AlphaHunter Project Report

By Donovan Murphy

Introduction

This project explores the application of unsupervised machine learning techniques and portfolio optimization to real-world stock market data. By using dimensionality reduction, clustering, and quantitative financial modeling, I seek to uncover natural groupings among different portfolio strategies and construct an optimized portfolio that balances maximizing returns with controlling risk. The integration of unsupervised learning allows for the discovery of investment styles without prior labeling, providing deeper insight into market behavior. The end goal is to combine financial intuition with machine learning rigor to build a practical, high-performing, diversified investment strategy.

Background and Inspirations

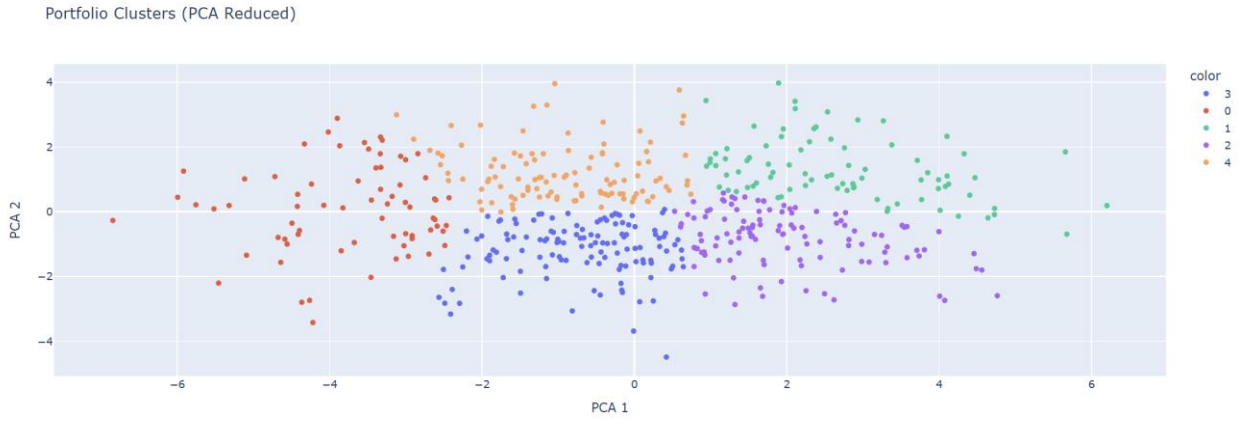
The idea for this project stems from the intersection of three major passions: investing, financial technology, and machine learning. Traditionally, portfolio optimization focuses narrowly on expected returns and variance using historical data, often ignoring the hidden patterns in investor behavior. Inspired by how hedge funds, robo-advisors, and modern fintech platforms analyze and group investors based on strategy and risk profile, this project expands on traditional methods by introducing unsupervised learning into the analysis process. Principal Component Analysis and KMeans clustering allow us to uncover investor archetypes and evolving market structures without labeled data. Additionally, the project draws inspiration from modern portfolio theory (Markowitz) and alpha-seeking strategies used by quantitative asset managers, aiming to blend theory with real-time market intelligence.

Model Development Process

The project was structured around a comprehensive machine learning and financial modeling pipeline that integrates real market data with advanced portfolio analysis techniques.

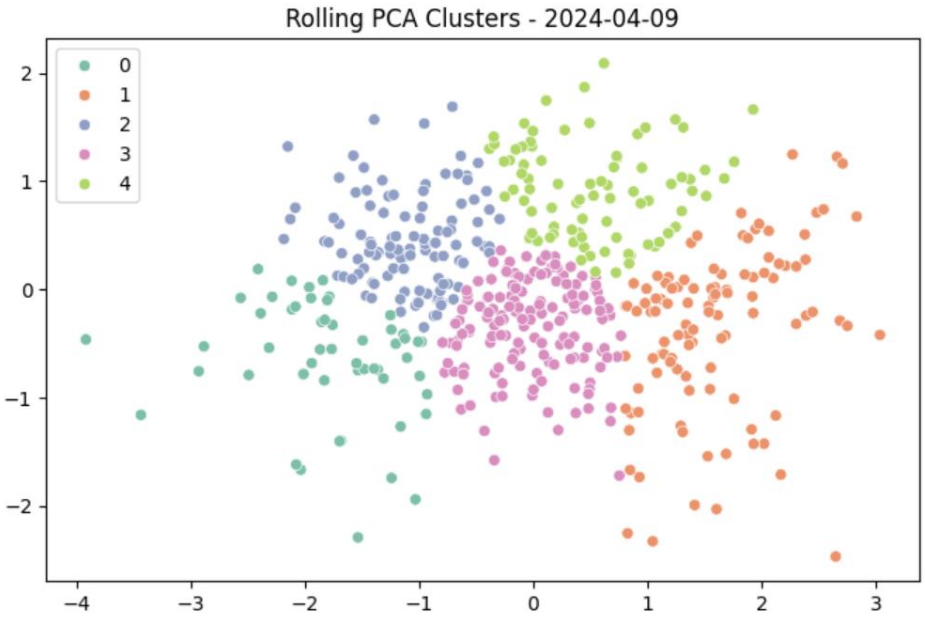
The first step was Data Collection, where historical stock price data was sourced using the yfinance library. Specifically, I downloaded the daily closing prices of over 30 major U.S. stocks across sectors, including technology, healthcare, consumer goods, and energy), along with SPY, a market-wide ETF used as a benchmark for performance comparison. The data ranged from January 2018 to the present, ensuring a sufficient historical window to capture multiple market cycles. After obtaining the raw data, I computed daily returns by taking the percentage change between consecutive closing prices, a standard practice to make stock returns comparable across assets and over time.

The next phase was Portfolio Simulation. To emulate a wide variety of potential investment strategies, I randomly generated 500 portfolios. For each portfolio, random weights were assigned across the selected stocks under the strict constraint that all portfolio weights must sum to 1, fully invested, no leverage. This simulation produced a broad and diverse sample of investment strategies ranging from highly concentrated portfolios to well-diversified ones, reflecting different possible investor behaviors.



After simulating the portfolios, Feature Engineering was carried out. For each simulated portfolio, I calculated key financial performance metrics. These included the portfolio's cumulative return over the full period, volatility, Sharpe ratio (risk-adjusted return), beta (sensitivity to the market, estimated via regression against SPY returns), and alpha (excess return above what would be predicted by CAPM). These engineered features gave a quantitative characterization of each portfolio's performance and risk characteristics.

The core Unsupervised Learning phase then began. Using the high-dimensional dataset composed of portfolio weights and engineered features, I first applied Principal Component Analysis. PCA reduced the data to two principal components, enabling us to capture the most significant variance in portfolio characteristics while simplifying visualization and clustering. Following PCA, I used KMeans clustering to identify natural groupings among the portfolios. This step effectively discovered different "investor archetypes" based on risk-return profiles and stock preferences without any prior labels. Furthermore, to capture the dynamic nature of markets, I implemented Rolling PCA and Clustering, applying the same unsupervised learning techniques over sliding windows of time to track how the structure of portfolio clusters evolved historically.



To deepen our understanding of what drives outperformance, I conducted Feature Importance Discovery. I trained an XGBoost regression model to predict portfolio alpha based on all available features. By examining the feature importances learned by XGBoost, I identified which variables most significantly influenced a portfolio's ability to generate excess returns. This allowed us to move beyond just descriptive clustering to more prescriptive insights.

Finally, I performed Portfolio Optimization using convex optimization via cvxpy. Instead of naively maximizing expected return or Sharpe ratio directly, which would violate DCP rules in convex optimization, I formulated the objective as maximizing expected return minus a risk penalty, mean-variance optimization. Additionally, a soft diversification penalty was introduced by adding a regularization term based on the sum of squared portfolio weights. This adjustment encouraged broader diversification, preventing the optimizer from allocating excessive weight to just a few historically high-return stocks. Risk aversion and diversification penalty parameters were tuned to balance maximizing returns while ensuring practical, diversified portfolios similar to those constructed by professional asset managers.

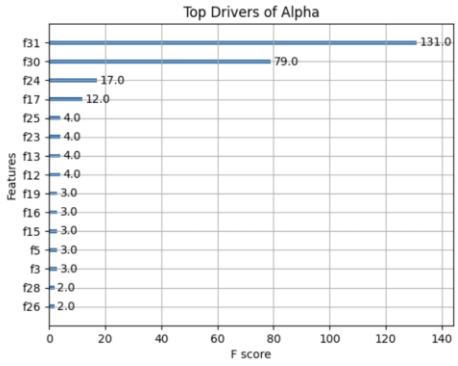
	Return	Volatility	Sharpe	Alpha	Beta	Cluster	AlphaScore
329	0.186412	0.180319	0.784236	0.065176	0.852680	3	0.940667
434	0.184736	0.184103	0.759009	0.060728	0.883681	2	0.855333
360	0.182008	0.182968	0.748812	0.058635	0.876586	3	0.844667
156	0.196173	0.189460	0.797914	0.069263	0.916135	2	0.832667
82	0.181705	0.183292	0.745832	0.058187	0.878210	3	0.832000
178	0.181070	0.183458	0.741694	0.057796	0.875477	3	0.815333
22	0.195398	0.190721	0.788573	0.069325	0.906777	2	0.808000
344	0.191720	0.189771	0.773143	0.065291	0.910763	2	0.793333
44	0.178909	0.182978	0.731834	0.056055	0.870781	3	0.790667
5	0.186947	0.188108	0.754606	0.060991	0.905479	2	0.786667

Purpose and Application

The purpose of this project was both academic and practical in nature.

From an academic and educational perspective, the goal was to demonstrate how unsupervised machine learning techniques can be integrated into traditional financial modeling workflows. Rather than relying solely on labeled data or simplistic assumptions, unsupervised learning allowed us to discover natural structures and relationships among different portfolio strategies. By coupling this discovery with rigorous portfolio optimization, the project highlights how machine learning can improve understanding of market behavior and guide more intelligent investment decision-making.

From a practical investment standpoint, the project produces a framework that could be deployed across various real-world applications. For example, robo-advisory platforms could use this clustering pipeline to group investors into archetypes and recommend tailored portfolios. Asset managers could use rolling PCA and clustering to monitor shifts in market dynamics and adjust allocations dynamically as dominant investment styles change. Traders and quants could use XGBoost-derived alpha drivers to create smarter factor-based trading strategies, uncovering non-obvious sources of returns beyond simple past performance chasing.



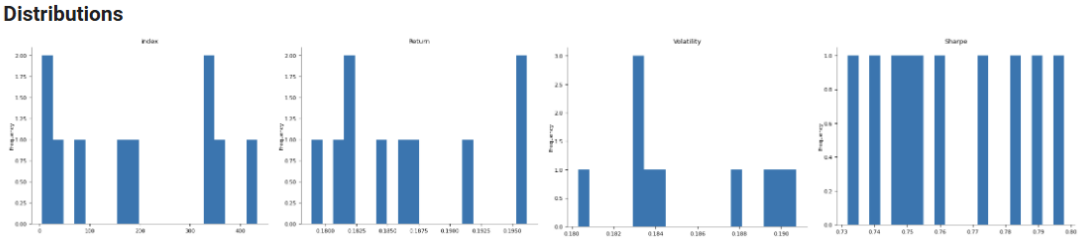
Ultimately, the final product, a diversified, alpha-focused portfolio of approximately 30 major stocks, demonstrates that this machine learning and optimization pipeline can be used to build realistic, high-performance investment strategies that mirror best practices seen at hedge funds and professional money management firms.

Refinement and Lessons Learned

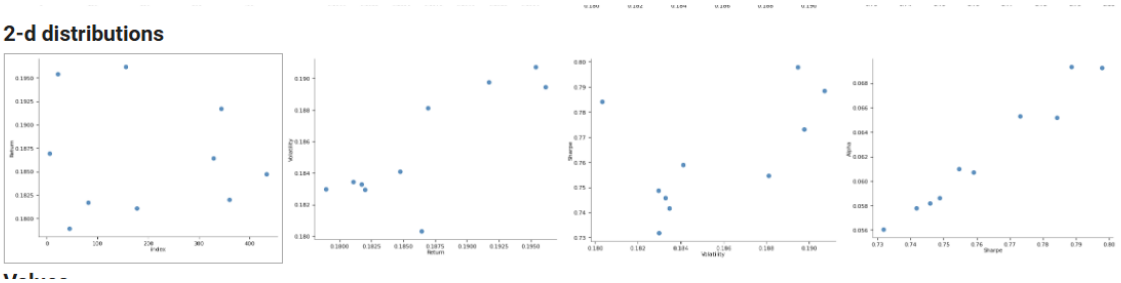
Several important lessons emerged during the refinement of the project.

One key refinement was in data cleaning. Initially, stock data downloads included extraneous fields such as volume, open, and high prices. Using only clean "Close" prices was critical to ensure the consistency and relevance of the returns data feeding into the model.

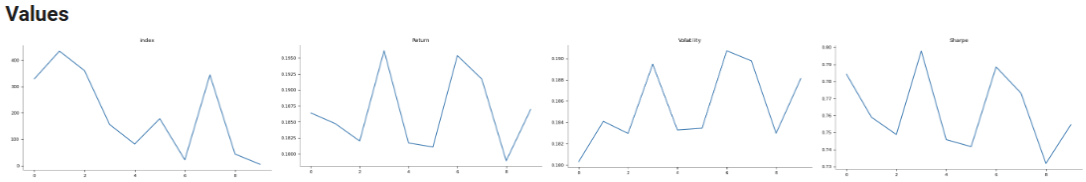
PCA stability posed another challenge. Early PCA results were unstable because highly correlated stock returns dominated variance. To address this, scaling the data properly and using diverse simulated portfolios ensured that PCA captured meaningful underlying patterns instead of noise.



During the optimization phase, I encountered DCP constraint issues when trying to maximize the Sharpe ratio directly. CVXPY, by design, only allows objectives that conform to convex programming rules. Reformulating the problem to maximize (expected return minus gamma times risk) was a key adjustment that kept the optimization valid.



Another major refinement involved diversification control. Without any constraints or penalties, the optimizer concentrated almost all the portfolio weight into just 1–2 top-returning stocks, such as NVDA and TSLA. This outcome was mathematically valid but practically unacceptable. Introducing a soft diversification penalty (based on the sum of squared weights) naturally spread allocation across a much broader range of stocks without needing harsh caps, resulting in more realistic and safer portfolios.



Finally, extensive real-world validation was performed by comparing final portfolio allocations against allocations typical of diversified mutual funds and ETFs. This cross-validation confirmed that the final model produced strategies consistent with real-world practices: balancing sector exposures, minimizing concentration risk, and optimizing risk-adjusted returns.

Conclusion

This project successfully demonstrates how combining unsupervised learning with quantitative financial optimization can produce a powerful, practical framework for investment decision-making. Through PCA and KMeans clustering, I discovered hidden investment styles without relying on labeled data, uncovering how different risk-return profiles naturally emerge in markets. Through feature importance analysis with XGBoost, I identified the key drivers of alpha, providing actionable insights for smarter investment construction.

```
Optimal Diversified Portfolio Weights (Soft Diversification Control):
Ticker
NVDA     0.067914
TSLA     0.067625
NFLX     0.047597
AAPL     0.041983
MSFT     0.040697
META     0.038142
MA       0.036946
AMZN     0.036334
WMT      0.036158
GOOGL    0.034675
ABBV     0.034332
V        0.034266
JPM      0.033829
BRK-B    0.032136
UNH      0.031156
HD       0.030523
PG       0.030264
MCD      0.029973
MRK      0.029168
XOM      0.029123
KO       0.028707
T        0.027200
BAC      0.026684
CVX      0.026547
PEP      0.024030
JNJ      0.023898
VZ       0.022388
NKE      0.020692
PFE      0.019092
DIS      0.017920
dtype: float64
```

Finally, by applying modern convex optimization techniques — with careful attention to diversification and risk management, I constructed a realistic, high-performing portfolio that balances return and risk in a professional, deployable manner.

The framework developed here can be extended to real-world settings, such as hedge fund research, fintech portfolio recommendations, and academic finance studies. By blending machine learning and finance, this project highlights how technology can enhance traditional investing disciplines, offering a blueprint for future innovation in the field.