li

Spring 2026

Abstract

This project studies whether machine learning models can use stock-level market signals to predict next-month returns and form economically meaningful portfolios. Using monthly U.S. equity data merged with NAICS industry classifications, we engineer cross-sectional momentum, excess return relative to the S&P 500, short- and long-window moving-average signals, and a moving-average crossover. These factors are standardized within month so that the models compare stocks relative to their contemporaneous peers. We first use principal component analysis (PCA) to compress correlated signals into a lower-dimensional baseline and then compare Linear Regression, Ridge Regression, Lasso Regression, Random Forest, and Gradient Boosting. The raw forecasting task is difficult: all models have very low out-of-sample R^2, and Random Forest achieves the best prediction fit with RMSE of approximately 0.2117 and R^2 of approximately 0.0054. However, converting model predictions into industry-neutral long-short portfolios changes the interpretation. The PCA baseline earns about 6.2% cumulative return over the five-month test period, Ridge Regression produces the strongest risk-adjusted portfolio performance, and Gradient Boosting achieves the highest cumulative return. The main conclusion is that prediction accuracy and portfolio quality are related but not identical: in this short sample, portfolio construction matters as much as raw model fit.

1. Introduction and Motivation

A central problem in empirical finance is whether observable market signals contain enough information to forecast future stock returns. The question is practically important because even small improvements in ranking securities can matter for portfolio construction. Investors do not necessarily need to predict every stock's exact return. In many trading settings, it is enough to identify stocks that appear relatively attractive or unattractive compared with their peers. This makes the problem especially appropriate for modern data mining: the objective is not only prediction, but also translating noisy predictions into decisions.

This project asks the following research question:

> *Can engineered stock-level market signals, combined with machine learning models, predict next-month stock returns well enough to produce useful industry-neutral long-short portfolios?*

The project is motivated by two facts. First, stock returns are noisy, fat-tailed, and strongly affected by common market movements. Second, machine learning models can exploit nonlinear patterns and interactions, but financial prediction is vulnerable to overfitting and weak signal-to-noise ratios. For that reason, the analysis emphasizes a complete workflow rather than the most complex possible model. We begin by cleaning and organizing the data, engineer economically

interpretable signals, standardize them cross-sectionally, apply PCA to reduce correlation across predictors, compare several models, and finally evaluate whether predictions are useful when converted into portfolios.

Our strongest claim is not that we have discovered a production-ready trading strategy. The test period is short, transaction costs and slippage are not fully modeled, and some preprocessing choices would need stricter implementation in a production backtest. Instead, the contribution of the project is to demonstrate a disciplined machine learning workflow for return prediction and portfolio evaluation.

2. Data Description

2.1. Data Sources and Unit of Observation

The unit of observation is a stock-month. The main dataset contains monthly market information for U.S. equities, including stock identifiers, monthly returns, trading volume, shares outstanding, market benchmark returns, and other CRSP-style variables. A separate industry file provides NAICS industry codes. The two datasets are merged by PERMNO and date so that each stock-month observation can be assigned to an industry group.

The final cleaned sample contains approximately 205,376 stock-month observations, 10,058 distinct stocks, and 26 usable months after requiring enough rolling history to compute the 12-month moving average. The sample runs from November 2022 through December 2024, but the supervised learning target requires next-month returns, so the final out-of-sample test window ends in November 2024.

2.2. Key Variables

The most important raw and constructed variables are summarized in Table 1. The target variable is each stock's next-month return. Industry groups are based on the first two digits of NAICS codes, allowing the portfolios to rank stocks within broad industries.

Variable	Description
`permno` / `ticker`	Stock identifiers
`date`	Monthly observation date
`ret`	Current monthly stock return
`next_month_ret`	Supervised learning target, through shifting returns forward one month
`volume`	Monthly trading volume
`shares_out`	Number of shares outstanding
`sp500_ret`	Market benchmark return
`naics`	Industry classification
`industry_group`	First two digits of NAICS code
`cs_momentum`	Stock return minus average stock return in the same month
`excess_ret_vs_sp500`	Stock return minus S&P 500 return in the same month
`ma_3`, `ma_12`	Three-month and twelve-month moving average returns
`ma_crossover`	Difference between `ma_3` and `ma_12`

Table 1: Core variables used in the analysis, using names from the project code.

2.3. Cleaning and Preprocessing

The cleaning process removes observations with missing stock identifiers, dates, tickers, returns, industry classifications, and other required fields. It also removes impossible or extreme returns by keeping current and next-month returns greater than -100% and less than 1000%. Finally, the code requires at least 12 months of stock-level return history before an observation is used, because the longest engineered moving-average feature uses a 12-month window.

The target variable is constructed with a one-period lead within each stock:

$$\text{next\_month\_ret}_{i,t} = r_{i,t+1}.$$

This aligns current-month signals with future returns and makes the supervised learning task a true forecasting problem.

3. Exploratory Data Analysis

The exploratory plots in this section are included not as raw software output, but as diagnostics that motivate later modeling choices. In particular, they show that returns are noisy and fat-tailed, that the sample portfolio is related to but distinct from the S&P 500, and that the engineered factors are correlated enough to justify dimensionality reduction.

3.1. Return Distributions

The exploratory analysis shows that both current-month returns and next-month returns are noisy and fat-tailed. Even after removing impossible values and inspecting winsorized versions of the distributions, the data contain many large positive and negative observations. This is typical of individual-stock return data and helps explain why raw stock-level return prediction is difficult.

This motivates the decision to focus on relative signals. Rather than asking whether a return is high in absolute terms, the project asks whether a stock looks strong or weak relative to other stocks in the same month. That logic is reflected in the cross-sectional z-score normalization described below.

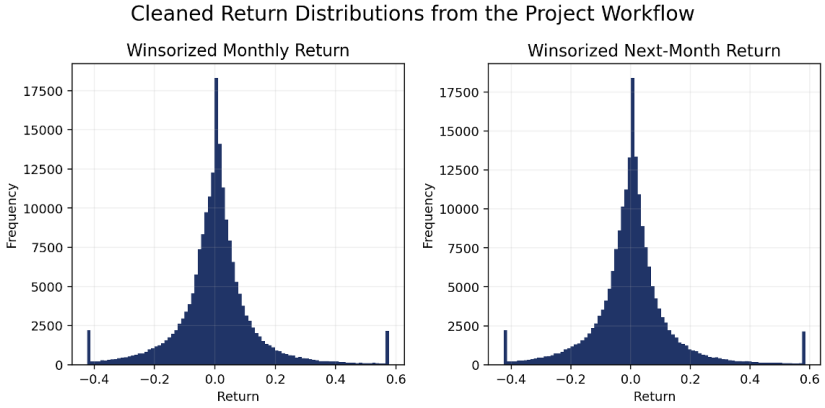

Figure 1: Winsorized current-month and next-month return distributions. The distributions remain noisy and fat-tailed, motivating relative cross-sectional signals rather than raw return levels.

3.2. Market Context

The project also compares the average stock return in the cleaned sample with the S&P 500. The average stock portfolio is computed as an equal-weighted monthly average of all available stock returns in the cleaned sample:

$$\bar{r}_t = \frac{1}{N_t} \sum_{i=1}^{N_t} r_{i,t}.$$

This differs from the S&P 500 because the cleaned sample contains a much broader universe of stocks and gives each stock equal weight, while the S&P 500 is a market-cap-weighted large-cap index. The comparison is still useful because it shows that broad market movement matters. Later, the long-short portfolio design is intended to focus less on market direction and more on relative stock selection.

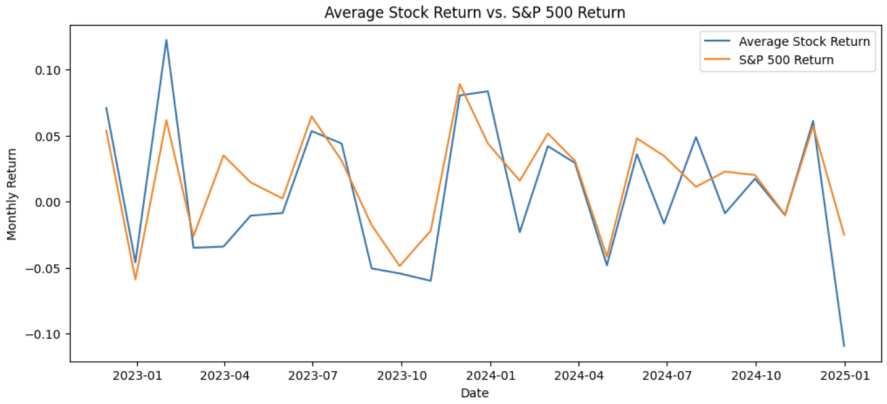

Figure 2: Monthly return comparison between the equal-weighted average stock portfolio in the cleaned sample and the S&P 500. The two series often move together, showing the importance of broad market exposure as a benchmark.

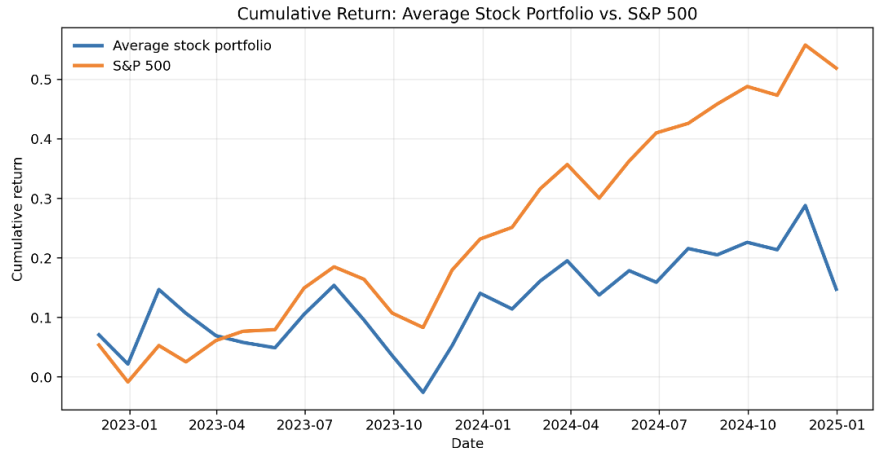

Figure 3: Cumulative return comparison between the equal-weighted average stock portfolio and the S&P 500. The series differ because the project sample is equal-weighted and broader than the market-cap-weighted S&P 500.

3.3. Standalone Factor Checks

Before fitting machine learning models, we examine the engineered factors through quintile analysis and simple long-short factor backtests. In these checks, stocks are ranked by a single factor each month, the top group is bought, and the bottom group is shorted. The 12-month moving-average z-score has the strongest individual performance among the simple factor tests, while several other signals perform weakly or negatively on their own. This finding supports the use of multivariate models and PCA rather than relying on one isolated predictor.

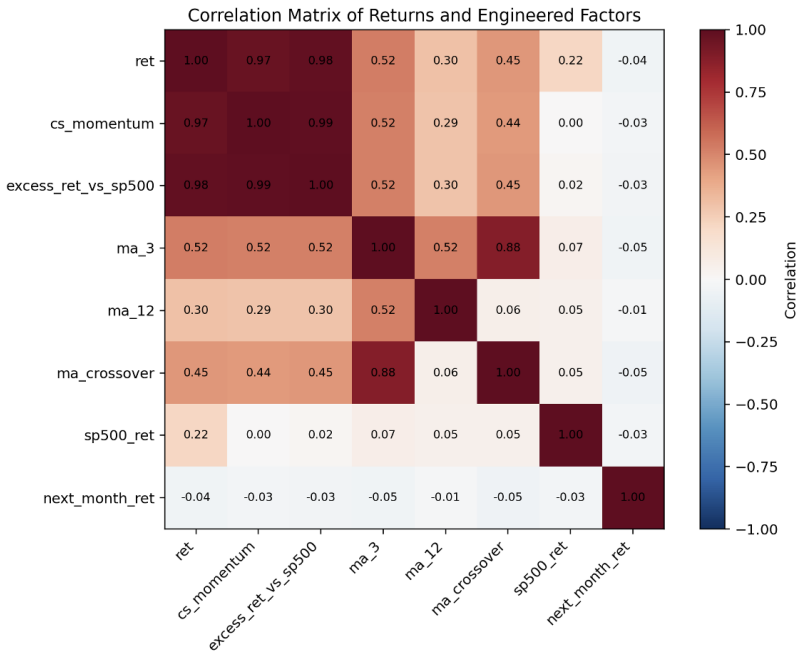

Figure 4: Correlation matrix for returns, engineered factors, market return, and next-month returns. Several engineered predictors are highly correlated, especially current return, cross-sectional momentum, excess return, and the short-window moving-average features.

4. Feature Engineering

The project engineers five main predictive signals from monthly stock returns and market benchmark returns:

1. **Cross-sectional momentum:** a stock's return minus the average stock return in the same month.
2. **Excess return versus S&P 500:** a stock's return minus the S&P 500 return in the same month.
3. **3-month moving average:** the average of the current and previous two monthly returns for the same stock.
4. **12-month moving average:** the average of the current and previous eleven monthly returns for the same stock.

5. **Moving-average crossover:** the 3-month moving average minus the 12-month moving average.

After constructing these signals, each factor is standardized within month:

$$Z_{i,t}^{(k)} = \frac{X_{i,t}^{(k)} - \bar{X}_t^{(k)}}{s_t^{(k)}},$$

where $X_{i,t}^{(k)}$ is factor k for stock i in month t, $\bar{X}_t^{(k)}$ is the cross-sectional monthly mean, and $s_t^{(k)}$ is the cross-sectional monthly standard deviation. This transformation makes the model compare stocks to their peers in the same month and reduces the influence of changes in the overall market environment.

5. Methods

5.1. Principal Component Analysis

The engineered signals are correlated. For example, cross-sectional momentum and excess return relative to the S&P 500 both depend heavily on current stock returns. Principal Component Analysis (PCA) is used to compress these overlapping predictors into a smaller set of orthogonal components. The first three principal components explain approximately 84.5% of standardized signal variation, with PC1 explaining about 51.9%, PC2 explaining about 16.7%, and PC3 explaining about 16.0%.

The PCA baseline model uses the first three principal components in a linear regression:

$$\widehat{r}_{i,t+1} = \beta_0 + \beta_1 PC1_{i,t} + \beta_2 PC2_{i,t} + \beta_3 PC3_{i,t}.$$

This baseline is intentionally simple. It tests whether compressed signal information can produce useful rankings before moving to more flexible models.

Figures 4 and 5, together with Table 2, motivate the PCA step. The correlation matrix shows that several engineered predictors overlap substantially, while the scree plot and explained-variance table show that the first three components capture most of the standardized signal variation.

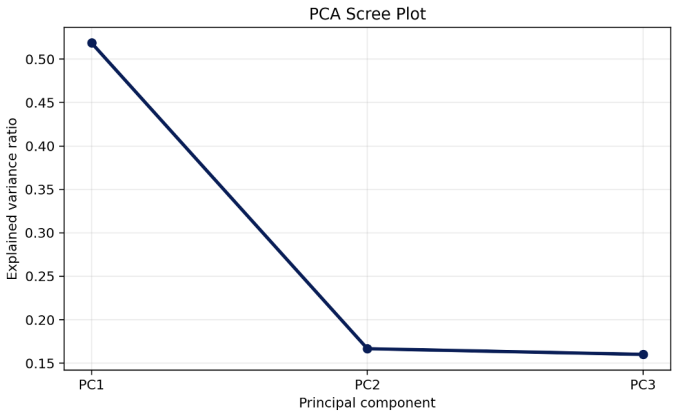

Figure 5: Explained variance by the first three principal components. The first component explains about 51.9% of standardized signal variation, while the first three components together explain about 84.5%.

Principal Component	Explained Variance	Cumulative Explained Variance
PC1	51.9%	51.9%
PC2	16.7%	68.5%
PC3	16.0%	84.5%

Table 2: Explained variance from the first three principal components.

A methodological limitation is that PCA is fit on the full modeling dataset in the current implementation. A stricter backtest would fit the scaler and PCA only on the training period and then apply the learned transformation to the test period. This is discussed again in the limitations section.

5.2. Machine Learning Models

The full model comparison uses both the PCA components and the original z-scored engineered factors. The feature set is

$$\{PC1, PC2, PC3, \text{cs\_momentum\_z}, \text{excess\_ret\_vs\_sp500\_z}, \text{ma\_3\_z}, \text{ma\_12\_z}, \text{ma\_crossover\_z}\}.$$

The models compared are Linear Regression, Ridge Regression, Lasso Regression, Random Forest, and Gradient Boosting.

Linear Regression provides a transparent benchmark. Ridge and Lasso add regularization, which can help control overfitting in correlated feature spaces. Random Forest and Gradient Boosting introduce nonlinear prediction rules and can capture interactions between variables. The models are trained on observations from November 2022 through June 2024 and tested out of sample from July 2024 through November 2024.

5.3. Prediction Metrics

The primary raw prediction metrics are RMSE and out-of-sample R^2. RMSE measures the average size of prediction errors:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i,t}(r_{i,t+1} - \widehat{r}_{i,t+1})^2}.$$

Out-of-sample R^2 measures how much variation in realized next-month returns is explained by the model in the test period. Because monthly stock returns are extremely noisy, we expect these values to be low.

5.4. Portfolio Construction

After generating predictions, the project converts them into portfolios. This step is crucial because the economic value of a model depends on whether its predictions produce useful rankings, not only whether it minimizes average prediction error.

For the PCA baseline, stocks are ranked by predicted return within each industry-month. The top 10% of stocks are assigned to the long portfolio, the bottom 10% are assigned to the short portfolio, and the remaining stocks are neutral. The realized long-short return in month t is

$$R_t^{LS} = R_t^{Long} - R_t^{Short}.$$

This creates an industry-neutral long-short strategy because ranking occurs separately within each industry group.

For the full model comparison, the portfolio is more selective. Within each industry-month, stocks are sorted into 20 buckets. The top 5% become long positions and the bottom 5% become short positions. The code uses rank-based weights: higher predicted returns receive larger positive weights on the long side, and lower predicted returns receive larger negative weights on the short side. Weights are then normalized so that total long exposure equals +1 and total short exposure equals −1 each month. The realized portfolio return is

$$R_t^{portfolio} = \sum_i w_{i,t} r_{i,t+1}.$$

Cumulative returns are then computed by compounding monthly portfolio returns:

$$\prod_{t=1}^{T}(1 + R_t^{portfolio}) - 1.$$

Risk-adjusted performance is measured using the annualized Sharpe ratio:

$$\text{Sharpe} = \frac{\bar{R}_{monthly}}{\sigma_{monthly}}\sqrt{12}.$$

6. Results

6.1. PCA Baseline

The PCA baseline has weak raw forecasting performance. Its out-of-sample RMSE is approximately 0.2122 and its R^2 is approximately 0.0007. This means the model explains very little of the variation in realized stock-level next-month returns.

However, the portfolio result is more encouraging. When PCA predictions are converted into an industry-neutral long-short portfolio, the strategy earns about 6.15% cumulative return over the five-month test period. The average monthly long-short return is about 1.22%, monthly volatility is about 2.08%, and the annualized Sharpe ratio is about 2.03. These results are summarized in Table 3 and Figure 6.

Metric	Value
RMSE	0.2122
Out-of-sample R^2	0.0007
Average monthly long-short return	1.22%
Monthly volatility	2.08%
Annualized Sharpe ratio	2.03
Cumulative return	6.15%
Test months	5

Table 3: PCA baseline prediction and portfolio results.

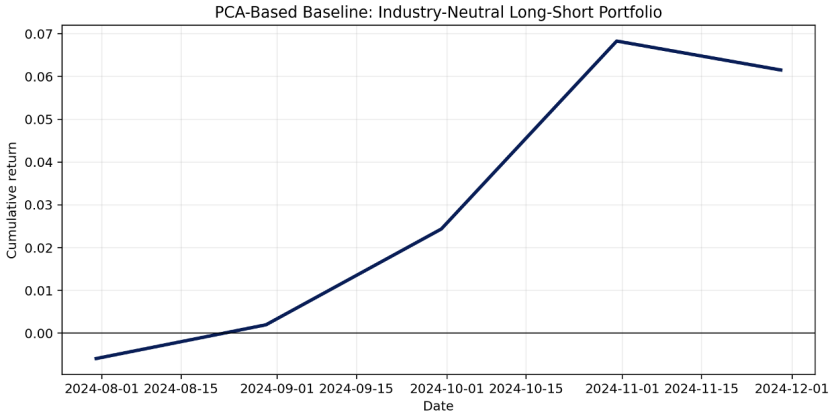

Figure 6: Cumulative return of the industry-neutral PCA-based long-short portfolio over the out-of-sample test period. Although raw prediction fit is weak, the PCA ranking strategy generates a positive cumulative return across the five-month test window.

The important interpretation is that exact return prediction and ranking performance are not the same. The PCA baseline does not precisely predict individual returns, but its predictions still contain enough ranking information to produce a positive long-short result in the test window.

6.2. Model Comparison

Table 4 reports the raw prediction results for the full model comparison. Random Forest has the best prediction fit, with the lowest RMSE and the highest R^2. However, the improvement over simpler models is small, and all R^2 values remain very low. This confirms that next-month stock return prediction is a difficult problem in this dataset.

Model	RMSE	R^2
Linear Regression	0.212206	0.000597
Ridge Regression	0.212205	0.000598
Lasso Regression	0.212202	0.000628
Random Forest	0.211694	0.005407
Gradient Boosting	0.214045	-0.016808

Table 4: Out-of-sample prediction performance by model.

The portfolio results tell a different story. Random Forest wins on raw RMSE, but Ridge Regression produces the strongest risk-adjusted portfolio performance, with a Sharpe ratio of approximately 2.53. Gradient Boosting achieves the highest cumulative return. This split across metrics is the main substantive finding of the project: the best forecasting model is not necessarily the best trading model.

Criterion	Best model
Lowest RMSE	Random Forest
Highest out-of-sample R^2	Random Forest
Highest Sharpe ratio	Ridge Regression
Highest cumulative return	Gradient Boosting

Table 5: Main model winners by evaluation criterion.

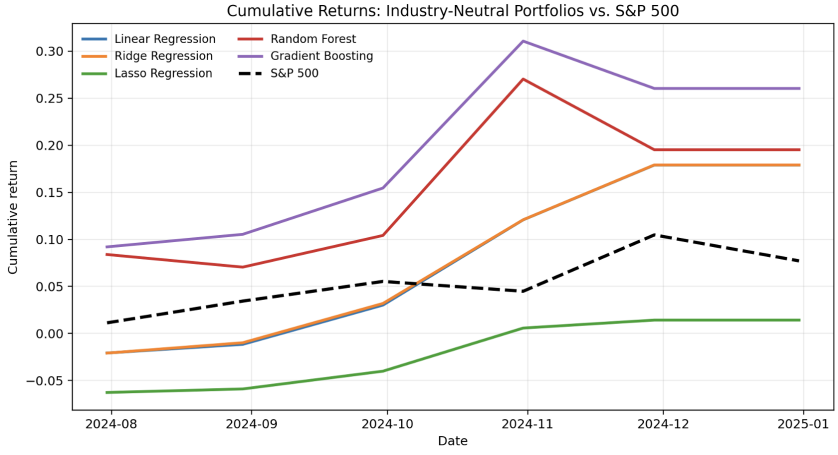

Figure 7: Cumulative returns of industry-neutral model portfolios compared with the S&P 500 during the test period. The figure emphasizes that the models should be judged not only by raw prediction error, but also by how their predictions translate into realized portfolio performance.

6.3. Interpretation of the Findings

There are three central findings. First, stock return prediction remains difficult. The best raw prediction model has an out-of-sample R^2 of only about 0.0054. Second, portfolio construction matters. The model with the best RMSE does not necessarily generate the best risk-adjusted trading strategy. Third, industry-neutral ranking is central to the results because it makes the portfolio less dependent on broad sector bets and more focused on relative stock selection.

These findings are consistent with the nature of financial prediction. Monthly stock returns contain a large amount of noise, so a model may not explain much overall variation. Still, small improvements in the ranking of stocks can matter when the strategy only trades the most extreme predicted winners and losers.

7. Discussion and Conclusion

This project demonstrates a complete machine learning workflow for portfolio construction. Starting from raw monthly equity data, we merge industry classifications, clean and filter observations, engineer interpretable signals, standardize factors within month, use PCA to reduce correlated features, compare several predictive models, and evaluate out-of-sample long-short portfolios.

The main conclusion is that statistical prediction quality and portfolio quality are related but not identical. Random Forest is the best model for raw next-month return prediction, but Ridge Regression is strongest on a Sharpe basis, and Gradient Boosting produces the highest cumulative return. This means that model evaluation depends on the final use case. If the goal is to minimize prediction error across all stocks, Random Forest is preferred. If the goal is to build a more stable risk-adjusted portfolio, Ridge Regression is more attractive in this sample.

The results should be interpreted cautiously. The final test period contains only five months, so the portfolio metrics are highly sample-dependent. A few strong or weak months can meaningfully change cumulative return and Sharpe ratio. The backtests also do not fully account for transaction costs, slippage, shorting constraints, borrow costs, or turnover. These considerations are especially important for long-short strategies, which may require frequent rebalancing and can be expensive to implement.

There is also a methodological limitation in the PCA implementation. In a stricter research design, the scaler and PCA transformation should be fit only on the training data and then applied to the test data. This would prevent any test-period information from entering the feature transformation. Future work should also use a longer sample, rolling or walk-forward retraining, explicit transaction-cost modeling, turnover analysis, and robustness checks across different market regimes.

Overall, the project supports a disciplined ML investing workflow, but not a claim of reliable permanent alpha. The strongest contribution is methodological: it shows how to move from raw stock data to engineered signals, from predictions to industry-neutral portfolios, and from raw model accuracy to economically interpretable performance measures.

8. Statement on AI Use

Generative AI tools, including ChatGPT, were used to help organize the written report, improve wording, create presentation explanations, and translate the code workflow into clear prose. AI was

not used to create the original dataset or independently verify the numerical results. The reported results come from the submitted project code and presentation materials.

A. Appendix: Reproducibility Notes

The analysis was conducted in Python. The main packages used include `pandas`, `numpy`, `duckdb`, `scikit-learn`, `matplotlib`, and `seaborn`. The submitted code file contains the full workflow: data loading, merging, cleaning, feature engineering, EDA, PCA, model fitting, portfolio construction, and backtesting. Data was collected from the CRSP (Center for Research in Security Prices) database.

B. Appendix: Summary of Workflow

1. Load monthly stock market data and industry classification data.
2. Merge the datasets using stock identifier and date.
3. Convert NAICS codes into two-digit industry groups.
4. Clean the sample and require enough rolling return history.
5. Construct the next-month return target.
6. Engineer cross-sectional momentum, excess market return, moving averages, and moving-average crossover.
7. Z-score factors within each month.
8. Apply PCA and retain the first three principal components.
9. Train models on November 2022 through June 2024.
10. Test models on July 2024 through November 2024.
11. Rank stocks within each industry-month by predicted return.
12. Form industry-neutral long-short portfolios and evaluate realized performance.

C. Appendix: Code Repository for Replication

All code needed to reproduce the main analysis is available in the project GitHub repository:

https://github.com/dordonez10/ApplicationsOfMLToPortfolioOptimization/tree/main

The repository contains the project code used for data loading, cleaning, feature engineering, exploratory data analysis, PCA, model fitting, portfolio construction, and backtesting. To replicate the analysis, readers should download the repository files, place the required raw data files in the expected working directory, and run the main analysis notebook or script from start to finish.