FIAM x DESAUTELS

Faculty of Management
McGill University

Navigating the Forefront of Capital Markets & AI

Team GitGud

Angela, Nathan, Nivedha, Ronald, Ryan

University of Toronto, Carleton, Queens

October 24th , 2024

Meet the team

Team Description

FIAM × DESAUTELS

Angela Chen

Work Experience:

shopify

Education:

Carleton University

Nathan Lim

Work Experience:

CREATIVE DESTRUCTION LAB

Education:

University of Toronto

Nivedha Madhivanan

Work Experience:

CPP Investments

Education:

University of Toronto

Ronald Sun

Work Experience:

intel

Education:

Queen's University

Ryan Diep

Work Experience:

BANK OF CANADA
BANQUE DU CANADA

Education:

Carleton University

Optimizing Stock Returns with AI and Portfolio Management to Outperform the S&P500

Executive Summary

FIAM × DESAUTELS

Hypothesis

By using machine learning and dynamic portfolio optimization, our team seeks to deliver sustainable, reliable returns while meeting diverse institutional needs for liquidity and risk management, consistently outperforming the S&P500.

Strategy

1 Feature Engineering

- Used a combination of quantitative and fundamental feature engineering
- Cross-referenced feature selection results to obtain the 30 most impactful features

2 Machine Learning Model

- The heart of our strategy uses the LSTM model, designed to predict monthly stock returns by analyzing historical price data and market trends

3 Mean Variance Optimization

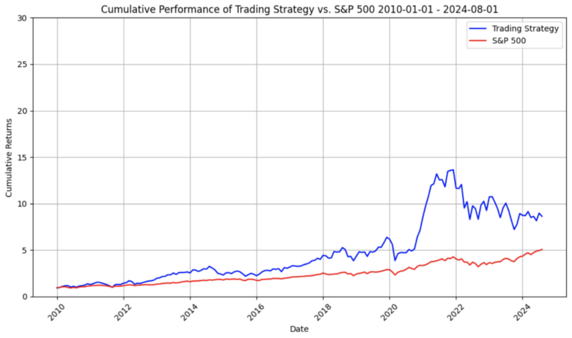
- Portfolio optimization converts predicted stock returns into investment decisions, maximizing risk-adjusted returns through mean-variance optimization.

Key Ideas

Customizable + Long Term Horizon + Efficiency + Heuristics = Alpha

Results

Cumulative Returns



Top 10 Holdings

Name	ROI
1. GME:	19.8805x
2. VNDA:	12.4019x
3. MSEL:	11.8530x
4. PTIE:	11.6093x
5. ARIA:	10.6043x
6. AIRB:	9.4735x
7. EGHT:	8.9897x
8. EXCA:	8.9705x
9. CVSN:	8.9672x
10. SNTA.:	8.5567x

Sharpe Ratio + returns vs S&P500

0.307 — *Sharpe Ratio*

19.5% — *Git Gud*

6.06% — *S&P500*

Investment Strategy

Feature Selection

The Fuel

Machine Learning Model

The Machine

Mean Variance Optimization

The Outputs

Data Parsing & Used a Combination of Quantitative & Fundamental Feature Engineering

Feature Selection

FIAM × DESAUTELS

Data Summary

The new data consists of small to mid-cap stocks from January 2000 to August 2024. It includes thousands of stocks with 147 firm-specific financial characteristics.

As a result, the amount of data we are processing at least 2-4 times bigger.

First Pass Stock Filter

- We are looking for companies with sufficient growth movements
- Deleted the stocks of the training data that never exceeds the bottom 30th percentile in Market cap.

Compute Time Reduction

- To reduce computing time, we're extracting the top 800 mkt cap stocks.
- Continuously changing the list as time goes on.
- Still ensures **correct proof of concept** of our model

Feature Engineering:

Fundamental

Using academic papers and expert references, we hand selected financial features.

These features are more fundamental factors including Seasonality, Cashflow and financial statement ratios.

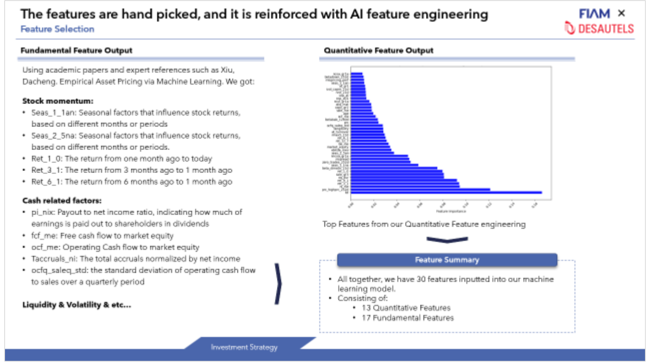
Reference: Xiu, Dacheng. Empirical Asset Pricing via Machine Learning - Dacheng Xiu
Please see appendix for more

+

Technical

Using five regressions, we quantitatively determined the most important features for predicting returns.

1. **Histogram-Based Gradient Boosting** fits each tree to the negative gradient of the loss function with respect to the predicted value
2. **XgBoost** for time series data, then computes features for each window in time
3. **Random Forest** based on bagging and use a majority vote to predict the outcome
4. **Lasso** regularization for a penalty that forces small feature coefficients to zero.
5. **Ridge** regularization which helped prevent overfitting by shrinking the coefficients



The features are hand picked, and it is reinforced with AI feature engineering

Feature Selection

Fundamental Feature Output

Using academic papers and expert references such as Xiu, Dacheng. Empirical Asset Pricing via Machine Learning. We got:

Stock momentum:

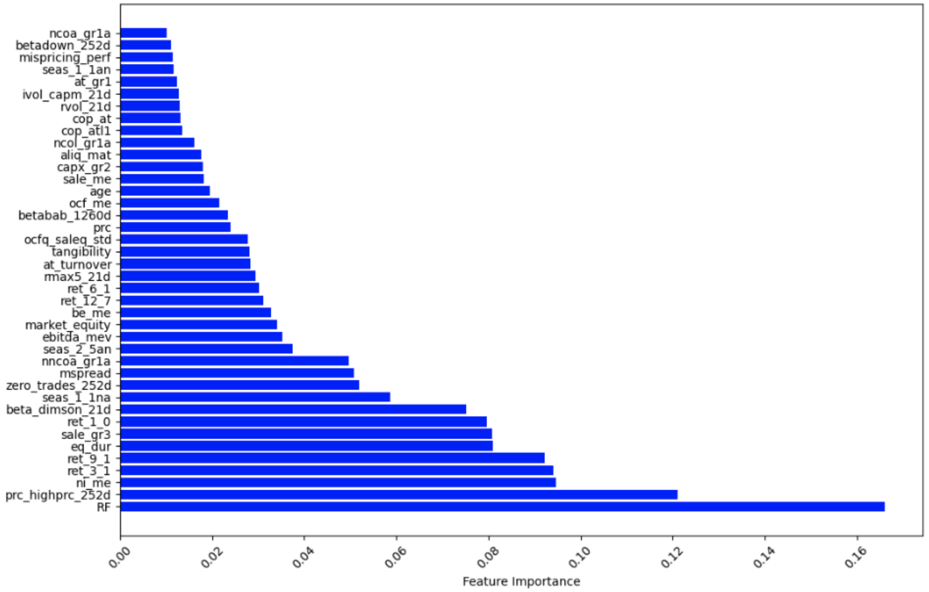
- Seas_1_1an: Seasonal factors that influence stock returns, based on different months or periods
- Seas_2_5na: Seasonal factors that influence stock returns, based on different months or periods.
- Ret_1_0: The return from one month ago to today
- Ret_3_1: The return from 3 months ago to 1 month ago
- Ret_6_1: The return from 6 months ago to 1 month ago

Cash related factors:

- pi_nix: Payout to net income ratio, indicating how much of earnings is paid out to shareholders in dividends
- fcf_me: Free cash flow to market equity
- ocf_me: Operating Cash flow to market equity
- Taccruals_ni: The total accruals normalized by net income
- ocfq_saleq_std: the standard deviation of operating cash flow to sales over a quarterly period

Liquidity & Volatility & etc...

Quantitative Feature Output



Top Features from our Quantitative Feature engineering

Feature Summary

- All together, we have 30 features inputted into our machine learning model.
- Consisting of:
  - 13 Quantitative Features
  - 17 Fundamental Features

Investment Strategy

Feature Selection

The Fuel

Machine Learning Model

The Machine

Mean Variance Optimization

The Outputs

The Pros and Cons of Various Machine Learning Models

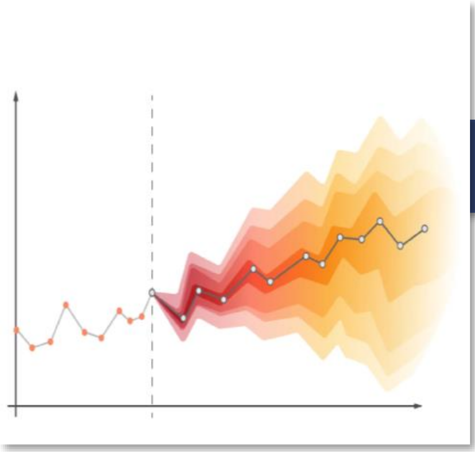
Investment Strategy

What We Tried

Researched and attempted many learning models.

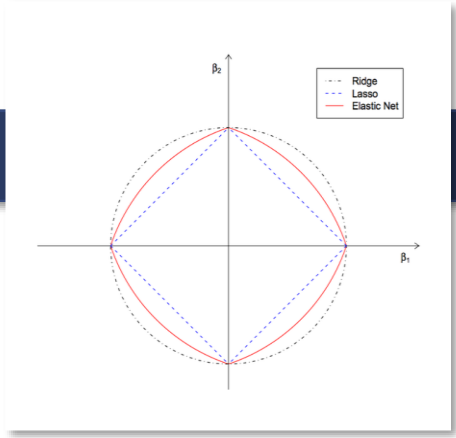
1 ARIMA

- Limited to linear patterns
- Requires stationary data
- Unreliable in long-term forecast
- Does not make use of given data
- Requires outlier pre-processing



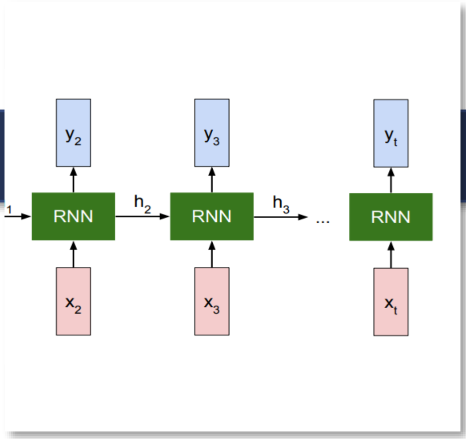
2 Elastic Net

- Deals with feature selection and regularization
- Can overfit on small datasets with high dimensionality
- Not great with long term predictions
- Struggles to capture non-linear relationships



3 Recurrent Neural Net

- Captures sequential dependencies
- Models complex patterns
- Vanishing gradient problem
- Struggles to remember earlier data



Lack of Time Series

Each method had limitations for time-series modeling, leading us to a more sophisticated approach.

Time-Series Financial Forecasting and Portfolio Optimization

Long Short-Term Memory LSTM

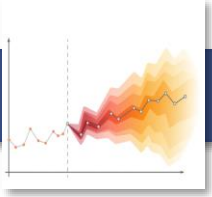
Honourable mentions

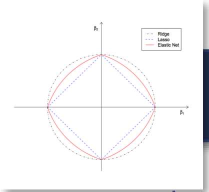
1

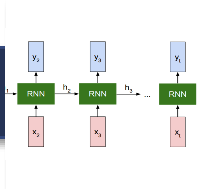
ARIMA
- Limited to linear patterns
- Requires stationary data
- Unreliable in long-term forecast
- Does not make use of given data
- Requires outlier pre-processing

3

Recurrent Neural Net
- Captures sequential dependencies
- Models complex patterns
- Vanishing gradient problem
- Struggles to remember earlier data







Elastic Net
- Deals with feature selection and regularization
- Can overfit on small datasets with high dimensionality
- Not great with long term predictions
- Struggles to capture non-linear relationships

2

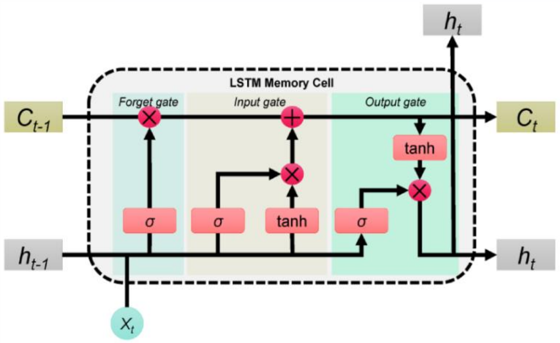
Lack of Time Series

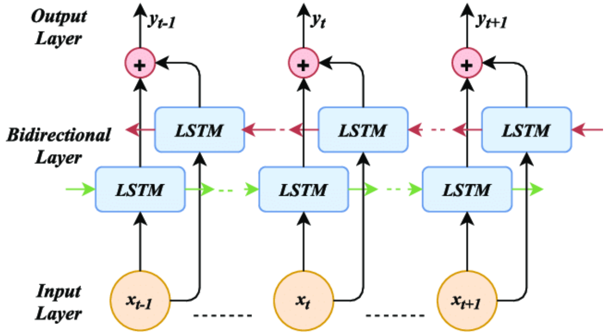
Each method had limitations for time-series modeling, leading us to a more sophisticated approach.

Our main model: LSTM – Long-Short Term Memory

- Sequential data: Excels in tasks where order of inputs is significant
- Adapts to changing patterns over time, which is suitable for stock market
- Mitigates VGP, improving long sequences training and convergence
- Learns importance of different features, capturing complex relationships

LSTM Architecture:





Memory Cell enables model to store relevant info over many time steps

Forget gate resolves the vanishing gradient problem

Input/Output determines how much information is passed

Bidirectional layers allows learning in both forward and backward directions

Breaking down the complexities of machine learning

LSTM Architecture

Layers

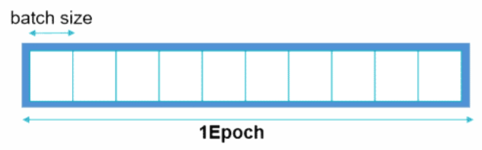
- The Sequential() class was used to build our model layer by layer, where parameters can be specified. Below is the architecture of our model.

We have a total of 4 layers to achieve one prediction

```
model = Sequential()
model.add(Input(shape=
    (X_train.shape[1],
    X_train.shape[2])))
model.add(Bidirectional(
    LSTM(32, return_sequences=True)))
model.add(Dropout(0.3))
model.add(Bidirectional(
    LSTM(16, activation='relu',
        return_sequences=False)))
model.add(Dropout(0.4))
model.add(Dense(16, activation='relu'))
model.add(Dense(1))
```

Batch Size and Epoch

- Per yearly out-of-sample predictions, we used 30 epochs, with batch sizes of 64. These decisions are balanced between computational efficiency and maintaining model performance.



Optimizer

AdaGrad: penalizes the learning rate too harshly
RMSProp: doesn't use momentum

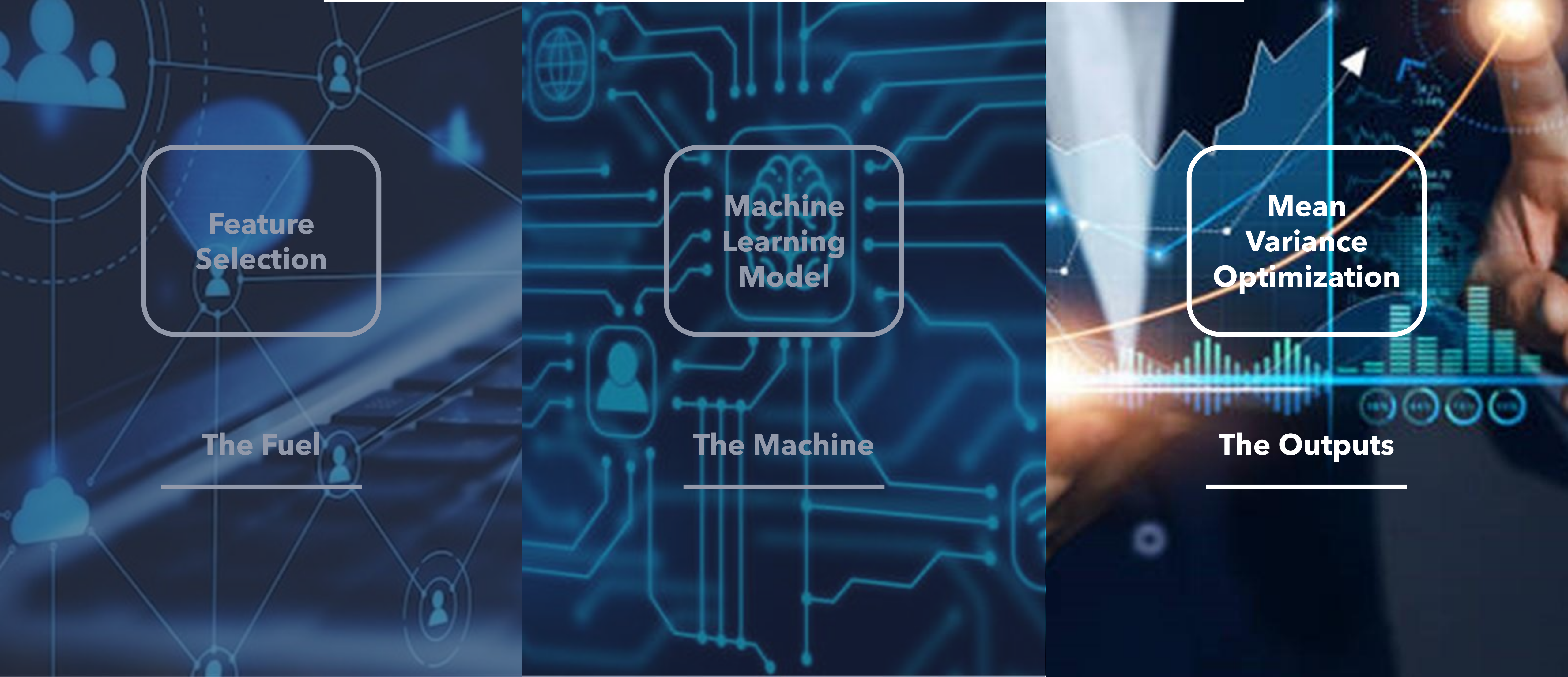
Adam optimizer was used for its efficiency in handling sparse and noisy gradients, combining the advantages of AdaGrad and RMSProp.

Loss Function

Mean squared error: penalizes large errors more heavily
Absolute error: treats all errors equally

The **Huber** loss function was chosen for its ability to handle outliers in stock returns, balancing sensitivity between mean squared error and absolute error.

Investment Strategy



Optimizing our portfolio

Mean Variance Optimization

Optimization

Portfolio optimization is the final and most critical element of our investment strategy. It is the final step in translating the predicted stock returns from our machine-learning model into actual investment decisions.

- **Maximizes risk-adjusted returns** (maximizing the risk-adjusted utility, a key performance measure;)
- Encourages diversification by selecting stocks with low or negative correlations using a **correlation matrix** (reducing volatility) with **shrinkage**

Objective Function:

$$\boldsymbol{max}\ E[Rp] - \lambda \cdot \sigma_p^2$$

Where,

- $E[Rp] = w^T\mu$, is the **expected portfolio return**
- $\sigma_p^2 = w^T\Sigma\omega$, is the **portfolio variance**
- λ, is the **risk aversion coefficient**

Constraints:

Positive weights for a long-only portfolio

Sum of Weights = 1

Turnover less than 25%

Ticker weight less than 10%

Iterative Process - Monthly

Data Preparation → Top 50 stocks each month passed into our optimization model

Compute Covariance Matrix → Sample covariance matrix with shrinkage

Set up Optimization Problem → Max risk-adjusted return against specific constraints

Future Applications

The risk aversion parameter can be adjusted as per the risk tolerance for different asset classes/investors

To reduce noise, and improve estimates and input quality for the MVO, use a factor model to predict returns and estimate covariance matrix

Portfolio Performance

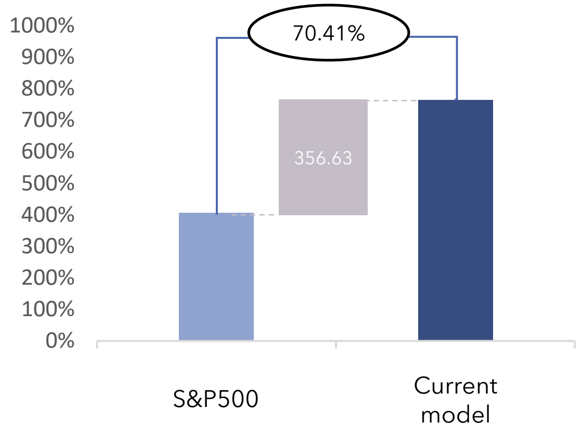
The Results

GitGud Portfolio Outperforms S&P 500 with 19.5% CAGR

Portfolio Performance Results

Cumulative Portfolio Returns Comparison

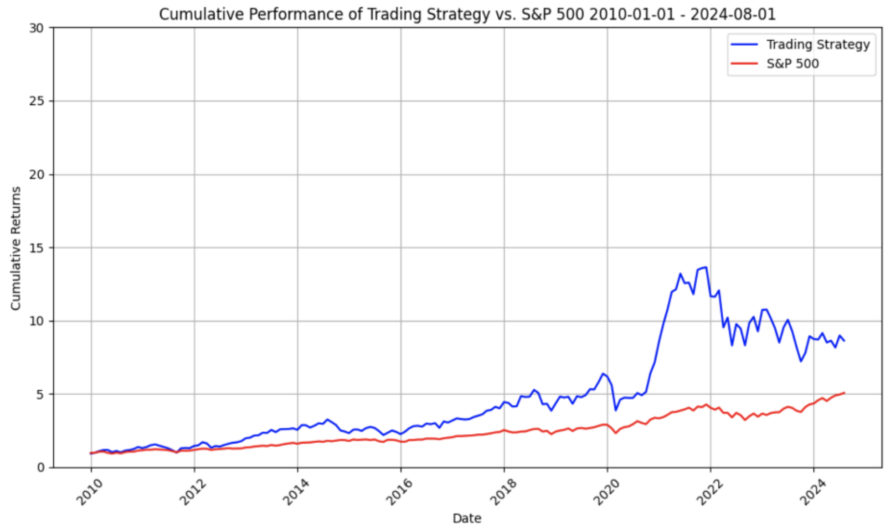
Model Outperformance



Details

- With a starting investment of 1 million, GitGud's portfolio returns 70% more than the benchmark

- For a stronger evaluation of the strategy performance, it would be good to evaluate our performance against the S&P Small Cap Index or other equivalent benchmark



Cumulative Portfolio Returns Comparison

	Git Gud Performance	S&P 500 Performance
Average annualized returns	**0.195**	0.117
Annualized Std Dev.	0.307	**0.146**
CAPM Alpha	0.022	n.a.
Annualized Alpha	0.260	n.a
Sharpe Ratio (annualized)	**0.636**	0.434
Information Ratio	0.849	n.a.
Maximum Drawdown	0.637	**0.746**
Maximum 1 Month Loss	-0.309	**-0.169**
Maximum 1 Month Loss	0.099	n.a.

Cumulative Portfolio Returns Comparison

Name	ROI
1. GME:	19.8805x
2. VNDA:	12.4019x
3. MSEL:	11.8530x
4. PTIE:	11.6093x
5. ARIA:	10.6043x
6. AIRB:	9.4735x
7. EGHT:	8.9897x
8. EXCA:	8.9705x
9. CVSN:	8.9672x
10. SNTA.:	8.5567x

Conclusion

The Reflection

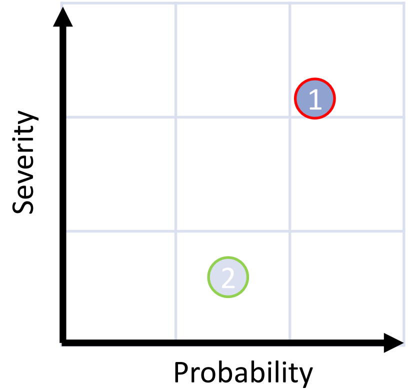
Future Potential
Reflection

FIAM × DESAUTELS

Risk and Mitigations

	Areas of Concern	Mitigation
Illiquidity	Lower levels of liquidity lead to higher transaction costs and challenges in executing trades	Consider introducing position sizing limits based on liquidity and use liquidity constraints when selecting stocks
Market Regime Shifts	The model may perform well under certain market but struggle during sharp market regime shifts (financial crises)	Build regime detection models that classify market environments and adjust stock selection accordingly

Threat Assessment



Risk Level

Next Steps

Further Tune Hyperparameters: For instance, increasing batch size/decrease epochs for overfitting (time constraints)

Web Scraping: Sentimental Analysis

Event Driven Analysis: High market cap stocks tend to be influenced by the news; **take Tesla's 20% increase today**

Institution Readiness

Scalable

Customizable

Intentional Innovation

Appendix

Citations/Resources

Thank you for Reading!

M. A. Istiake Sunny, M. M. S. Maswood and A. G. Alharbi, "Deep Learning-Based Stock Price Prediction Using LSTM and Bi-Directional LSTM Model," 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES), Giza, Egypt, 2020, pp. 87-92, doi: 10.1109/NILES50944.2020.9257950. keywords: {Computer architecture;Logic gates;Microprocessors;Predictive models;Forecasting;Autoregressive processes;Time series analysis;RNN;LSTM;BI-LSTM;Stock Market Prediction;Deep Learning},

Hum Nath Bhandari, Binod Rimal, Nawa Raj Pokhrel, Ramchandra Rimal, Keshab R. Dahal, Rajendra K.C. Khatri,Predicting stock market index using LSTM,Machine Learning with Applications,Volume 9,2022,100320,ISSN 2666-8270,https://doi.org/10.1016/j.mlwa.2022.100320

Y. Liu, Z. Wang and B. Zheng, "Application of Regularized GRU-LSTM Model in Stock Price Prediction," 2019 IEEE 5th International Conference on Computer and Communications (ICCC), Chengdu, China, 2019, pp. 1886-1890, doi: 10.1109/ICCC47050.2019.9064035. keywords: {Logic gates;Predictive models;Indexes;Neural networks;Computational modeling;Time series analysis;Computer architecture;LSTM;GRU;Time series prediction},

Adil Moghar, Mhamed Hamiche,Stock Market Prediction Using LSTM Recurrent Neural Network,Procedia Computer Science,Volume 170,2020,Pages 1168-1173,ISSN 1877-0509,https://doi.org/10.1016/j.procs.2020.03.049

Verifying LSTM stock price prediction effectiveness using TQUANT lab (part 1) - tej. TEJ台灣經濟新報. (2024, August 16). https://www.tejwin.com/en/insight/lstm-stock-price-prediction/#:~:text=Finally%2C%20we%20set%20up%20an,will%20stop%20to%20prevent%20overfitting.&text=After%20about%2025%20epochs%2C%20the,that%20the%20model%20converges%20quickly

Mukherjee, A., Singh, A., & Vardhan, S. (2023, May). 034adarsh/stock-price-prediction-using-LSTM: This project is about predicting stock prices with more accuracy using LSTM algorithm. for this project we have fetched real-time data from yfinance library. GitHub. https://github.com/034adarsh/Stock-Price-Prediction-Using-LSTM

mplappertmplappert 54111 gold badge44 silver badges44 bronze badges, Santanu_PattanayakSantanu_Pattanayak 37622 silver badges44 bronze badges, mpratmprat 25122 silver badges1010 bronze badges, PeterPeter 7, & rigorigo 16144 bronze badges. (1961, June 1). Guidelines for selecting an optimizer for training neural networks. Data Science Stack Exchange. https://datascience.stackexchange.com/questions/10523/guidelines-for-selecting-an-optimizer-for-training-neural-networks

Xiu, D. (n.d.). Empirical asset pricing via machine learning - Dacheng Xiu. https://dachxiu.chicagobooth.edu/download/ML.pdf

Mengmeng Ao, Li Yingying, Xinghua Zheng, Approaching Mean-Variance Efficiency for Large Portfolios, The Review of Financial Studies, Volume 32, Issue 7, July 2019, Pages 2890–2919, https://doi.org/10.1093/rfs/hhy105

Some icons are from: Flaticons: Constraint icons created by Freepik - Flaticon

Investment Strategy

Feature Selection

The Fuel

Machine Learning Model

The Machine

Mean Variance Optimization

The Outputs