February 2, 2021

Get Your Hands off My Money! – A Machine Learning Approach to Asset Allocation

In the 1950's Harry Markowitz introduced the financial world to Modern Portfolio Theory (MPT). The engine behind modern portfolio theory is called mean variance optimization (MVO). MVO is a simple optimization problem whereby a practitioner seeks to identify efficient asset allocations based on expected returns (mean), and risk (variance), subject to any number of constraints about position size, long/short rules, leverage, and others. The theoretical importance of this idea was strong enough to win Markowitz a Nobel Prize in 1990. However, MPT in its original form has one glaring problem…it's not particularly applicable to the real world. To be clear, most technical issues that MPT presents for real world practitioners are manageable, it's the theoretical ones that are more problematic. Particularly problematic is the use of expected values for means, and variances/covariances, after all there is no crystal ball.

The following document is divided into two sections. The first section focuses on supervised machine learning methods, the second on unsupervised methods. Both sections include a brief discussion of the features used, a discussion of any transformations or augmentation of the data, a description of the machine learning methods used, and an evaluation/discussion of both machine learning outcomes and simulated real world applications of the model. The intent is to go beyond simply showcasing various machine learning methods and explore the results. Instead, each model is structured as a pipeline that begins with raw data and ends with a portfolio construction and management process developed using machine learning methods, and back tested using real historical data. As such models are evaluated both from a machine learning perspective and from a portfolio performance perspective. In other words, was the machine learning method additive in a real-world environment?

Supervised Methods.

It is often said that trying to time the stock market is a fool's errand. In fact, there is a very well-known theory in the financial world that describes the stock market as a random walk. The premise behind the random walk hypothesis is simple. Asset prices evolve randomly and therefore cannot be predicted. Assuming for a moment this theory holds, it begs the question, "what exactly are financial professionals doing all day?" At some point during their career every seasoned asset manager concludes that investing is not about prediction, it is about probabilities. To use the parlance of the data science world this is a classification problem not a prediction problem.

Formally, the first model proposed seeks to answer one question, ***"given the most up to date macro economic data and financial market data is it possible to accurately assess the probabilities of an adverse event in the financial markets?***"

In practice, the model takes in high impact macro/market data as features and uses supervised classification models to create a signal. Next the model creates a stock and bond portfolio that is actively managed and rebalanced using the signal alone.

Data and features

1. Shiller's PE ratio. Also known as the cyclically adjusted price to earnings ratio (CAPE). This multiple gauges value in the stock market. It is like the well-known and commonly used price to earnings ratio. The difference is that the price to earnings ratio generally uses the previous 12 months of earnings data, whereas Shiller's PE uses 10 years of earnings adjusted for inflation. The theory behind the CAPE ratio is that economies works in cycles, and while in the short term earnings can be very noisy, over the span of 10 years, they are more stable once adjusted for inflation. As such the CAPE ratio should provide a stronger indication regarding a firm/market's earnings power.
2. Slope of the Yield Curve. The slope of the yield curve is calculated by subtracting the yield of a three-month United States Treasury bill from the yield of a United States 10 year Treasury Note. There is strong empirical evidence supporting the idea that the slope of the yield curve alone is an immensely

powerful predictor of looming recessions (Luca Benzoni, 2018). Simply put, an inverted yield occurs when the short-term bond yield is higher than the long-term bond yield. When this happens, economic trouble is usually on the way. That being said there is no clear consensus on why or when the yield curve inverts, or even how long short-term yields must be above longer-term yield for the yield curve to be considered truly inverted. Some theories include the idea that market practitioners expecting The Federal Reserve to lower rates to combat a slowing economy, attempt to ensure generally more attractive yields by buying longer duration bonds ahead of interest-rate cuts. This drives bond prices up and bond yields down. When this happens enough the 10-year yield drops below the 3 month and an inversion occurs.

3. <u>Core inflation as measured by the Consumer Price Index (CPI)</u>: The Consumer Price index is a measure of inflation in consumer goods. This index is released each month by the US Bureau of Labor Statistics. It is meant to reflect the changes in the prices of goods and services purchased by consumers in the United States. The standard Consumer Price Index has some problems. Particularly, because it includes a few items that have very volatile prices like food and energy. To deal with this the BLS also put out the Core Consumer Price Index that excludes both these categories and provides a more stable and consistent measure of inflation.
4. <u>Unemployment</u>: Unemployment is for the most part considered a lagging indicator. The highly publicized non-farm payrolls are announced on the first Friday of every month and provide the general public with information about the jobs market from the month before. In other words, market practitioners and economist generally don't expect to see unemployment rise until after trouble has started. This may raise questions about the usefulness of the metric given the task at hand. Why then is it included? The theory of full employment clearly states that structural or frictional unemployment may remain even when the economy is at for employment, in fact the OECD believes that the full employment unemployment rate in the United States ranges between about 4 and 6 1/2% (OECD, 2000). In other words, even when there is a job available for everyone, some individuals will still be out of work. Given the fact that economies are cyclical, it follows that when an economy reaches its full employment unemployment rate for a prolonged period, it has potentially peaked. From an economy's peak there is generally only one way to go. So instead of looking for lower unemployment numbers as confirmation that the jobs market is still healthy the model uses one minus the unemployment rate as a potential indicator that the jobs market has peaked. If this number reaches levels of 95%+ it could be a sign of a slowdown.
5. <u>ISM purchasing managers index (ISM-PMI)</u>. This is often the first major piece of information economists and market practitioners receive every month regarding the health of the economy. The index is based on the survey conducted monthly by the Institute for Supply Management. The survey includes questions regarding production, hiring, supplies, deliveries, and inventories. The index is adjusted for seasonality. A PMI above 50 indicates expansion in the manufacturing segment of the economy. A PMI below 50 indicates a contraction.
6. <u>Personal income and outlays</u>. This Number is reported monthly by the Bureau of Economic Analysis and tracks consumer income and spending. Historically household consumption which represents the C in the well known C (consumption) + G (Government spending) + I (Investment) +[X-M] (Exports - imports/ net exports) formula for gross domestic product, averages about 68% of total GDP in the United States. In other words, 68% (CIA, 2021)of economic activity in the United States depends on consumer's wallet and how much he/she is willing to spend. As such we expect that this metric can provide insight into a slowing economy.
7. <u>The University of Michigan Consumer Sentiment Index.</u> This index is a measure of consumer confidence. It is published monthly by the University of Michigan. It's stated objectives are among others:

   1. To assess near term consumer attitudes on the business climate personal finance in spending.
   2. To gauge the economic expectations in probable future spending habits of consumer.
   3. To judge the consumers level of optimism/pessimism.
   4. To promote an understanding of into forecast changes in the economy.
8. The monthly survey consists of 50 core questions and includes at least 500 telephone interviews across the contiguous United States (excludes Alaska Hawaii and other US territories) (Wikipedia University of Michigan Consumer Sentiment Index , 2021)
9. NBER business cycle (**Model Label**)- The data used to create the label for the model is published monthly by the national Bureau of economic research. It is a binary index. If the economy is an expansion the index reads zero, if the economy is in recession the index reads one.

Additional Data:

In addition to the features described above the model used the asset prices to create a portfolio.

1. The Vanguard 500 stock index fund : The Vanguard 500 stock index fund is the first of its kind and the oldest passive investment vehicle still active. The fund began trading in August 1976 and currently has About 637 billion dollars in assets under management. This is incorporate in the model as a vehicle for investment in the broad stock market.
2. The Vanguard total bond market index fund: Also the first of its kind this fund provides exposure to the broad US bond market it began trading in December 1986 and is incorporated in the model as how an investor would invest in the bond market.

Future preparation

Access

All the raw data was imported via the datastream web services API. This is a subscription based API (the models run off included csv files to ensure continuity but all notebooks contain he original code used to access the data for evaluation). The requests asks for since inception data using monthly intervals for all macro and financial market information discussed previously. This data is stored in CSV file for further use. A second call to the data stream web services API imports the total return index for both Vanguard funds. This data is also stored in a CSV file for later use.

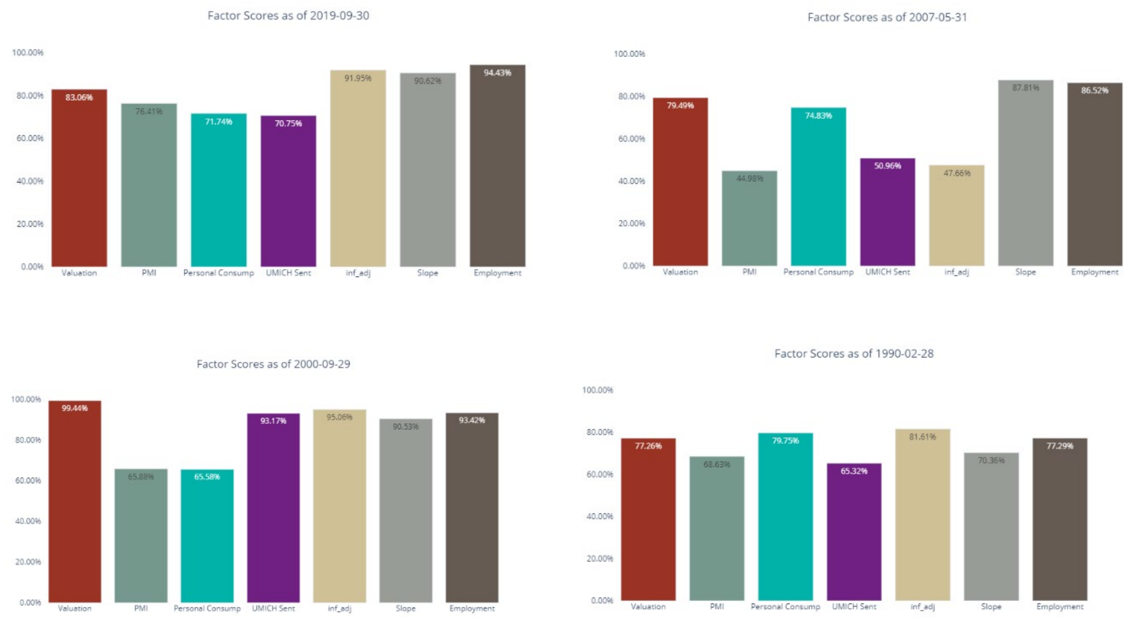
Feature Preparation and Processing

Label alignment The signal the model creates must be forward-looking to be of any value. An indicator that accurately identifies that the economy is in recession two months after the recession has started is in a word…worthless. After shifting the label data back by 12 months the features are now aligned with an event in the future. For example, The data indicates that the United States economy officially entered a recession in February of 2020, this is represented in the raw data as with a one next to the time stamp February of 2020, by shifting all observations back 12 months and marking February 2019 with a one instead, and fill forward missing data with zeros. The decision to fill missing data with zero stems from the fact that economies are usually NOT in recession. The model will now evaluate conditions a year before a known recession.

Rolling normalizations: Constructing an indicator using timeseries data that is normalized over the entirety of a timeframe can lead to misleading results. Consider for a moment the Shiller's's PE ratio a feature used in the model. This indicator reached an all-time high in December 1999 during the dot com bubble. The issue is that every month between April 1998 in December of 1999 was also a new high. Standard normalization across the entire data set would not capture this properly because when the information was reported a practitioner would have drawn conclusions based on each of those months being the all-time high. As such as new information is reported the mean of the information changes the standard deviation changes and every observation's standard score changes. To account for this, the model transforms the data using an expanding normalization process with a 36-month minimum observations window. This process works as follows:

1. Calculate the expanding mean (the mean of all data that came before and including a given observation)
2. Calculate expanding Standard deviation.
3. Subtract a given observation from its corresponding expanding mean and normalize by the expanding standard deviation.

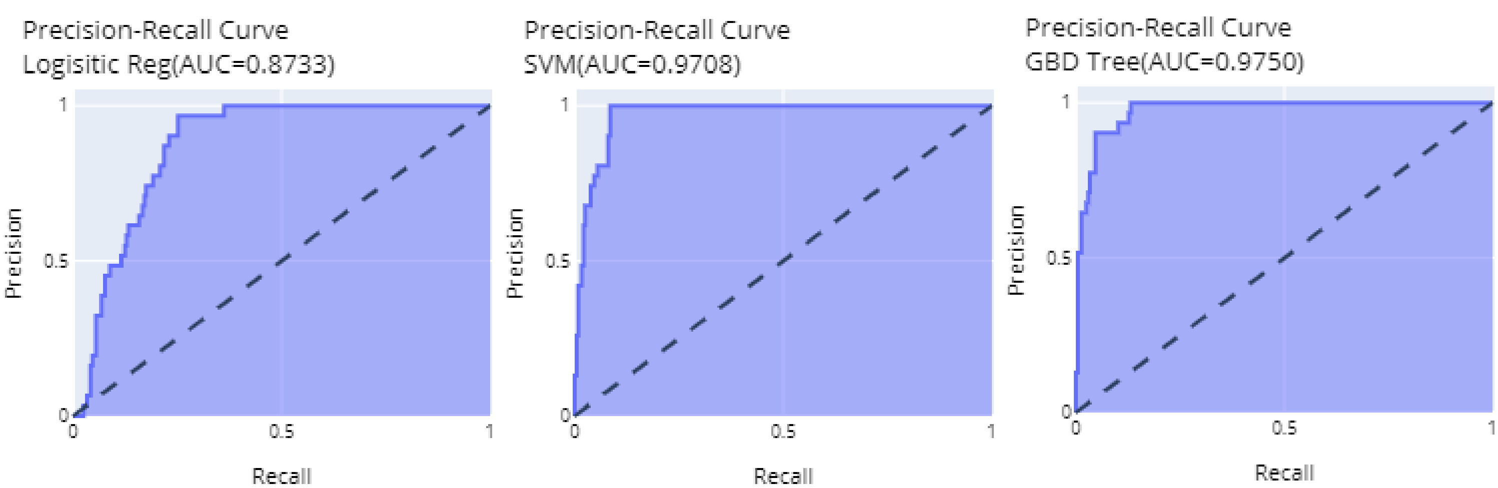
This results in only the last observation being normalized using ALL the data.



(A comparison of feature p values during various economic peaks)

Train - test split This is the standard SK learn train test split function using a test size parameter of one half of the dataset. This is necessary to reduce potential sampling errors that result from the fact that recessions are on one hand not common, but perhaps not uncommon enough to justify oversampling solutions.

Model Selection training



Three classifiers were considered for this model. These are logistic regression, support vector machine classifiers, and the gradient boosted decision tree classifier.

The procedure followed for selecting a classifier was:

1. Establish parameter ranges for evaluation for each model
    1. C parameter for support vector machines and logistic regression the parameters tested were [0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]
    2. Learning rate for gradient booster decision trees, the parameters tested were [0.0001, 0.001, 0.01, 0.1 , 1, 10]
2. Run Grid search with 3 folds for each classifier.
3. Models with the best results were selected for comparison based on the ROC AUC chart and the and F1 Scores.

Based on the grid search The best learning rate for the gradient boosted classifier was 0.1. Specifically, for gradient boosted classifier the model ran in additional tuning for the best number of estimates (weak learners) The model identified 200 as the ideal number. These parameters produce a precision recall curve with an area under the curve of 0.9636. The Best C parameter for the logistic regression was 0.2 This however produced a precision recall curve with an area under the curve of only 0.89 far less than the GBD classifier. The best C parameter for the support vector machines classifier had a value of 1. The support vector machine classier produced the best results in terms of area under the curve at 0.9713 slightly better than the gradient boost classifier and F scores were comparable. The decision to continue with gradient booster decisions classifiers came down to the fact that gradient booster decisions classifiers work better with non-standardized data and the standardization process employed here is not conventional.

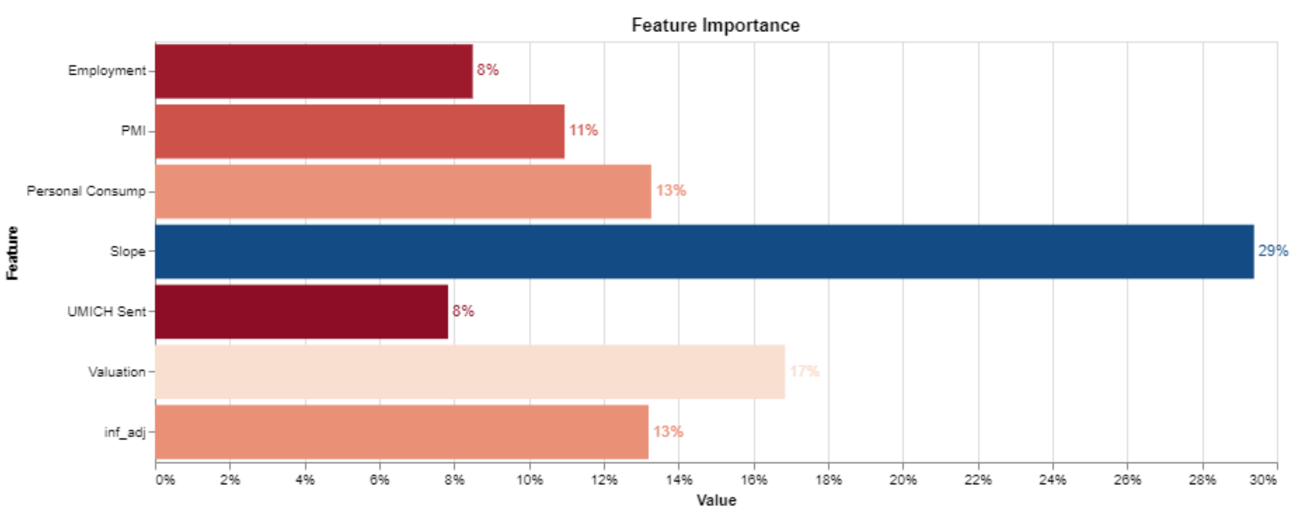
As mentioned earlier Gradient boosted classifiers are sensitive to two parameters, The first is the number of estimates the classifier will use and the second is the learning rate. The number of estimators corresponds to the number of small decision trees used in the model, whereas the learning rate controls the emphasis on fixing errors from one decision tree to the next. These parameters were selected during the classifier selection process and kept for the remainder of the process. Once the model was trained, fit, and applied to the test data, the probabilities of a recession were extracted using <predictproba_> method.

<u>Discussion of results</u>

Model scores

For this model minimizing false negatives and false positives was more important than the overall accuracy of the model. As such the F1 statistic was the most logical choice to use as a test statistic. It's important to remember that the intent behind the model is to assist in making decisions about portfolio allocation and as any seasoned portfolio manager knows, avoiding mistakes is exponentially more important than making good calls (and no they are not the same thing). The F1 micro statistic for the test data was 0.939.

Feature importance



(Feature Importance)

Not surprisingly, The slope of the yield curve posted the strongest feature importance score at 29%. This was followed by valuation at 17%, inflation personal consumption and purchase managers index we're fairly similar with each between 13% and 11%, lastly employment and University of Michigan sentiment indicators both came in at 8%.

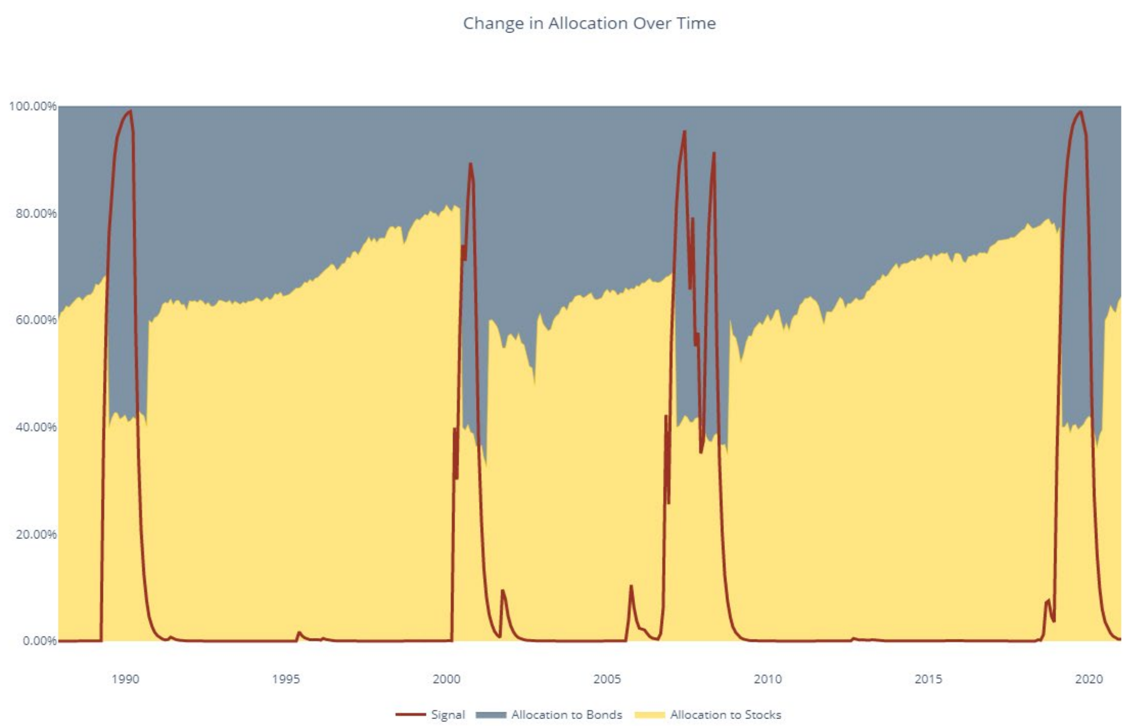
The results observed make sense. Consider the strongest two features in terms of feature importance (slope of the curve and valuation), both have to do with financial markets, and both are forward-looking mechanisms that discount future expectations to today's terms. In other words, both features are constantly providing information in real time about what to expect in the future. As such, when these two features reach certain levels the implications for the future given what is currently known is likely accounted for.

Real world application.

The initial indicator had a few issues. First, the movements in the raw indicator were very sharp. The probability of a recession in the next 12 months often went from extremely low to extremely high very quickly. This issue was mitigated somewhat by taking the exponentially weighted moving average of the raw indicator with a span decay of four. This helped create a smoother signal with less of a whipsaw effect.

Second it was immediately clear that the indicator does a particularly good job of warning the user that a recession is coming when a recession is 12 months away. However, as time goes on the model begins to pick up on the fact that a recovery is now 12 months away, often before the expected recession even begins. This is likely the result of the notion that the average recession post WWII lasts about 11 months (NBER, 2021), whereas the average expansion lasts 5.3 years. Unfortunately this could pose a real problem for portfolio managers attempting to use this indicator as it stands. Simply put, the sell signals will occur early and in advance of a recession as desired, but subsequent buy signals will occur as the recession is starting and well head of the desired entry point. To be clear this is not a problem with the model rather it clarifies how to use the model. In other words, there is no need to change the model, but there is a need to account for this possible outcome when putting the model to use.

Asset allocation



(The changes in asset allocation as the signal gets stronger/weaker)

At this stage in the process the model has provided reasonably good results, it has taken financial market and macro economic data and classified it each observation (a given month in time) as recession or not recession, and also provided the probability of each outcome. The exponentially waited moving average of the probability output from that model acts as an indicator that a portfolio manager could use to make changes to his/her asset allocation. While the model evaluation has established that the indicator can provide reasonable insight into the probability of a recession beginning in 12 months, It has not yet been established that the making decisions based on the indicator would add any value. To that end, The next step in the process is to simulate portfolio performance using a rules-based process for allocating capital that revolves solely around the indicator.

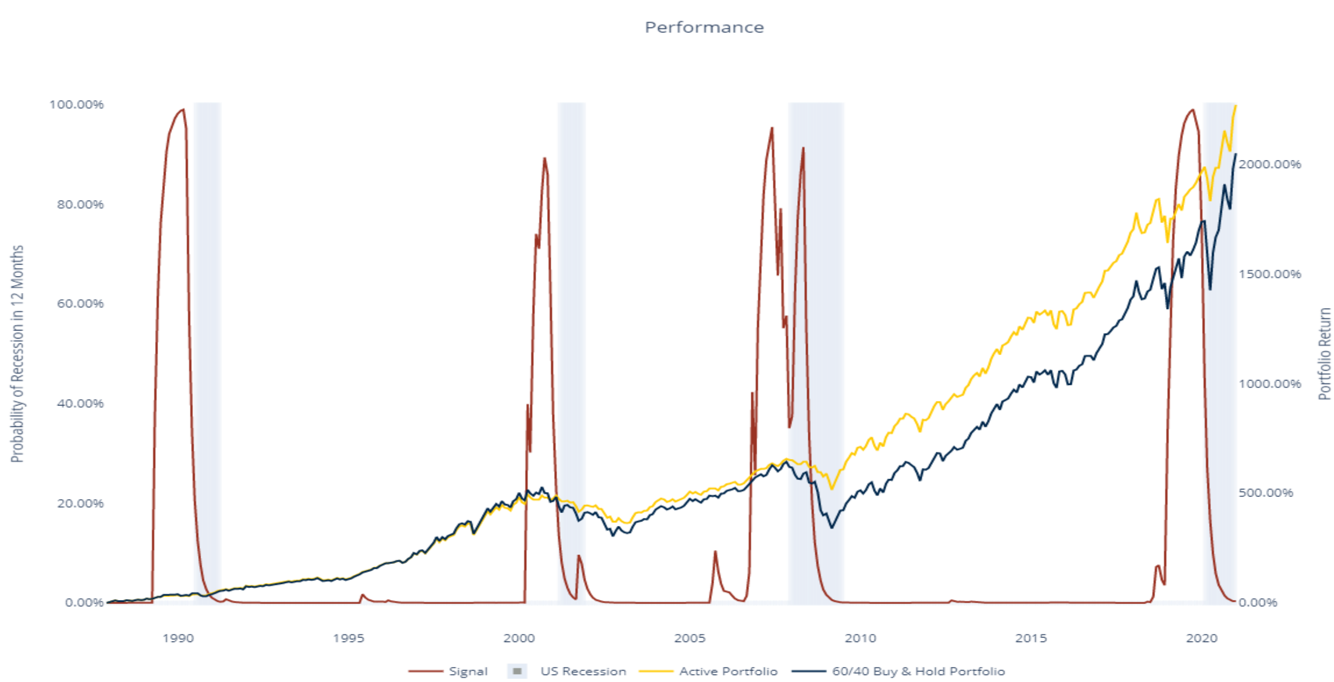
Portfolio Construction

The portfolio used in the simulation is a simple passive portfolio that consists of stocks and bonds alone. The use of passive vehicles eliminates any undesired affect from security selection or sector allocation and allows for the truest evaluation of the decisions made by the indicator.

The portfolio construction process works as follows:

1. Start with a standard 60% stock 40% bond portfolio
2. If the indicator is at or above 0.7 (70%) at any given time over the past six months, The portfolio is rebalanced to 40/60 stocks/bonds.
    1. This rule helps reduce the impact of early re-entry described earlier.
3. Once the indicator has not read a level of 0.7 or higher for six months The portfolio is rebalanced back to 60/40
4. The allocation is shares based, as such no changes are made to the portfolio as long as there is no signal from the indicator. This means that shares are calculated the moment The indicator gives a signal, and the number of shares held is constant until the next signal is given by the indicator.

(Portfolio performance with the signal and recession overlaid)



The back test assumed that trading occurred in monthly intervals between November of 1987 and December of 2020. Performance was evaluated in two ways. First and probably most obvious is the portfolio's cumulative performance over time in comparison to a 60/40 buy-and-hold portfolio. A 60/40 buy-and-hold portfolio is simply a portfolio that allocates 60% to Stocks and 40% to Bonds and just holds the same amount of shares in both without changes or rebalancing. Over the period in question the portfolio created using the indicator had a total return of 2172.12%, where is the buy-and-hold portfolio only posted a 1951.7% total return. In addition to absolute performance, the Sharpe ratio was used to evaluate performance. The Sharpe ratio is probably the most common metric used to evaluate portfolio performance on a risk-adjusted basis. Calculation of the shower for ratio is as follows:

$$\frac{[E(\bar{r}) - rf]}{\sigma}$$

(The average annual return of a portfolio minus the average risk-free rate divided by the portfolios standard deviation.)

The intuition behind the Sharpe ratio is straightforward. Investors are interested in maximizing return for every unit of risk taken. As such by isolating only the portion of returns coming from risky assets and then normalizing it by the portfolios volatility which is a proxy for risk, investor can understand how much return for every unit of risk they are getting. As such the higher the Sharpe ratio the better. In this case, the portfolio constructed using the gradient boosted decision tree indicator achieves a portfolio Sharpe of 0.91 whereas the 60/40 buy-and-hold portfolio only achieved a Sharpe ratio of 0.69 indicating that the model constructed here appears to add value over a 60/40 by and hold approach.

Unsupervised Methods

PCA based Risk Parity

Perhaps one of the great greatest challenges in the quantitative portfolio management is the delicate balance between the need to be forward-looking and the complexity and difficulty needed to produce strong forward-looking models. Making good predictions about future asset prices is very difficult, practitioners are never operating with all the information, market participants do not always act in a rational ways, and exogenous events expected to occur once every hundred years seem to happen every 7 to 8 instead. Even if one could make strong predictions about asset prices, this is not enough. Equally important and exponentially more difficult is making predictions about covariances/correlations between assets. Using standard econometric tools when attempting to project joint distributions of many assets into the future is an extremely difficult task, and often times requirements are relaxed and stationarity and normality are simply assumed. This often leads to less-than-optimal results. This issue begs the question, "***can machine learning methods be used to reduce the complexity of projecting asset covariance/correlations?"***

Principal components analysis

This model proposed seeks to take advantage of two key features of principle components analysis.

1. Dimensionality reduction- The ability to deal with less data without sacrificing much from the results is very beneficial as it allows users to focus on information that is truly impactful on the decision making process.
2. Orthogonal Feature extraction - The very process of principal components analysis results in data sets that are by definition not correlated to each other. This eliminates the need to project covariances or correlations when doing forward-looking work.

This model takes a series of asset prices from various asset classes, decomposes their returns into principal components, and selects the minimum number of principle components that can cumulatively explain 95% of variance among asset classes. Once this is complete the model projects the volatility of each of those principal components over a given time horizon. The projected volatility combined with the eigen vectors (not normalized) of the Principle components produces a new forward looking covariance matrix. This new covariance matrix is fed into a risk parity asset allocation program to produce a portfolio.

Features

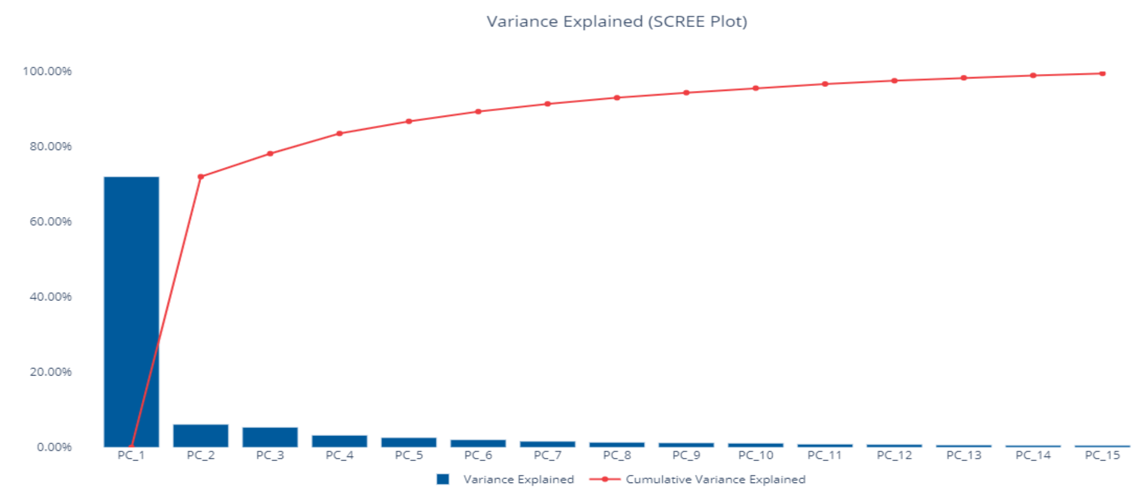
The features for this model include daily prices from the following exchange traded funds as proxies for various asset classes. Given the fact that all of the assets are equity ETFs with similar behavior the PCA model was done using raw returns. To be clear, a comparison of total variance explained by PCs showed that less than a one percent difference in variance explained by PC1 with standardization, vs PC1 using raw returns.

Access

All data for the ETFs above was accessed via the Refinitive Eikon API (this was formerly called Thomson Reuters). As in the previous section the data is imported and stored in a CSV file for later use. The models in the notebooks run by reading the csv files but the functions used to call the data are still in place should they be needed Approximately 22,700 observations were collected for this model.

All prices are converted to percentages and any not a number (nan) observations that result from this process are replaced with zero.

Principal component selection.



The following method selects the number of principal components included in the model:

1. Train and Fit a principal components analysis on all the data.
2. Evaluate the explained variance ratio for the results
3. Select the minimum number of principal components required to explain 95% of variance among features.
4. Train and fit a new principal components analysis based on the established number of principal components.

Volatility projection

Once the principal components analysis is complete, the model proceeds with volatility projections. To do so the employs the generalized autoregressive conditional heteroscedasticity process or GARCH(p,q) process. The GARCH process was developed by an economist named Robert Engle in 1982, Engle won the 2003 Nobel prize for economics for this contribution. The underlying principle behind the GARCH process is that unlike a linear process when dealing with volatility error terms are not assumed constant. This makes any attempt to project volatility using linear methods problematic. The formula for a GARCH process is as follows:

$$\sigma_t^2 = \omega + \alpha\varepsilon_{t-1}^2 + \beta\sigma_{t-1}^2$$

Where σ_t^2 is the conditional volatility, and ε_{t-1}^2 is the error term for the previous period.
This formula is very common amongst financial practitioners and is fairly straightforward to implement in python via the arch package.

Once volatility projection is complete of our principal components, the model constructs a new covariance matrix using the following steps:

1. The 30 day volatility projection for all principal components are arrange in a diagonal matrix (Delta)
2. Multiply the Delta matrix by the transpose of the principal components eigenvectors Matrix (Alpha)
3. Multiply the product of the previous step by the matrix alpha.

The result is a new covariance matrix that accounts for short term expected volatility levels. This matrix is now ready for the asset allocation process. The process described above is called Orthogonal Garch. It was developed by Professor Carol Alexander at the University of Sussex and is detailed a great length in her book "Market Models a Guide to Financial Data Analysis."

Risk parity

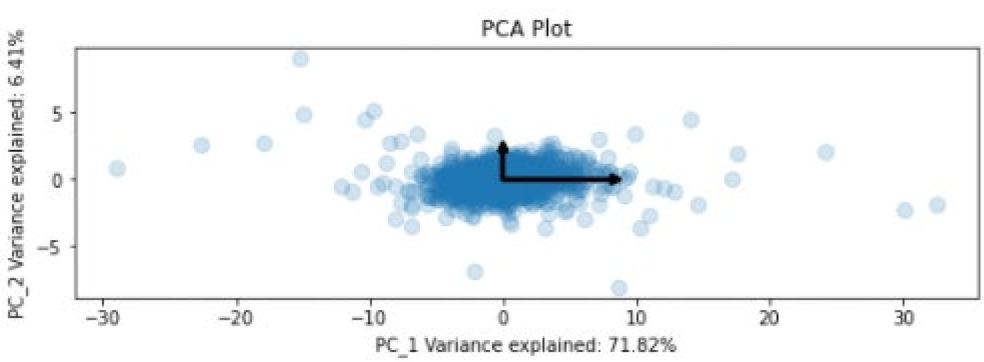
For the most part modern asset allocation techniques are simple (yet frustrating) optimization problems. Modern portfolio theory takes a vector of returns, and a covariance matrix and tries to either minimize variance, maximize return, or maximize return per unit of risk. The output of this process is simply a vector of weights that meet the established goal, given the stated constraints.

Risk parity is a variant of the above-described process except that the focus in risk parity is risk budgeting. The idea is to ensure that assets are contributing to risk equally.
The advantage of using the risk parity model for this situation is that risk parity is entirely independent of expected return inputs. All that matters is the covariance matrix. This makes the model a perfect candidate to evaluate the covariance matrix created using principal components analysis.
The risk parity portfolio construction process works as follows:

1. Begin with an arbitrarily weighted portfolio.
2. Calculate the portfolio's volatility this is calculated as $W'\Sigma W$ where W is a vector of weights, Σ and is a covariance matrix.
3. Calculate each assets contribution to risk
4. Calculate the error term based on these results
5. Adjust weights to attempt to improve on previous iteration

This process is implemented in python via the Scipy stats optimizer - minimize function. The optimization function included in equality constraint to ensure that the sum of weights equal one, and an equality constraint to ensure no short selling.

Discussion of results



Principal components analysis
The data set includes 15 features all of which are financial instruments that tree daily. As such we expected a relatively high degree of explanatory power from principal components. In fact the first principal component explains 71.82% of the variance among assets The next component explains about 6.41% of the data it

takes the next 8 principal components combine to reach the 95% threshold. While it is impossible to tell exactly what each of the principal components are, it is highly likely but the first one represents the global stock market.

Portfolio construction

To evaluate the process overtime we once again ran a back test using the following process:

1. Portfolio trades on pre-establish dates in this case the 15th of every March, June, September, and December. Should this day fall on a weekend The closest trading day is selected.
2. On every trading returns from the previous year are fed into the principal components analysis.
3. The transformed features are extracted and a GARCH(1,1) process is used to project 30 days of volatility for every one of the 10 principal components kept.
4. The collected data is used to create the new covariance matrix as described.
5. The covariance matrix is fed into the risk parity optimization function.
6. Once the risk parity function provides the weights the portfolio is rebalanced.

<u>Evaluation of results</u>

Machine learning results.

The principal components analysis during each iteration performed as expected. Whenever employing principle components analysis our expectation is that we end up with

A) a data set that is smaller and easier to deal with without sacrificing too much information

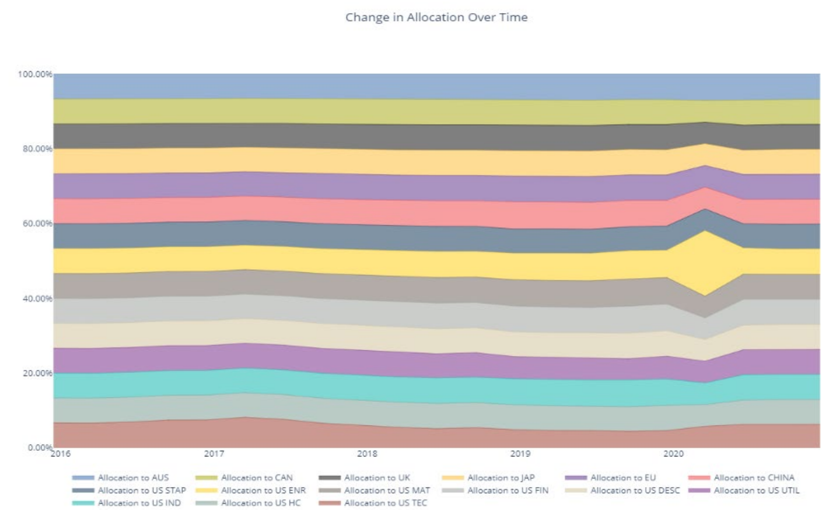
B) that all the principal components are not correlated to each other

Given the fact that we did manage to reduce the Data sets and the resulting data set allowed us to project volatility quickly and easily into the future this method is highly effective for such purposes given the notion that the alternative is far more time-consuming, complex, and does not ensure better results.

Evaluation of real-world application

Just as in section 1 we evaluated our process using The same methodology, The total return of the portfolio over the period in question and the portfolio Sharpe ratio. Unfortunately in this case, the simulation produced less than desirable results. The portfolio sharp was 0.5 vs the MSCI All Country World Index's Sharpe of 0.61.

This lackluster performance is likely attributable to the following factors.



Evaluation of the asset allocation clearly shows that nothing changes much over time, the allocation remains stable. In situations like this the risk parity model may penalize investors by allocating capital to less favorable assets simply because of predetermined risk budgets.
The correlation among assets may be too high. As stated these are all equity portfolios. They are baskets of various stocks based on sector and or country. While there may be some variance among them, the performance chart shows that they generally behave in similar ways, this may not be very suitable for the risk parity process. Potential improvements to the model include changing the mix of asset classes, more frequent trading periods, potentially a different optimization processes. It is important to note that all the changes required stem from domain issues, **not from machine learning/data science issues.**

Clustering

Thus far the models developed have used machine learning methods as tools to assist in the asset allocation process. The first model used a supervised machine learning model to develop an indicator that provides portfolio managers with a trading signal. The second model used unsupervised methods to help decompose data into principal components thus simplifying the extremely important yet very difficult process of projecting covariances among assets. But the question remains, ***can machine learning methods alone be used to allocate assets and construct portfolios?*** To that end, model number three uses various clustering methods on fundamental, valuation, and momentum data to generate a stock portfolio.

Features

Access

All the data required for this model was accessed via the Refinitiv Eikon API and was stored in a series of CSV file for further use. A quick but important to note: upwards of 250,00 data points were pulled for this model (505 S&P 500 Stocks, 21 features per stock, over 24 quarters), do to the rather large size, pulling the data took over 45 minutes.

Features

The features used in this model all fall into one of three categories

Valuation:

This subset of features focuses on some of the more common valuation multiples used when picking and analyzing stocks.

1. Earnings yield- earnings yield is the reciprocal of the price to earnings multiple. This multiple is calculated by dividing a company’s net income by their market cap or alternatively by using earnings per share and the company stock price. Note, this calculation uses the forward earnings yield which uses analyst expectations for net income as opposed to historical data.
2. Book yield- this is calculated using the company’s total book value and dividing it once again by market cap. This multiple is usually more stable than the earnings yield because the accounting (book) value of

equity is far more stable than a company's year to year earnings. The price to book is often considered by value investors as the most reliable measure of value.

3. EBITDA to EV: EBITDA stands for earnings before interest taxes depreciation and amortization and EV stands for enterprise value. EBITDA is often used as a proxy for cash flow. Enterprise value is calculated by adding a company's market cap to its net debt. (Net debt equals total debt minus cash) to calculate our multiple simply divide EBITDA by EV. The importance of enterprise value-based multiples is that they unlike their market cap equivalents account for the risks associated with leverage.
4. EBIT to EV: This is like the feature above, except that the first multiple does not capture depreciation and amortization, and EBIT more closely resembles operating profits than cashflows. The reason for including both is that companies that operate in industries with a high capital requirement that have significant depreciation expenses May unfairly advantaged via the EBITDA number, or unfairly disadvantaged via the EBIT number.2.00

These features focus on company fundamentals, economic output, and cost of capital.

1. return on invested capital. Return on investment capital is often considered one of the most important indicators regarding the quality, efficiency, and profitability of a businesses. It is calculated as the net operating profit after taxes (NOPAT) divided by A company's total invested capital.
2. Weighted average cost of capital: The weighted average cost of capital is an estimate of what a company has to pay it's investors. It is in affect the financial cost of doing business for a company. The WACC is usually calculated by taking all the components of a company's capital structure, calculating their costs, and calculating a weighted average of those two. There are in most cases considerations for taxes that must be made and costs of certain parts of the capital structure like equity must be estimated.
3. Long-term expected growth. This is nothing more than Wall Street's aggregate estimate for expected growth for a given company over the next five years.

Momentum

This set of features focuses on the price momentum of stocks over different time frames. The features are constructed using 3, 6, and 12 month time frames.

1. Risk adjusted relative momentum: Risk-adjusted relative momentum is calculated by subtracting a stock's total momentum over a given period from the benchmark's (in this case the S&P 500) risk-adjusted return over the same timeframe. To adjust for risk, we multiply the stock's beta coefficient (a coefficient from a linear regression where the benchmark's returns is the X variable and the stock's returns is the y variable) to its benchmark.
2. Relative momentum: Standard relative momentum is simply the returns of a stock over a given period less the returns of the benchmark over the same time frame.

A good stock should have strong relative and strong relative risk-adjusted momentum, whereas a poor stock may not have strong risk-adjusted momentum

<u>Feature prep</u>

Winsorization- Financial data is often plagued with extreme observations. It is not uncommon to see price to earnings multiples in the thousands, stock momentum that has outpaced the rest of the market five or six-fold, or extreme amounts of leverage on a company's balance sheet. The winsorization process helps deal with these problems. Winsorization simply replaces the extreme top and bottom x percent of all observations with the value of the X percentile. In other words assume a range of numbers between zero and 100, after winsorizing the top and bottom 5%, observations 95 through 100 would now equal 95, and observations five through zero would now equal five. This model winsorizes the bottom 5% and the top 20% of all features.

Main mix scaling -The min mix scaling processes is used to normalize features and minimize the impact any remaining anomalies may have on the model.

<u>Model selection</u>

Three different clustering models were considered for this segment. The first was DBScan, Second agglomerative clustering, and the third Kmeans. The clustering models selected was the Agglomerative and the KMeans. DBSCAN produced far too many outliers. This is expected given the high levels of dispersion within the data. Agglomerative and Kmeans produced similar results.

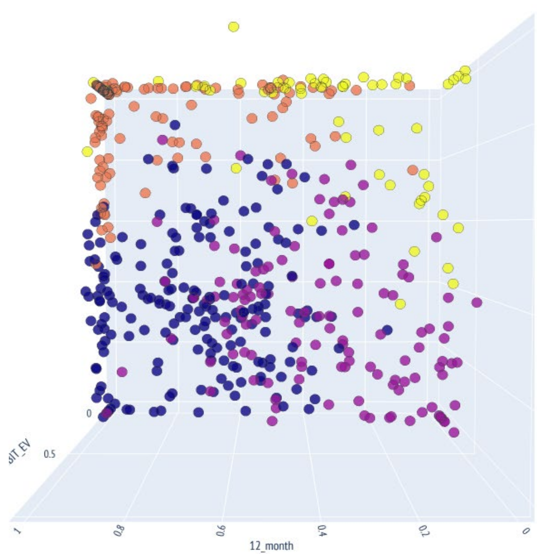
Process

Both models were set to generate 4 clusters. We fit the model using cross-sectional data for all stocks in the S&P 500 at a particular point in time.

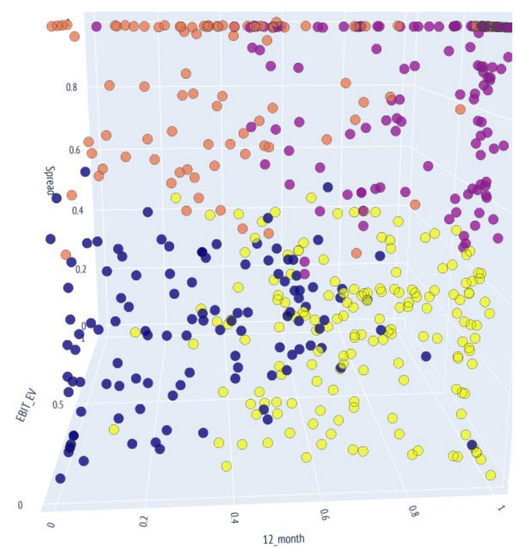
Once the model had created the clusters, the average feature scores for each group were calculated. The group with the highest score, was selected as the portfolio, and the stocks purchased in equal weight. This process repeated every quarter and rebalanced the portfolio every time.

Results

Aglom Clusters

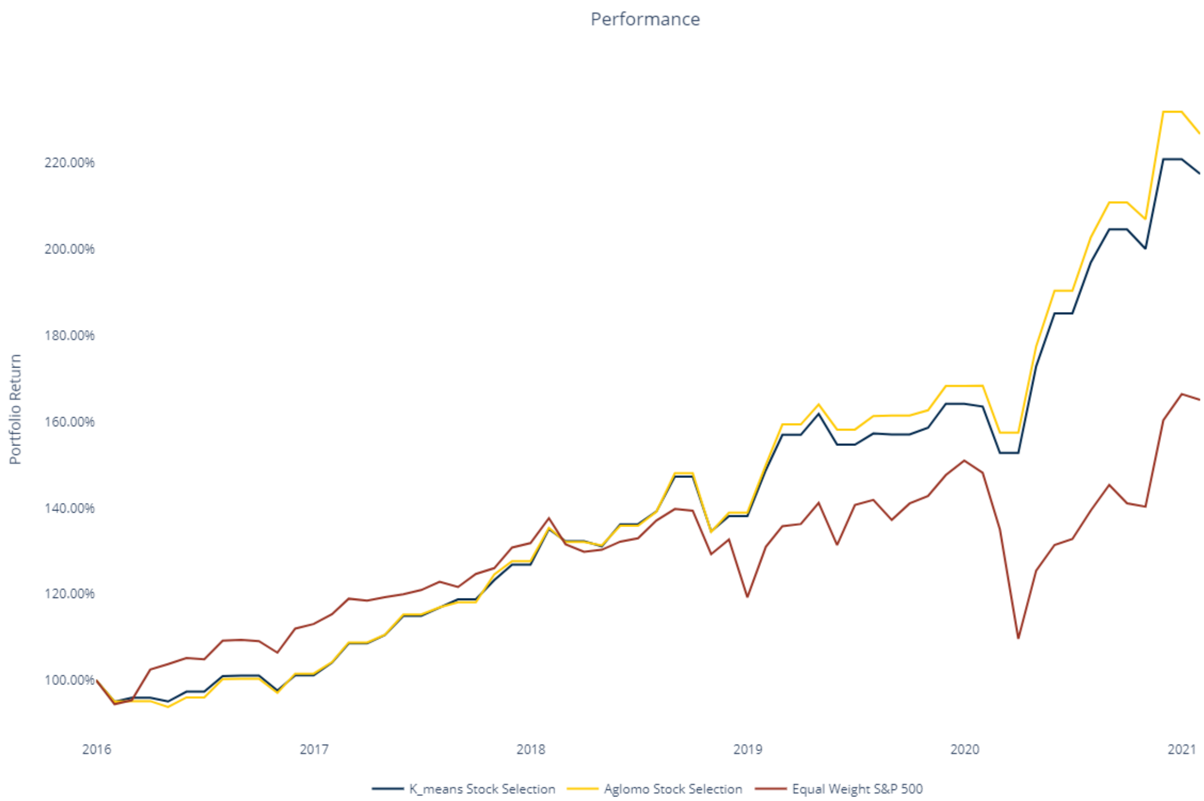


Kmeans Clusters



A 3-D plot of the clusters shows that both The Kmeans and the Agglomerative clearly identified four groups of Stocks. The challenge initially was deciding which group to pick as a portfolio. It is after all never clear which of the three factors that were used as features in this model are in fact driving the market. But further evaluation of the feature data relative to the clusters revealed that for the most part there is always one cluster that is slightly better than the rest in both models.

Real world applications



The KMeans portfolio performed extremely well. Between 2016 in January 2021, The portfolio posted a gain of 117.49% versus 65.08% for the S&P 500 equal weight portfolio. From a risk-adjusted perspective, The Sharpe ratio over the period in question was 5.88 for the Kmeans portfolio in 2.99 for the equal weight S&P 500 portfolio. The agglomerative outperformed the Kmeans slightly over the period in question posting a 126.76% return and a Sharpe ratio of 5.98.

Conclusion

The study set off to explore the potential practicality and usefulness of various machine learning methods for asset allocation. Specifically, the study attempted to answer three questions.

1) Given the most up-to-date macro-economic data and financial market data is it possible to accurately assess the probabilities of an adverse event in the financial markets?
2) Can machine learning methods be used to reduce the complexity of projecting asset covariance and correlations.
3) Can machine learning methods alone be used to allocate assets and construct portfolios?

With respect to question number one. Based on the evidence presented in model number one, One could easily draw the conclusion that supervised machine learning methods particularly classification methods would be very effective in providing ample warning ahead of a recession. Be that as it may, The enemy of the quantitative financial model is regime change, and it remains to be seen how well this models will perform in the future.

With respect to question number two, there is no doubt, that the unsupervised method used to decompose asset returns simplifies the process of projecting volatility greatly. Unfortunately, as model number two showed the

method employed does nothing to improve the quality of a projection. In other words, no matter how effective the principal components analysis was in simplifying, reducing, and rotating the data, garbage in... garbage out.
Lastly with respect to question number three. Can Machine learning message be used to allocate assets? The answer to this is simply yes. The use of Kmeans, agglomerative, or any other clustering algorithm that uses features grounded in economic and financial data, is without question as legitimate and effective a way to allocate assets as any other method out there. In fact the evidence presented from model number three suggest that K means can be effective can in fact be effective. The question is can it be effective persistently and in various market conditions.
The moral of the story here, is that NO model, method or algorithm can compensate for or fix bad data. Furthermore, it is easy to imagine an a board room full of executives in various slides and visualization similar to those found in this report drawing the conclusion that they have found the “Magic Formula”. Truth to power…There is no magic formula, there is not easy way, and No none of these methods are a sure thing. It is important to make sure that they are never presented or understood as such.

Statement of work
I did this project on my own. As I’m certain you can tell from the paper I live breathe eat and sleep Financial markets. This was probably the most fun I’ve ever had putting together a research paper, I hope you find it as useful and insightful as I hope it is.

Works Cited

CIA. (2021). *World Fact Book United States Economu*. Retrieved from CIA World FactBook: https://www.cia.gov/the-world-factbook/countries/united-states/

Luca Benzoni, O. C. (2018). Why Does the Yield-Curve Slope Predict Recessions? *Chicago Fed Letter, No. 404, 2018*.

NBER. (2021). *NBER Business Cycle Dating*. Retrieved from National Bureau of Economic Research: https://www.nber.org/research/business-cycle-dating

OECD. (2000). *REVISED OECD MEASURES OF STRUCTURAL UNEMPLOYMENT*. Retrieved from OECD: http://www.oecd.org/social/labour/2086120.pdf

Wikipedia University of Michigan Consumer Sentiment Index . (2021). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/University_of_Michigan_Consumer_Sentiment_Index