Unsupervised Learning: K-Means N-Clustering

Allen Zagorodnyuk

May 2025

Introduction

K-Means N-Clustering is an unsupervised machine learning process that groups unlabeled data sets into a specified number of clusters (k). This process works by iteratively assigning each data point to the nearest cluster center, denoted as a centroid, and then recalculating the centroid based on the assigned points. The process continues until the cluster assignments stabilize.

FreeCodeCamp offers an unsupervised trading strategy which focuses on the 150 most liquid stocks in the SPY 500, and uses K-Means N-Clustering to group the stocks into 4 different clusters. The strategy focuses on a 5-year training period from 2017 to 2023 and uses *Garman-Klass, RSI, Bollinger Bands, ATR, MACD*, along with *Fama-French* risk measures and monthly returns to successfully cluster the data.

My goal is to improve this strategy by adding new indicators, testing different stock sizes, and adding a Hedge Position. The new indicators I will add are a *Fast Stochastic Oscillator, P/E Ratios, and Variance-Ratio Statistic.* The different stock sizes will test how the strategy performs against the 100 vs 250 most liquid stocks in the SPY 500, and a hedge position will be included by taking short positions.

To measure the success of these changes, I will compare the overall % return of the 100 and the 250 stocks strategy against the initial 150 strategy. Additionally, I will test this strategy over a larger time frame, ensuring success.

Initial Strategy and ML Process

Focusing on the initial strategy, we start with downloading SPY 500 stock data from 2015 to 2023 into a data frame. Next we add all the necessary indicators:

- **Garman-Klass Volatility:** An estimate of a stock's true daily price variability that used the high, low, open, and close prices to consider intraday swings and overnight jumps.
- **Relative Strength Index (RSI):** A momentum oscillator that scales recent gains and losses into a (0,100) range to signal overbought or oversold conditions.
- **Bollinger Bands:** A moving average flanked by upper and lower bands set at a fixed number of standard deviations, in order to highlight periods of high and low volatility as prices push or retreat from boundaries.
- **Average True Range (ATR):** Measures the average of the true range over a given window to quantify market trends and assist in setting sensible stop levels.
- **Moving Average Convergence Divergence (MACD):** Tracks the difference between two exponential moving averages, alongside its signal line to identify shifts in trend.
- **Dollar Volume:** Multiply trading volume by price to reveal how much money is exchanging hands.

After adding these indicators, we convert daily time series data into monthly data to improve training speed and use the dollar volume metric to calculate the 5-year rolling average of each stock in the SPY 500. We use this metric to pull the 150 most liquid stocks in the SPY 500. Then we use *pct_change(lag)* to calculate monthly returns identified by a lag factor, which denotes our time horizon.

Using our new data, we add *Fama-French* factors that show how exposed our data is to common risk factors. These exposure levels are calculated through linear regression, and are measured as *MktRF, SMB, HML, RMW, and CMA*. Using these new factors, we compute rolling betas using *RollingOLS* and add it back into our main data frame. Now, the data is ready to be clustered using K-Means N-Clustering.

After using K-Mean N-Clustering with clusters set to 4 and centroids set to target RSI values of 30, 45, 55, and 70, we have created 4 data clusters, which we use to back-test a portfolio based on maximizing the Sharpe ratio. Of clusters 0-3, we will focus on cluster 3 as it considers of stocks with the highest RSI levels. Using *pypfopt*, we import necessary packages and define an *opt_weights* function which uses the returns, covariance, and the Efficient Frontier package to calculate the weights that maximize the Sharpe ratio. Using these maximized weights, we plot our back-tested portfolio against the SPY 500 and measure returns as a percentage.

Thought Process and Assumptions

My first idea for improving the initial strategy was to test whether additional indicators would affect how the K-Means N-Clustering process would cluster the stocks, and whether or not this would result in better returns. To test this, I included:

- **Money Flow Index (MFI):**

$$TP_t = \frac{H_t + L_t + C_t}{3}$$

$$MF_t = TP_t \times V_t$$

$$\mathrm{MFR}_t = \frac{\sum_{i=0}^{n-1} PositiveFlow_{t-i}}{\sum_{i=0}^{n-1} NegativeFlow_{t-i}} \quad , \quad \mathrm{MFI}_t = 100 - \frac{100}{1 + \mathrm{MFR}_t} \quad \in [0, 100].$$

  How it works: Take the average of the high, low, and close of the day, then multiply that by volume to get the raw money flow. The flow of each day is considered positive or negative depending on whether the price rose or fell, and the sum of positive and negative flows over the period forms the money-flow ratio. The ratio is converted into an oscillator ranging from (0, 100).
  What it shows: Distinguishes true buy/sell pressure from price moves on low volume.

- **Rolling Skewness and Excess Kurtosis**

$$r_{t-i} = \ln(C_{t-i}) \; - \; \ln(C_{t-i-1}),$$

$$\bar{r} = \frac{1}{n}\sum_{i=0}^{n-1} r_{t-i}, \quad \sigma = \sqrt{\frac{1}{n-1}\sum_{i=0}^{n-1}(r_{t-i} - \bar{r})^2},$$

$$\mathrm{Skew}_t = \frac{1}{n}\sum_{i=0}^{n-1}\Big(r_{t-i} - \bar{r}\sigma\Big)^3, \quad \mathrm{Kurt}_t = \frac{1}{n}\sum_{i=0}^{n-1}\Big(r_{t-i} - \bar{r}\sigma\Big)^4 - 3.$$

  How it works: The daily log returns are computed for each date, then over a window n, the mean and standard deviation of these returns are calculated. Skewness measures the average of the cubed standardized deviations, which captures whether the distribution is lopsided towards one direction, and kurtosis is the average of the fourth-power standardized deviations, which measures how heavy the tails are compared to a normal distribution.
  What it shows: Captures the shape of returns, identifying crash-prone or rally-prone tail behavior that volatility metrics alone cannot.

- **Fast Stochastic Oscillator (%K, %D)**

$$L_n = \min_{0 \le i < n} L_{t-i}, \quad H_n = \max_{0 \le i < n} H_{t-i}$$

$$\%K_t = 100 \times \frac{C_t - L_n}{H_n - L_n} \quad , \quad \%D_t = \text{SMA}_m(\%K_t) \quad .$$

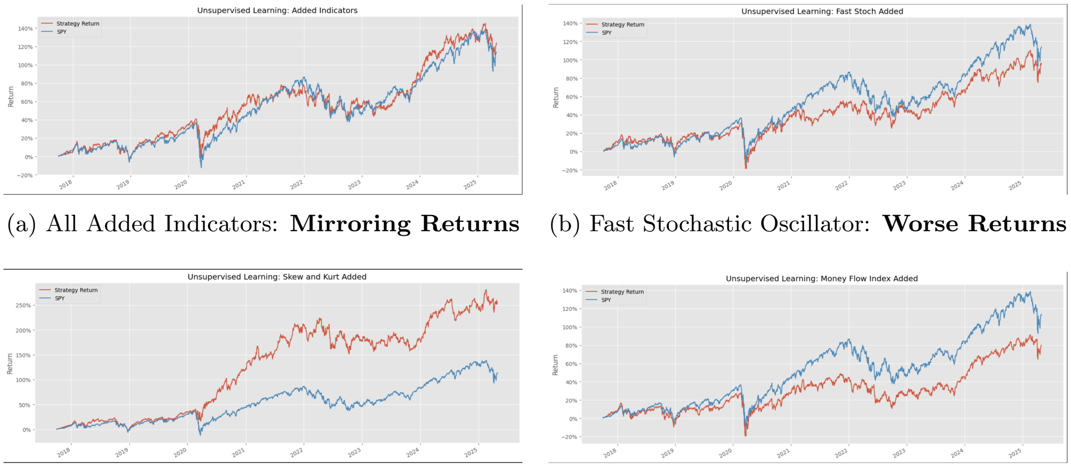
How it works: Over a window n, the highest high and lowest low are captured. Then the close is expressed as a percentage of that range, which produces %K, which is then converted to get %D. These two oscillators move between 0 and 100 to represent short-term overbought and oversold market conditions.
What it shows: Detects very short-term oscillations and overbought/oversold conditions.

I tested each indicator separately, and all three together to capture their impact on the overall return. Next, I tested the most profitable of these strategies across an increased time frame to capture whether or not these increased returns would continue. Following this, I tested whether the amount of stocks considered by the K-Means N-Clustering process would have an effect on the returns. To test this, I ran the strategy by focusing on the 100 most liquid stocks and the 200 most liquid stocks in the SPY 500. My assumption was that compared to initial set of 150, focusing on 100 stocks may provide better returns, as this group of stocks is most likely made up of the best and largest companies in the SPY 500, but may see more volatility; while focusing on 200 stocks will spread out our risk and may lower returns but will provide a more stable rate of return.

Additionally, I assumed that a hedge position (short) may be profitable, as the initial strategy only considered longs. For the hedge position, I focused on cluster 0, which was made up of stocks with the lowest RSI levels, and I shorted those companies. I ran a K-Means N-Clustering on this new stock list to create an optimized short portfolio and integrated it into my long portfolio. To test different intensities of hedging, I considered a %5, %10, %15, and %20 levels of hedge as a percentage of the long portfolio.

Data & Results

Out of the three additional indicators, the only one to generate a positive return after being added to the cluster was Rolling Skewness and Excess Kurtosis. Tested over the initial 150 stocks, this added indicator was able to slightly increase the overall return. Below you can see charts after each additional indicator is added, and the chart after all three indicators are used at once.



(a) All Added Indicators: **Mirroring Returns**

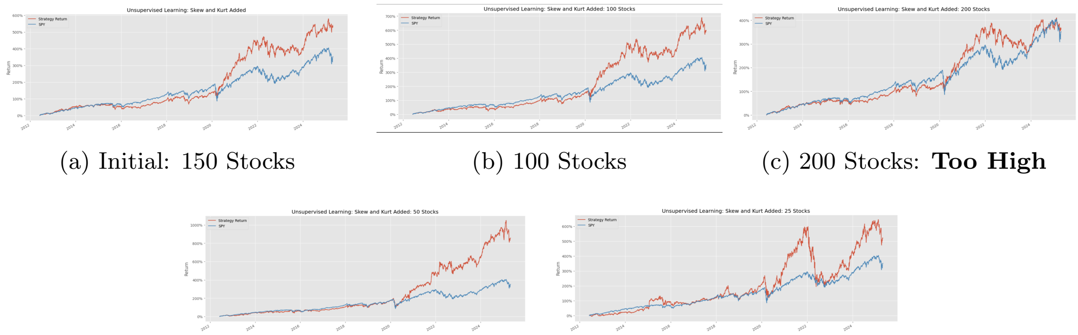
(b) Fast Stochastic Oscillator: **Worse Returns**

(c) Skewness and Kurtosis: **Best Returns**

(d) Money Flow Index: **Worst Returns**

Figure 1: Additional Indicator Results

Continuing with Rolling Skewness and Excess Kurtosis, the results of changing the number of stocks considered are that as you consider a decreased number of stocks, a higher return may be generated, but at the cost of increased volatility. When we consider that the centroid for the K-Means N-Clustering algorithm is set based on RSI, this environment exposes our portfolio to the risk of having little to no stocks meet our RSI requirement, which can cause an inability to maximize the Sharpe ratio and an increased exposure to one specific stock. However, if we increase the amount of stocks, we spread out too much and lose any advantage our strategy generated, thus, we end up with returns matching that of the overall market. Below you can see charts representing the effects of changing the stocks considered compared to the initial 150, these figures are represented over a larger time frame from 2012 to 2025.



(a) Initial: 150 Stocks

(b) 100 Stocks

(c) 200 Stocks: **Too High**

(d) 50 Stocks: **Best Returns**

(e) 25 Stocks: **Too Low**

Figure 2: Stock Consideration Results

Regarding our hedge assumption, because our K-Means strategy always recalibrates into 4 clusters based on the RSI distributions, the members of those clusters can change every rebalance. At the same time, any stock that drops out of the SPY 500 is replaced by another, and over the long run, the index tends to grow. Under those conditions, trying to hold a persistent short position in “surviving” SPY 500 stocks contradicts the assumption of overall market growth, so it isn’t compatible with this clustering framework.

Conclusion

In conclusion, based on my assumptions, the most profitable changes that could be made to this K-Means N-Clustering trading strategy are the addition of a Rolling Skewness and Excess Kurtosis indicator, along with a decrease in the stocks considered by the algorithm from 150 to 50. The addition of a Rolling Skewness and Excess Kurtosis assisted in the K-Means N-Clustering and resulted in slightly better returns, and the decrease in the number of stocks considered narrowed our portfolio and generated far better returns. However, it is important to note that the decrease in stocks considered is specific to our SPY 500 data, as the SPY 500 is a Large Cap Index and contains stocks with High RSI and High Dollar Volume. If attempted across other smaller indices, there is no guarantee this strategy will generate the same style of return without first back-testing.

Percentage Return of Initial (2012 to 2025): **533.27%**

Percentage Return of Updated (2012 to 2025): **858.02%**

Key Takeaways

- **K-Means N-Clustering:** K-Means N-Clustering is an unsupervised machine learning process that can use multi-indicator features to effectively cluster data and represent market opportunities, however, careful feature collection is required to ensure stable and meaningful clustering.
- **Adding Indicators:** Looking at our results, only 1 of 3 features results in an increased return compared to our initial strategy. This result indicated that adding too many or redundant indicators can dilute signal strength and lead to diminishing returns.
- **RSI Clustering Basis:** Continuously recalibrating clusters based on RSI, risks overfitting to patterns. Thus, regular validation against new data/periods is essential. An additional initial centroid could be added to K-Means to prevent overfitting and create new clusters.
- **Potential Future Projects:** Future work could explore a hedge position, where instead of focusing on ”surviving” SPY 500 stock, we could track stocks that are removed from the SPY 500 and short them. This removes the s̈urviving” bias and doesn’t go against the basic assumption of overall market growth.

Click Here To View Code