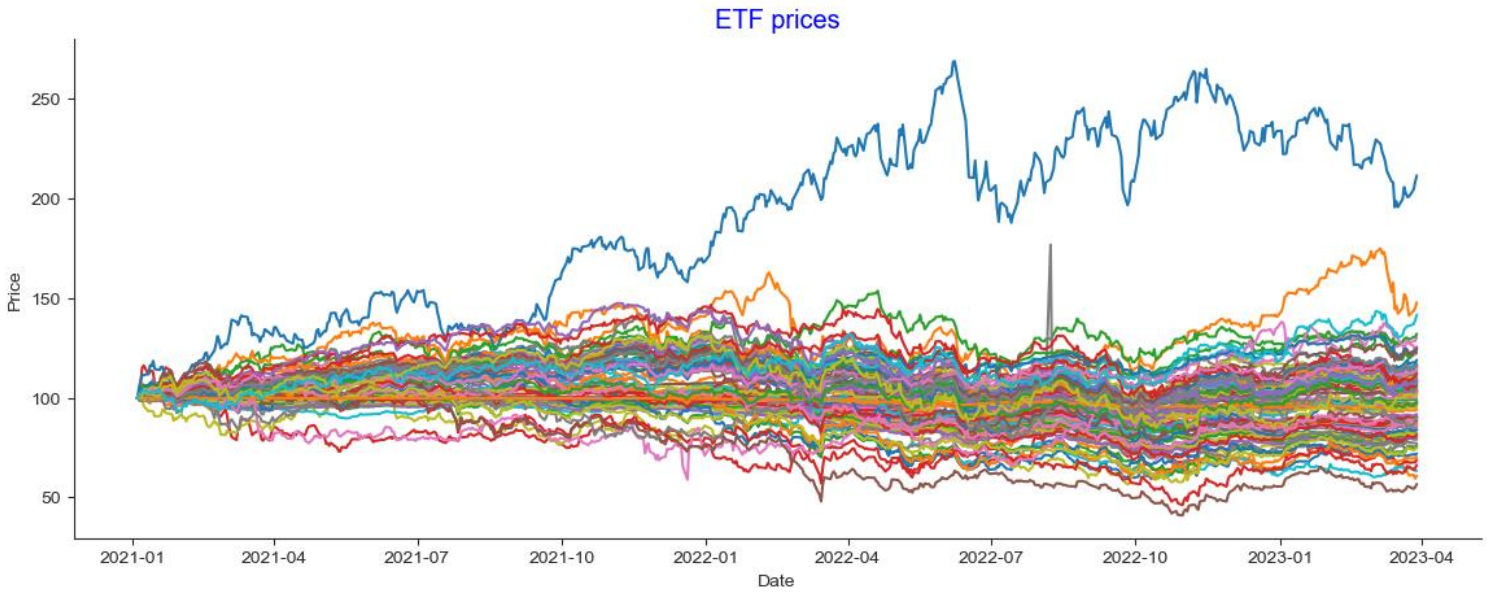
Applied Data Science – ETF classification

MONFORT Baptiste

ETFs price over time visualization:



Overall, all the ETFs share the same behavior staying close to the based price of 100. We can spot one outlier so far which is the ETF_41.

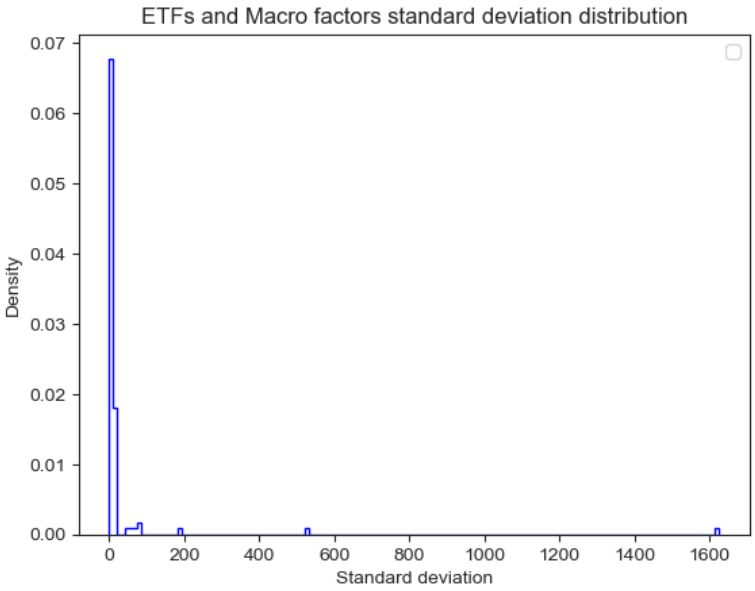
ETFs and Macro factors returns and risk:

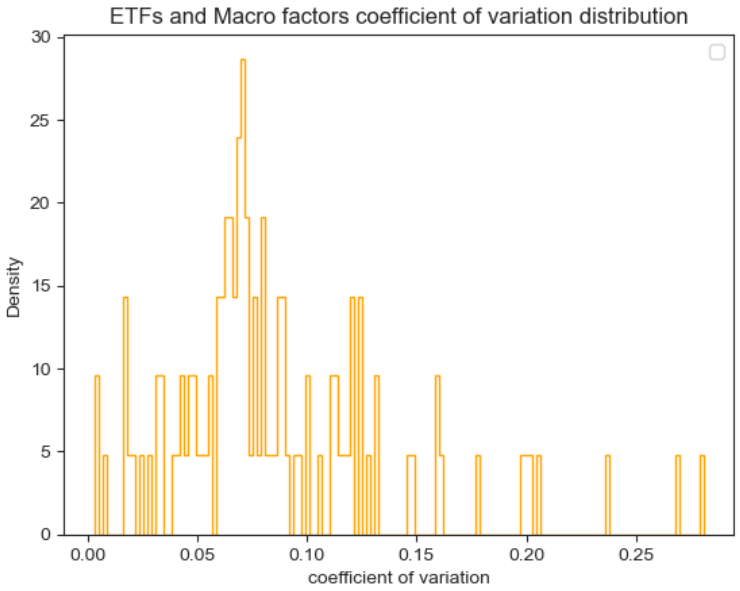
We can see in below left figure that most of the ETFs and Macro factors seem to have similar standard deviation, hence similar historical volatility. The outliers here are some of the Macro factors which have a greater standard deviation. That difference is due to different scales. For instance, the Japan Nikkei 225 has a unit of 45047.04 at starting date whereas ETFs unit is 100. This difference motivates a need for rescaling our data so we can properly compare them.

Rather than looking at standard deviation we will then look at the coefficient of variation, a much better metric when working with different scales. Because the standard deviation of data must always be understood in the context of the mean of the data. In contrast, the actual value of the coefficient of variation is independent of the unit in which the measurement has been taken, so it is a dimensionless number.

Coefficient of variation formula:

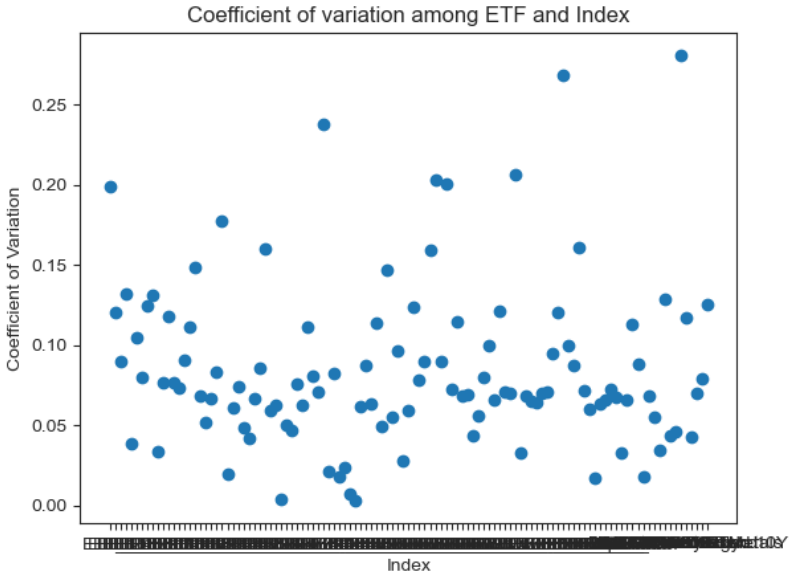
$$CV = \sigma/\mu$$



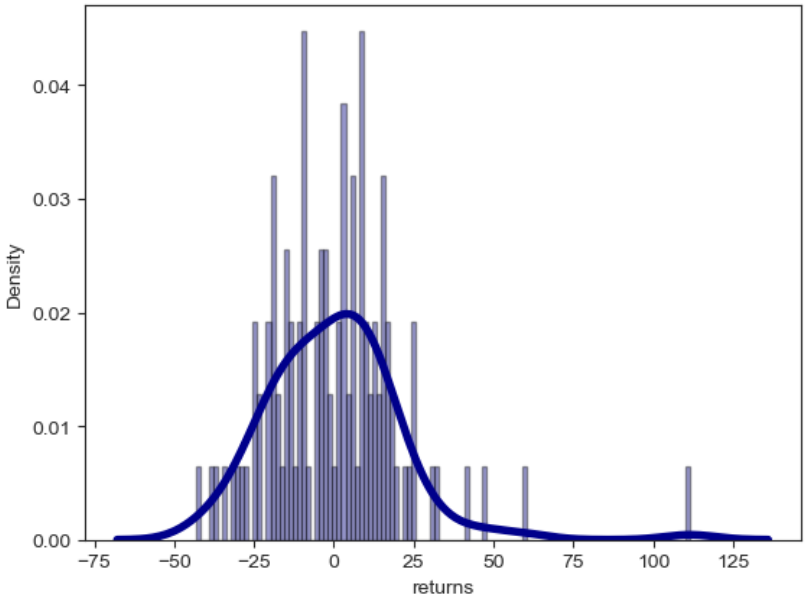


We can now see in above right figure that we have a much better metric for ETFs and Macro factors comparison in terms of their respective volatility scaled by the mean. We can now point out a true accumulation of funds with similar or close volatility as there is one big spike that can be located between 0.05 and 0.10, hence most of the ETFs and Macro factors had between 5% and 10% historical volatility on the period.

Put another way we can see that accumulation line in such coefficient for ETFs and Macro factors.



From the below figure, a similar story tends to appear for the returns. We have a balanced distribution around 0, which is almost symmetrical or slightly left skewed.



Let's have a look now at the distribution of the risk to reward metric, namely, the Sharpe ratio. The Sharpe ratio is defined as the marginal unit of return gained from adding a unit of risk.

Sharpe ratio formula:

$$Sharpe\ ratio = \frac{Return - R_f}{\sigma}$$

First, we need to define what is our risk-free rate. Usually, it is considered that the US 1Y Treasury Bill is a risk-free investment. Hence from the Treasury.gov website we obtained the following risk-free rate quotation as of 01/05/2023: 4.72%.

Select type of Interest Rate Data

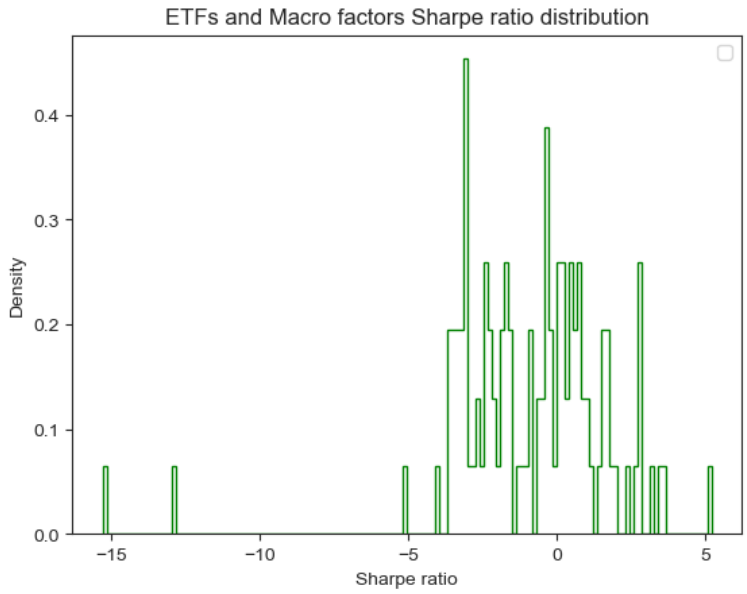
Daily Treasury Bill Rates

Select Time Period

2023

Apply

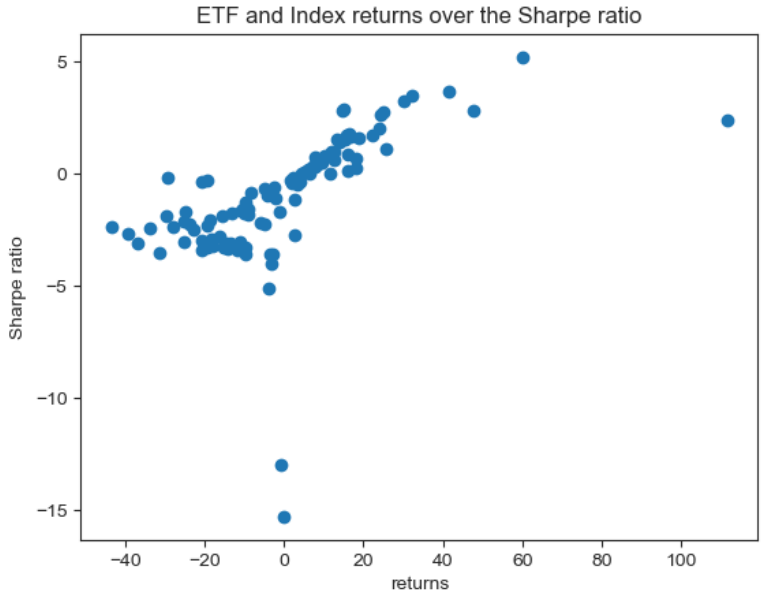
	4 WEEKS		8 WEEKS		13 WEEKS		17 WEEKS		26 WEEKS		52 WEEKS	
Date	BANK DISCOUNT	COUPON EQUIVALENT	BANK DISCOUNT	COUPON EQUIVALENT	BANK DISCOUNT	COUPON EQUIVALENT	BANK DISCOUNT	COUPON EQUIVALENT	BANK DISCOUNT	COUPON EQUIVALENT	BANK DISCOUNT	COUPON EQUIVALENT
01/03/2023	3.96	4.03	4.29	4.38	4.40	4.51	4.57	4.70	4.63	4.81	4.50	4.72
01/04/2023	4.00	4.07	4.28	4.37	4.41	4.52	4.58	4.71	4.64	4.82	4.49	4.71
01/05/2023	4.12	4.19	4.44	4.53	4.51	4.62	4.62	4.76	4.68	4.86	4.56	4.79



Most of the ETFs and Macro factors are in the range [-5:5]. There are some outliers especially some elements with far left on the distribution Sharpe ratio.

Sharpe ratio against returns visualization:

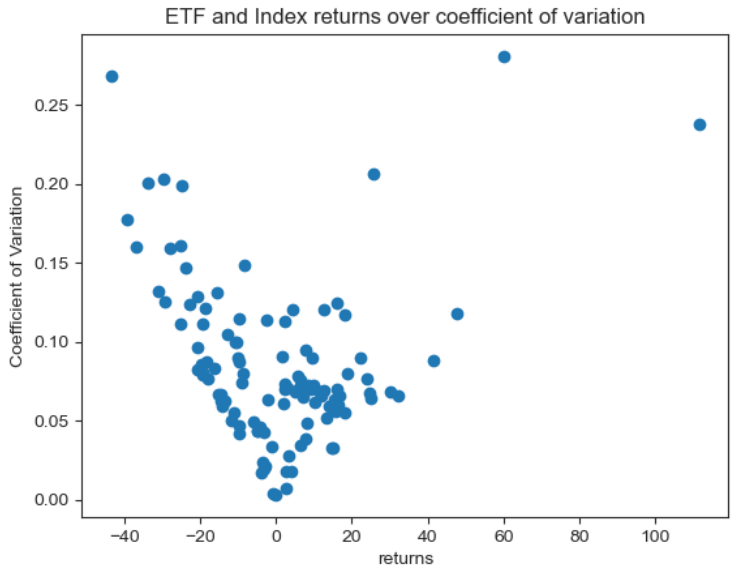
A clear trend emerges where returns seem almost perfectly correlated (hence linear relationship) with risk. However, we do find some outliers. This seems in line with previous visualizations.



Coefficient of variation against returns:

Trends are highlighted here as well. This cone pattern make stronger the claim over the linear relationship between volatility and returns, however in contrast to the Sharpe ratio we can see it goes both ways in positive and negative returns in almost the same magnitude.

Again, some outliers.



Classification of ETFs and Macro Factors through clustering:

In order to obtain a classification of our ETFs, one has to split the data into groups.

In that purpose, we will use cluster algorithm which are unsupervised learning methods of splitting data based on similarities. We will use two different methods of clustering. The K-means and the Gaussian mixture.

K-means clustering is a popular unsupervised machine learning algorithm used for clustering data into k number of clusters based on their similarity, where each data point belongs to the cluster with the nearest mean, called the centroid.

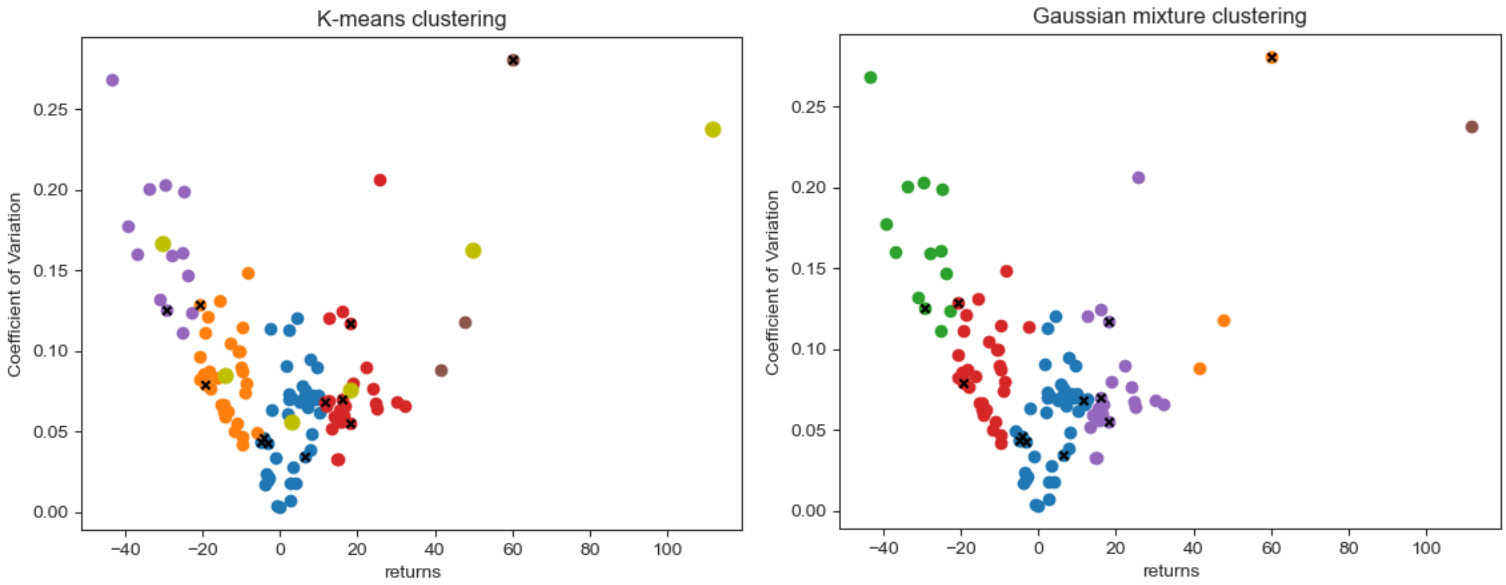
Gaussian Mixture Clustering is a probabilistic clustering algorithm that assumes the data points are generated from a mixture of Gaussian distributions with unknown parameters. The goal is to find the parameters of the Gaussian mixture model that best fit the observed data. The number of Gaussian distributions and their parameters (mean, variance, and mixing coefficients) are estimated using the Expectation-Maximization algorithm.

Once the model parameters have been estimated, the algorithm can assign each data point to the Gaussian distribution that has the highest probability of generating it. This results in a clustering of the data points into different groups, each associated with a different Gaussian distribution.

We will use both as K-means work well for linear relationships and gaussian mixture work well for non-linear relationships hence we will cover a lot of ground with those generic methods.

After those algorithms we will have for each ETFs and Macro factors a corresponding cluster. With this we can make links between them and tell which ETF is more related to US equity or Commodity or whatever. Later in this work we will also look at correlation coefficients to add more information in our classification.

Clustering the coefficient of variation against returns:



Yellow dots are the clusters centroids. Black cross are the Index points (macro factors) All the other dots are our ETFs.

So far, we can see that the K-means and the Gaussian mixture produced the same clusters for the Coefficient of variation against the returns.

Macro factors clusters:

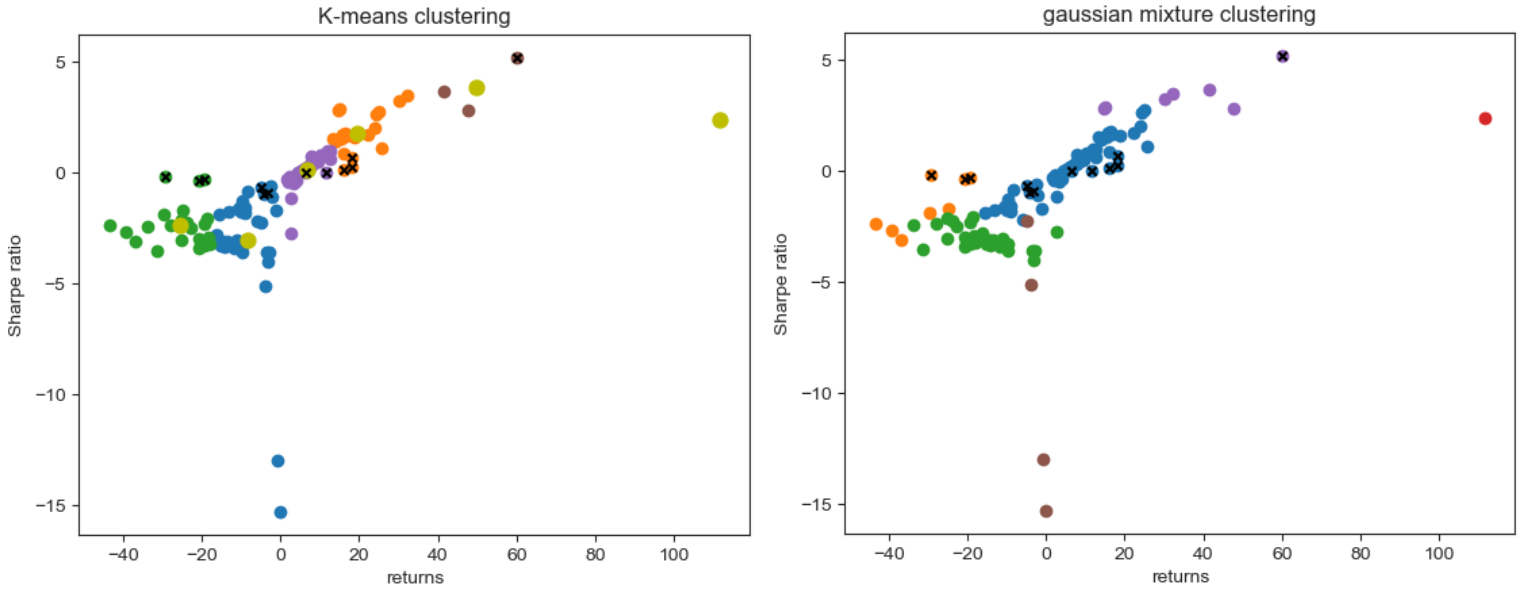
	mean	stdev	returns	coef var	risk reward	Sharpe ratio	cluster
US S&P 500	7668.516003	527.006350	11.519765	0.068723	167.625120	0.012903	0
EU Stoxx 600	1004.096690	55.506069	18.098873	0.055280	327.405970	0.241034	0
Japan Nikkei 225	47122.240755	1624.776070	6.403639	0.034480	185.720262	0.001036	3
MSCI EM USD	571.416528	73.426632	-20.709513	0.128499	-161.164386	-0.346325	1
US HY	313.719423	13.770712	-4.727549	0.043895	-107.701332	-0.686061	3
EU HY	203.568884	9.422265	-4.332767	0.046285	-93.609830	-0.960785	3
BCOM Energy	37.925627	10.645386	59.945693	0.280691	213.564643	5.187759	5
BCOM Industrial Metals	164.315915	19.197382	18.025767	0.116832	154.287725	0.693103	0
BCOM Gold	198.213625	8.542641	-3.285611	0.043098	-76.235532	-0.937135	3
Dollar Index	1152.923664	80.455146	15.940474	0.069784	228.427277	0.139462	0
US 10Y Bond	1059.905413	83.796698	-19.308782	0.079061	-244.227791	-0.286751	1
Germany Bund 10Y	1555.875280	194.707928	-29.150169	0.125144	-232.933645	-0.173954	4

*for information, the “risk reward” col is a proposition of metric similar to the Sharpe ratio. It is the returns over the coefficient of variation, meaning roughly the marginal unit of return gain over a unit of vol.

We see the limitation of the clustering methods selected here. We do not have a pure segmentation between Macro factors based on their underlying. For instance, the S&P500 and

EU Stoxx 600 are in the same cluster but not the Nikkei 225 which is also an equity index. In the same way, the BCOM Industrials Metals is composed of futures contracts and is in the same cluster as equity indexes. Lastly the US10Y Bond and Germany Bund 10Y aren't in the same cluster despite them being bonds index.

Clustering the Sharpe ratio against returns:



Yellow dots are the clusters centroids. Black cross are the Index points (macro factors) All the other dots are our ETFs.

In that case we can see that k-means and Gaussian mixture produce much different clusters. And for the matter of this work, we will use the gaussian produced cluster as we'll see below, they produce a segmentation more pure to the underlying of the macro factors.

Macro factors clusters:

	mean	stdev	returns	coef var	risk reward	Sharpe ratio	cluster_2
US S&P 500	7668.516003	527.006350	11.519765	0.068723	167.625120	0.012903	1
EU Stoxx 600	1004.096690	55.506069	18.098873	0.055280	327.405970	0.241034	1
Japan Nikkei 225	47122.240755	1624.776070	6.403639	0.034480	185.720262	0.001036	2
MSCI EM USD	571.416528	73.426632	-20.709513	0.128499	-161.164386	-0.346325	3
US HY	313.719423	13.770712	-4.727549	0.043895	-107.701332	-0.686061	1
EU HY	203.568884	9.422265	-4.332767	0.046285	-93.609830	-0.960785	2
BCOM Energy	37.925627	10.645386	59.945693	0.280691	213.564643	5.187759	1
BCOM Industrial Metals	164.315915	19.197382	18.025767	0.116832	154.287725	0.693103	1
BCOM Gold	198.213625	8.542641	-3.285611	0.043098	-76.235532	-0.937135	2
Dollar Index	1152.923664	80.455146	15.940474	0.069784	228.427277	0.139462	1
US 10Y Bond	1059.905413	83.796698	-19.308782	0.079061	-244.227791	-0.286751	3
Germany Bund 10Y	1555.875280	194.707928	-29.150169	0.125144	-232.933645	-0.173954	3

Here, the clustering of the Sharpe ratio against the returns under the gaussian method produced cluster that encompass both 10Y bonds index, furthermore the BCOM Industrials Metals and BCOM Energy are under the same cluster which is accurate I believe. Lastly, we do still have the S&P and EU Stoxx in the same cluster. Unfortunately, the Nikkei and high yield indexes still remain a challenge for such cluster exercise.

Furthermore, we can note that the K-means allocate more clusters to the index, we could count 5 over 6 clusters allocated between our macro factors whereas the gaussian method tends to spread the macro factors over less clusters, we have only 3 over 6 clusters allocated to the macro factors.

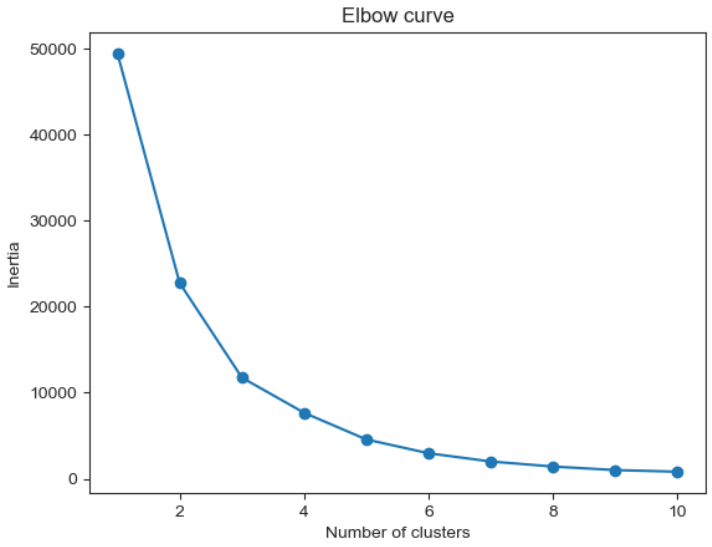
We can conclude that clustering with the Sharpe ratio produced a better depiction of the reality than the coefficient of variation which is the historical scaleless volatility as defined above. Hence, the Sharpe ratio is a better discriminant factor to identify patterns and asset class among our ETFs through a matching with the macro factors' clusters.

Determining the number of clusters:

Quickly, here's an explanation of why we selected 6 clusters.

This number is found iteratively by running a loss functions. We can see from below figure that beyond 6 clusters we do not reduce that much the "inertia". Which is a metric used to evaluate the quality of clustering in unsupervised learning algorithms such as k-means clustering. It is defined as the sum of squared distances of all data points to their assigned cluster center, where the cluster center is the mean of all data points in the cluster.

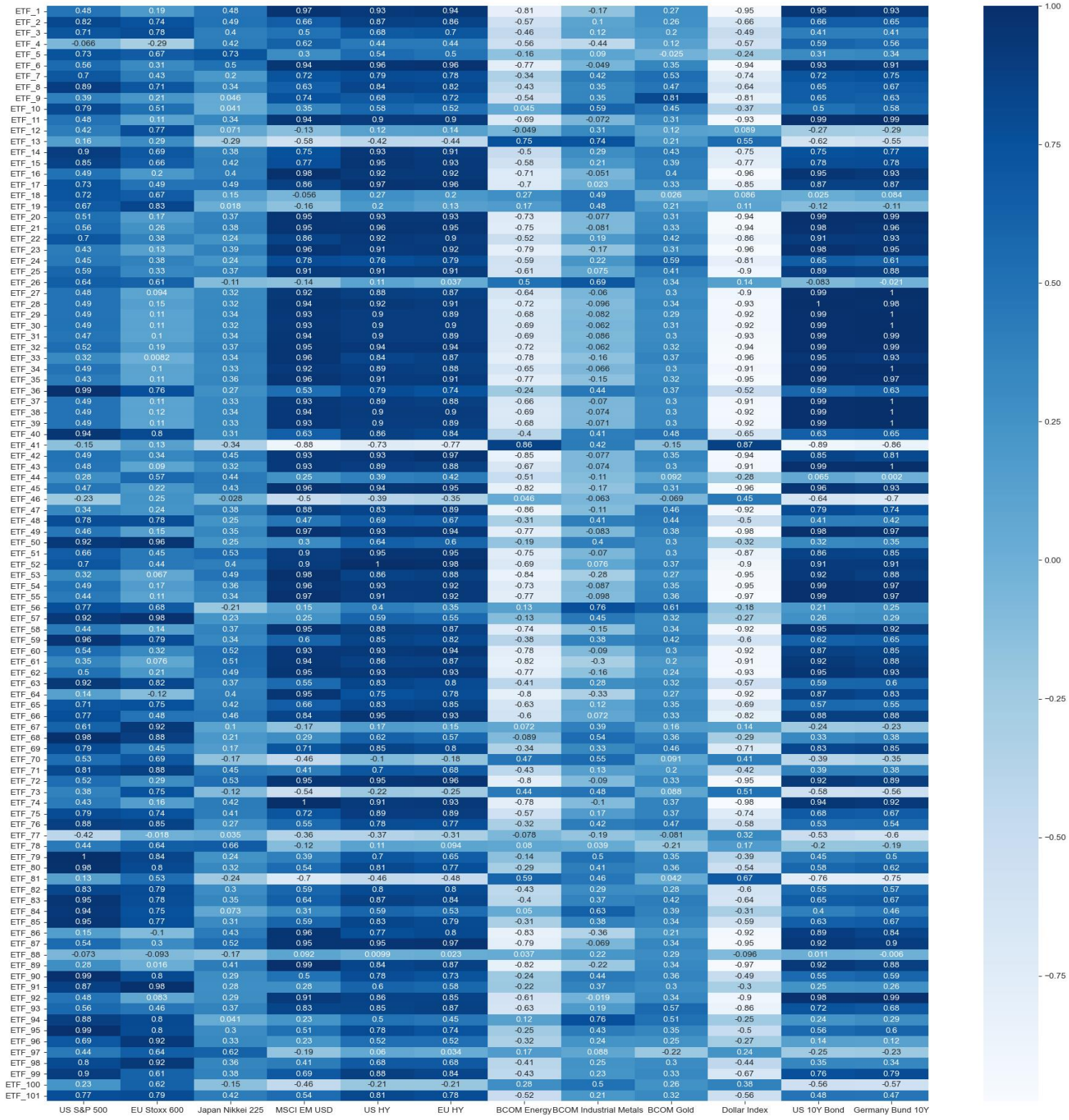
The inertia value measures how compact and well-separated the clusters are. A low inertia value indicates that the clusters are tightly packed around their centroids and well-separated from each other, which is desirable. On the other hand, a high inertia value indicates that the clusters are spread out and overlapping, which is not desirable.



Correlation coefficients between ETFs and Macro factors:

Beyond making clusters we can also look at the correlation between our ETFs and the different macro factors. We would have good reason to think that, for example, a highly correlated ETF with the S&P500 and EU Stoxx 600 is an equity ETF, not fixed income. In the same way, the sign of the coefficient bears a lot of information.

Using the Pearson correlation, we obtain the below matrix.



I note a few things:

- We have some ETFs that are perfectly correlated (1) with some factors. ETF79 is perfectly correlated with US S&P 500 for instance. The inverse is not true, I did not spot perfect negative correlation (-0.98 at maximum).
- Most of ETFs are negatively correlated with the Dollar index and the BCOM Energy. My hypothesis is that the Dollar index being a currency index, only a few of our ETFs tracks currencies related index. For the BCOM energy I don't have much guess.
- Most of the ETFs display a high positive correlation with the US high yield and EU high yield indexes as well as for the US 10Y bond and Germany bund 10Y.

Classification examples:

Easy case:

ETF39 :

- Correlation: perfect positive (1) with Germany 10Y bund
- Coefficient of variation K-means: Cluster 0 shared with US10Y bond, Germany 10Y outside of the cluster
- Sharpe ratio Gaussian: Cluster 2 shared with both US10Y and Germany 10Y

Hence this is a fixed income ETF.

Hard case:

That case is hard because this ETF is relatively low correlated to all macro factors and the clustering can display radically different asset classes.

ETF26 :

- Correlation: 0.6 at maximum with S&P and EU Stoxx
- Coefficient of variation K-means: Cluster 3 shared with Nikkei, US HY, EU HY10Y and BCOM gold
- Sharpe ratio Gaussian: Cluster 1 shared with Nikkei and BCOM gold

We can see equity, commodities futures and high yield. I could not state which is the more dominant but clearly this ETF is more risk oriented than the ETF39 case seen above.

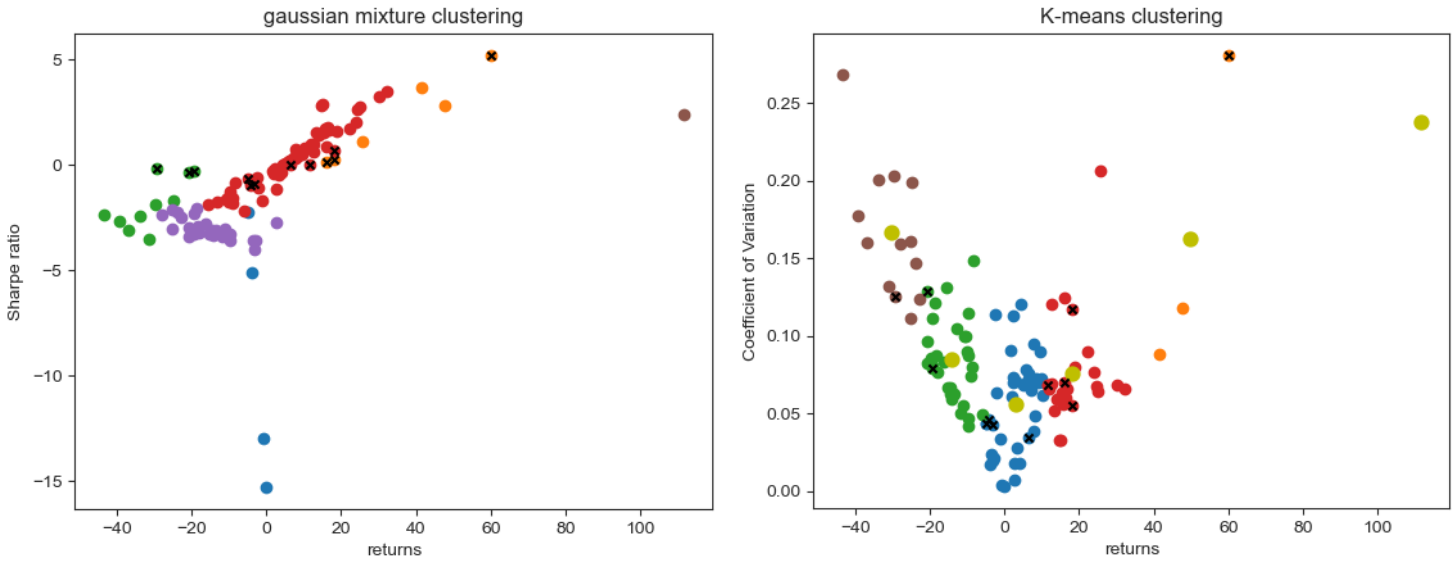
Hence even without clearly identifying the asset class, we can still classify this one based on its risk profile.

Bonus: ETF 41

At beginning we spotted one outlier on the ETFs price charts. Let's see his type.

- Correlation: 0.8 at maximum with Dollar index and BCOM gold, poorly correlated elsewhere
- Coefficient of variation K-means: Cluster 4 shared by no macro factors
- Sharpe ratio Gaussian: Cluster 3 shared shared by no macro factors

Looking back at the clustering map:



It appears that in this case our methods of clustering fail to include it inside of a cluster encompassing macro factors. The ETF41 is the far-most above right dot.

We can only rely on the correlation coefficient for this one unfortunately. As such we are either facing a currency-based ETF as it is positively correlated to 0.8 with the Dollar index, either a commodity based ETF as it is also positively correlated to 0.8 with the BCOM Energy. My guess is that maybe it is an inflation theme ETF tracking an index like the “ Markit iBoxx Euro Inflation Linked” for instance.

Which ETF for which investor?

Now that we have seen the returns and risk of these ETFs as well as a way of classifying them in relation with our macro factors, we can quickly indicate whose investors are suited for them.

I believe we can either think in term of risk/returns couple or asset class.

Traditionally, investors that are risk averse are directed toward fixed income funds, but in the negative rates environment of last years I don't think it is recommended at all. Looking at the Bonds macro factors' (US 10Y and Germany 10Y) returns we can see a negative returns over the period. In the same time we can see that with a smaller coefficient of var (our unitless historical volatility) the Nikkei produced a better return on investment. Hence for risk averse investors I would recommend an ETF among ours that is highly correlated and falls under the same cluster as the Nikkei.

Samely, the MSCI EM USD give a better return that the fixed income fund with same vol.

On the other side, investor that are moderated in their risk appetite, should opt for the equity ETFs that are classified like the US S&P 500 and the EU Stoxx 600.

Lastly investors that are risk neutral should definitely look for the high yield and commodities tracking funds. Hence ETFs that are classified samely as the US HY, EU HY and BCOM indexes.

	mean	stdev	returns	coef var	risk reward	Sharpe ratio	cluster
US S&P 500	7668.516003	527.006350	11.519765	0.068723	167.625120	0.012903	5
EU Stoxx 600	1004.096690	55.506069	18.098873	0.055280	327.405970	0.241034	5
Japan Nikkei 225	47122.240755	1624.776070	6.403639	0.034480	185.720262	0.001036	1
MSCI EM USD	571.416528	73.426632	-20.709513	0.128499	-161.164386	-0.346325	3
US HY	313.719423	13.770712	-4.727549	0.043895	-107.701332	-0.686061	1
EU HY	203.568884	9.422265	-4.332767	0.046285	-93.609830	-0.960785	1
BCOM Energy	37.925627	10.645386	59.945693	0.280691	213.564643	5.187759	2
BCOM Industrial Metals	164.315915	19.197382	18.025767	0.116832	154.287725	0.693103	5
BCOM Gold	198.213625	8.542641	-3.285611	0.043098	-76.235532	-0.937135	1
Dollar Index	1152.923664	80.455146	15.940474	0.069784	228.427277	0.139462	5
US 10Y Bond	1059.905413	83.796698	-19.308782	0.079061	-244.227791	-0.286751	3
Germany Bund 10Y	1555.875280	194.707928	-29.150169	0.125144	-232.933645	-0.173954	0

Mystery allocation:

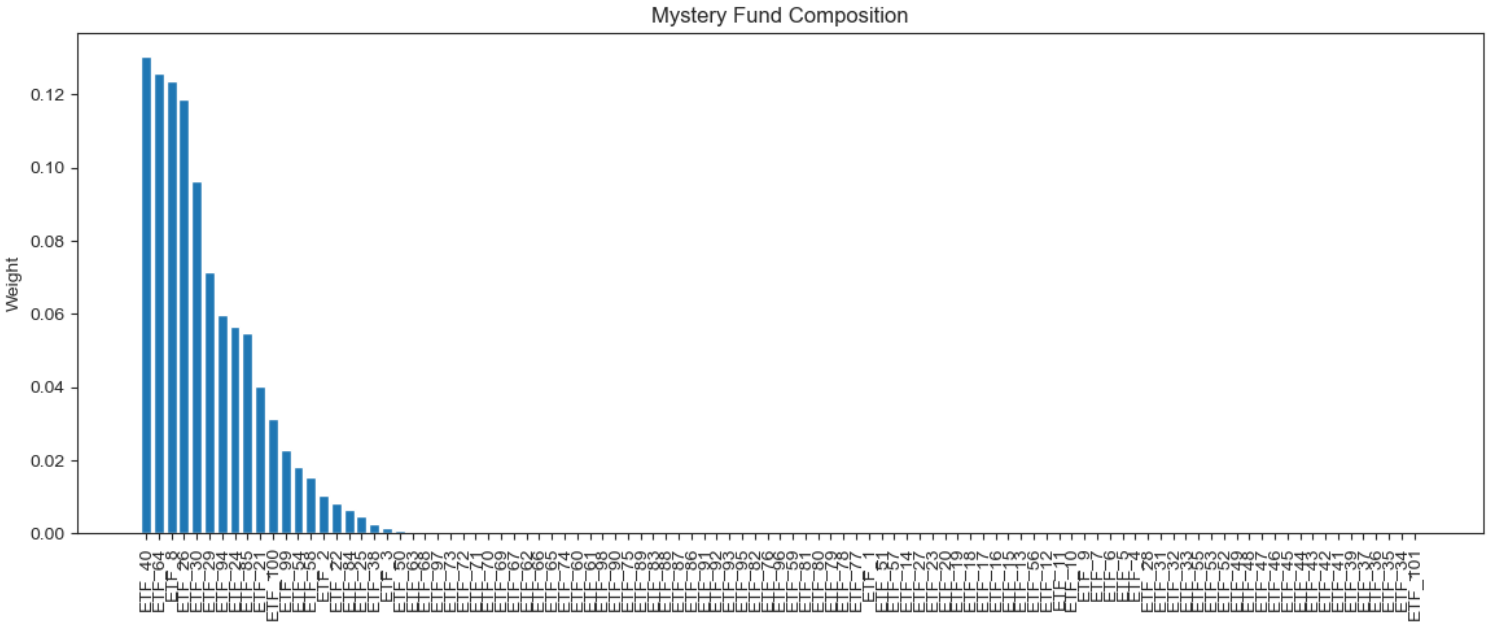
To determine the mystery allocation, one has to address a regression problem in essence. As we are given with the time series of the mystery allocation and the time series of the ETF, one has just to address which combination of Time series with correct weights produces the same time series as the mystery allocation.

In this regard, we will use the LASSO regression.

LASSO stands for Least Absolute Shrinkage and Selection Operator. It is a type of regularized regression method that adds a penalty term to the standard regression equation. The penalty term is proportional to the absolute value of the regression coefficients, which shrinks the coefficients towards zero. The amount of shrinkage is controlled by the hyperparameter λ. As λ increases, the magnitude of the coefficients decreases, which reduces the risk of overfitting the model. In the context of the mystery allocation problem, LASSO regression can be useful for feature selection. Since we have many ETFs, LASSO can automatically shrink the coefficients of the irrelevant ETFs to zero, which effectively removes them from the model. This simplifies the model and reduces the risk of overfitting.

LASSO regression also has the advantage of producing a sparse solution, where only a small subset of the coefficients are nonzero, furthermore we can add a non-negative coefficient constraint, which is nice in a weight problem. This makes the model more interpretable and easier to understand. Additionally, LASSO method encourages solutions with many zero coefficients. This property can be useful in the context of the mystery allocation problem, where we want to identify a small subset of ETFs that are most important for the mystery portfolio.

The regression produces the below results:



Here is the detailed weights:

Those are to be seen as %.

	Coefficient
ETF_38	0.002717
ETF_25	0.004510
ETF_84	0.006557
ETF_22	0.008337
ETF_2	0.010180
ETF_58	0.015410
ETF_54	0.018261
ETF_99	0.022695
ETF_100	0.031266
ETF_21	0.040272
ETF_85	0.054653
ETF_24	0.056615
ETF_94	0.059818
ETF_29	0.071535
ETF_30	0.096423
ETF_26	0.118569
ETF_8	0.123662
ETF_64	0.125976
ETF_40	0.130250

The sum of those is 0.997705, I did not include the rest because it is statistically non-significant.

Mystery allocation v2:

For the second mystery allocation which is a time varying allocation, rebalanced each day, I did not successfully implemented in python the Kalman filter method. Specifically, the transition matrices were troubleshooting, and I didn't figure it out in time. Hence, I don't have results to display here.

However, this is the way I would have done this puzzle. The Kalman filter is a technique that can help when trying to assess unknown variables, here the weights. It is relevant to current problem because the Kalman filter is updating its parameter continuously hence for a daily changing weights problem it is relevant.

I tried to follow the steps as written in the paper "Deep decoding of strategies" that you and others has published.

Please find the Python code I tried for the Kalman filter:

```
ETF_universe["Unnamed: 0"] = pd.to_datetime(ETF_universe["Unnamed: 0"])
Mystery_allocation_v2["Unnamed: 0"] = pd.to_datetime(Mystery_allocation["Unnamed: 0"])

combined_data_v2 = pd.merge(ETF_universe, Mystery_allocation_v2, on='Unnamed: 0')
combined_data2_v2 = combined_data_v2.drop('Unnamed: 0', axis = 1)
combined_data2_v2
from pykalman import KalmanFilter
X = combined_data2_v2.iloc[:, :-1]
y = combined_data2_v2.iloc[:, -1]
kf = KalmanFilter(n_dim_obs=100, n_dim_state=100)
initial_state_mean = allocation['Coefficient'].values
initial_state_covariance = np.diag(np.ones(100) * 0.01)
transition_matrix = np.eye(100)
observation_matrix = np.eye(100)
estimated_weights = []
state_mean = initial_state_mean
state_covariance = initial_state_covariance

for i in range(len(combined_data2_v2)):
    # Get the observed prices of the ETFs for this day
    observation = combined_data2_v2.iloc[i, :-1].values

    # Use the Kalman filter to estimate the weights of the mystery fund for this day
    state_mean, state_covariance = kf.filter_update(state_mean, state_covariance, observation,
transition_matrix, observation_matrix)

    # Store the estimated weights for this day in a DataFrame
    estimated_weights.append(state_mean)

estimated_weights = pd.DataFrame(estimated_weights, columns=allocation.index)
```

Sources and references:

T-Bills data:

https://home.treasury.gov/resource-center/data-chart-center/interest-rates/TextView?type=daily_treasury_yield_curve&field_tdr_date_value_month=202304

LASSO regression:

- The Elements of Statistical Learning, 2016.
- Applied Predictive Modeling, 2013.

Kalman filter:

- “Deep decoding of strategies” : Jean-Jacques Ohana, Eric Benhamou, David Saltiel, Beatrice Guez