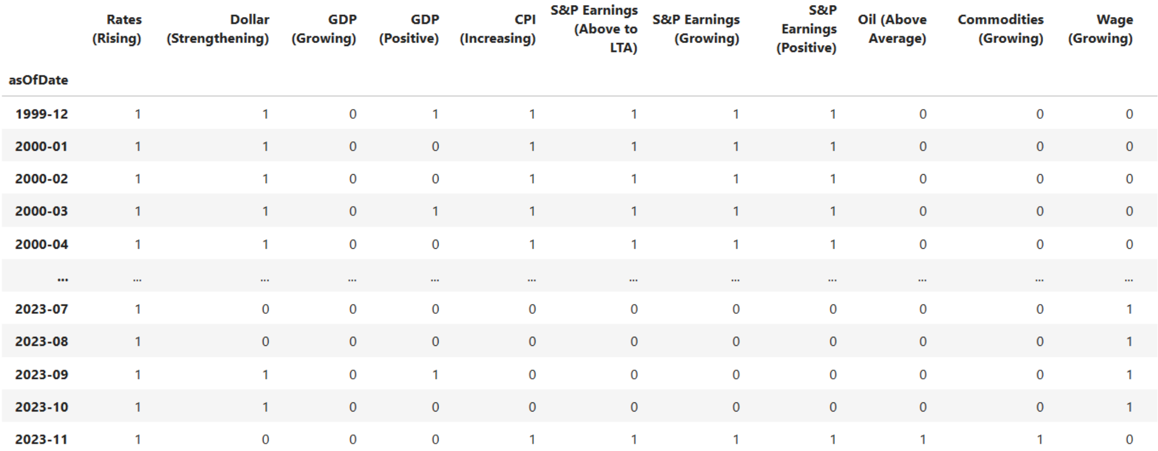
White Paper – Variable Replacement and Clustering

The firm's objective is to build CART models for ETFs. To make the models more intuitive, the variables used are Boolean transformations of macroeconomic variables. Below you can find some examples of the variables used for the CART models.



	Rates (Rising)	Dollar (Strengthening)	GDP (Growing)	GDP (Positive)	CPI (Increasing)	S&P Earnings (Above to LTA)	S&P Earnings (Growing)	S&P Earnings (Positive)	Oil (Above Average)	Commodities (Growing)	Wage (Growing)
asOfDate											
1999-12	1	1	0	1	1	1	1	1	0	0	0
2000-01	1	1	0	0	1	1	1	1	0	0	0
2000-02	1	1	0	0	1	1	1	1	0	0	0
2000-03	1	1	0	1	1	1	1	1	0	0	0
2000-04	1	1	0	0	1	1	1	1	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...
2023-07	1	0	0	0	0	0	0	0	0	0	1
2023-08	1	0	0	0	0	0	0	0	0	0	1
2023-09	1	1	0	1	0	0	0	0	0	0	1
2023-10	1	1	0	0	0	0	0	0	0	0	1
2023-11	1	0	0	0	1	1	1	1	1	1	0

Variable Replacement

The aim of this section is to identify suitable macroeconomic variables for substituting in the CART Models. To achieve this, we employed various procedures to analyze the macroeconomic variables, helping us pinpoint the most appropriate replacements.

Custom Hamming Distance

For this procedure, we conceptualize each macroeconomic variable as a point within an n-dimensional space, where n represents the total number of observations. Our objective is to calculate the distances between these variables in this multi-dimensional space. Given that our data is Boolean in nature, traditional Euclidean distances may not offer a meaningful representation. Consequently, we opt for a modified version of the Hamming distance as our chosen distance measure.

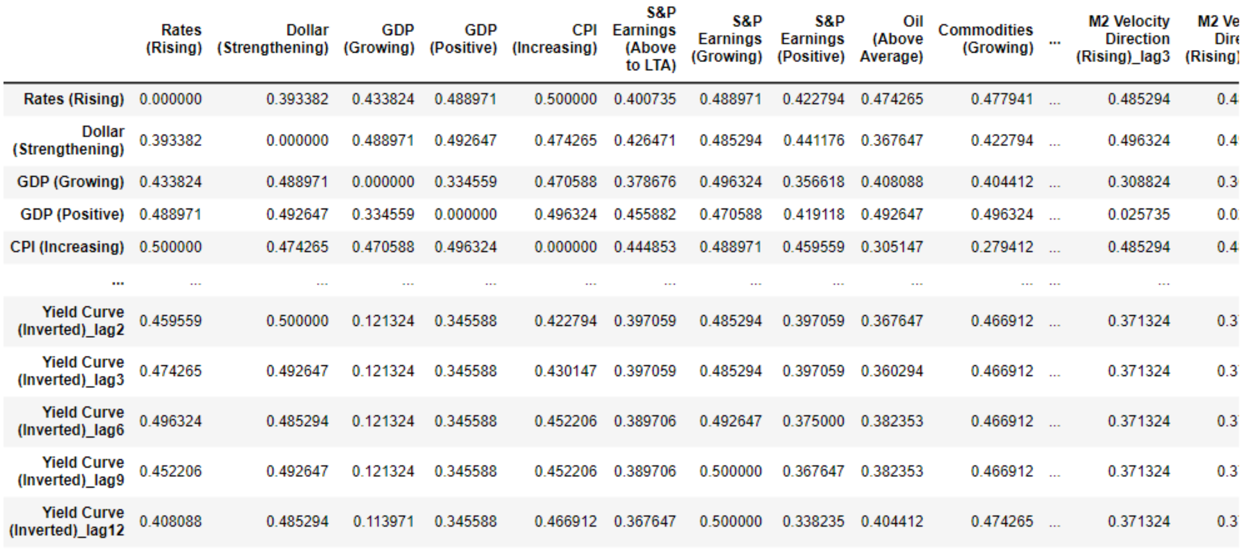
The Hamming distance quantifies the dissimilarity between two boolean series by counting the number of differing values. This distance metric yields values within the range of 0 to 1, with smaller values indicating a higher degree of commonality between the two series.

$$Hamming\ Distance = \frac{number\ of\ positions\ at\ which\ the\ corresponding\ values\ are\ different}{Total\ number\ of\ positions}$$

However, given the Boolean nature of our series, a variable positioned at a Hamming distance of 1 from another variable (where all elements differ) can be a strong candidate for a replacement variable. Consequently, we introduce a modified version of the Hamming distance that assigns greater significance to series with Hamming distances approaching 1.

$$Custom\ Hamming\ Distance = \min\left[\, Hamming\ Distance, \qquad 1 - Hamming\ Distance \,\right]$$

Below there is a portion of the distance matrix that was estimated using this measure.

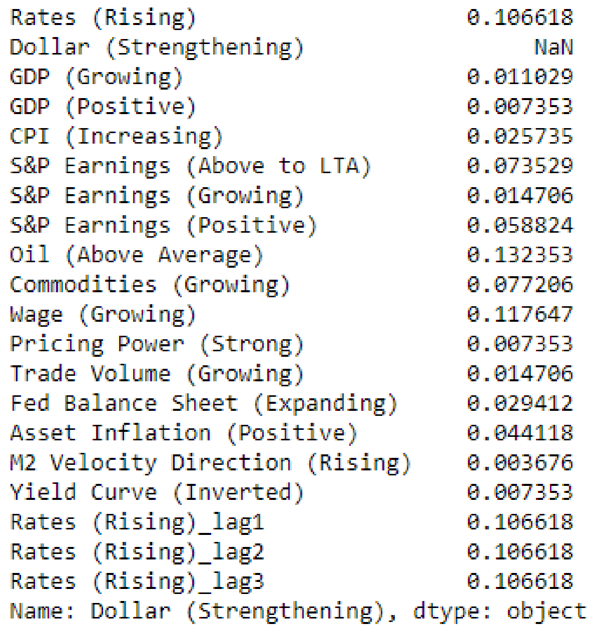


	Rates (Rising)	Dollar (Strengthening)	GDP (Growing)	GDP (Positive)	CPI (Increasing)	S&P Earnings (Above to LTA)	S&P Earnings (Growing)	S&P Earnings (Positive)	Oil (Above Average)	Commodities (Growing)	...	M2 Velocity Direction (Rising)_lag3	M2 Ve Dire (Rising]
Rates (Rising)	0.000000	0.393382	0.433824	0.488971	0.500000	0.400735	0.488971	0.422794	0.474265	0.477941	...	0.485294	0.4
Dollar (Strengthening)	0.393382	0.000000	0.488971	0.492647	0.474265	0.426471	0.485294	0.441176	0.367647	0.422794	...	0.496324	0.4
GDP (Growing)	0.433824	0.488971	0.000000	0.334559	0.470588	0.378676	0.496324	0.356618	0.408088	0.404412	...	0.308824	0.3
GDP (Positive)	0.488971	0.492647	0.334559	0.000000	0.496324	0.455882	0.470588	0.419118	0.492647	0.496324	...	0.025735	0.0
CPI (Increasing)	0.500000	0.474265	0.470588	0.496324	0.000000	0.444853	0.488971	0.459559	0.305147	0.279412	...	0.485294	0.4
...	...	...	...	...	...	...	...	...	...	...	...	...	
Yield Curve (Inverted)_lag2	0.459559	0.500000	0.121324	0.345588	0.422794	0.397059	0.485294	0.397059	0.367647	0.466912	...	0.371324	0.3
Yield Curve (Inverted)_lag3	0.474265	0.492647	0.121324	0.345588	0.430147	0.397059	0.485294	0.397059	0.360294	0.466912	...	0.371324	0.3
Yield Curve (Inverted)_lag6	0.496324	0.485294	0.121324	0.345588	0.452206	0.389706	0.492647	0.375000	0.382353	0.466912	...	0.371324	0.3
Yield Curve (Inverted)_lag9	0.452206	0.492647	0.121324	0.345588	0.452206	0.389706	0.500000	0.367647	0.382353	0.466912	...	0.371324	0.3
Yield Curve (Inverted)_lag12	0.408088	0.485294	0.113971	0.345588	0.466912	0.367647	0.500000	0.338235	0.404412	0.474265	...	0.371324	0.3

The raw scores generated by the tree models are higher when performance is better. However, this trend does not hold true for the Custom Hamming Distance, where lower values of this metric signify superior performance. To reconcile this and maintain consistency with the scoring scheme of previous models, we will apply a minor adjustment to our distance measure for estimating the raw score.

$$Hamming\ Raw\ Score = 0.5 - Custom\ Hamming\ Distance$$

Below you can see the raw scores that were estimated to determine the replacement of Dollar (Strengthening).



```
Rates (Rising)                     0.106618
Dollar (Strengthening)                  NaN
GDP (Growing)                      0.011029
GDP (Positive)                     0.007353
CPI (Increasing)                   0.025735
S&P Earnings (Above to LTA)        0.073529
S&P Earnings (Growing)             0.014706
S&P Earnings (Positive)            0.058824
Oil (Above Average)                0.132353
Commodities (Growing)              0.077206
Wage (Growing)                     0.117647
Pricing Power (Strong)             0.007353
Trade Volume (Growing)             0.014706
Fed Balance Sheet (Expanding)      0.029412
Asset Inflation (Positive)         0.044118
M2 Velocity Direction (Rising)     0.003676
Yield Curve (Inverted)             0.007353
Rates (Rising)_lag1                0.106618
Rates (Rising)_lag2                0.106618
Rates (Rising)_lag3                0.106618
Name: Dollar (Strengthening), dtype: object
```

.

Custom Concordance Rate

The Concordance rate also quantifies the similarity between two boolean series by counting the number of agreements and subtracting the number of disagreements. This distance metric yields values within the range of 0 to 1, with bigger values indicating a higher degree of commonality between the two series.

$$Concordance\ Rate = \frac{number\ of\ agreements - number\ of\ disagreements}{Total\ number\ of\ observations}$$

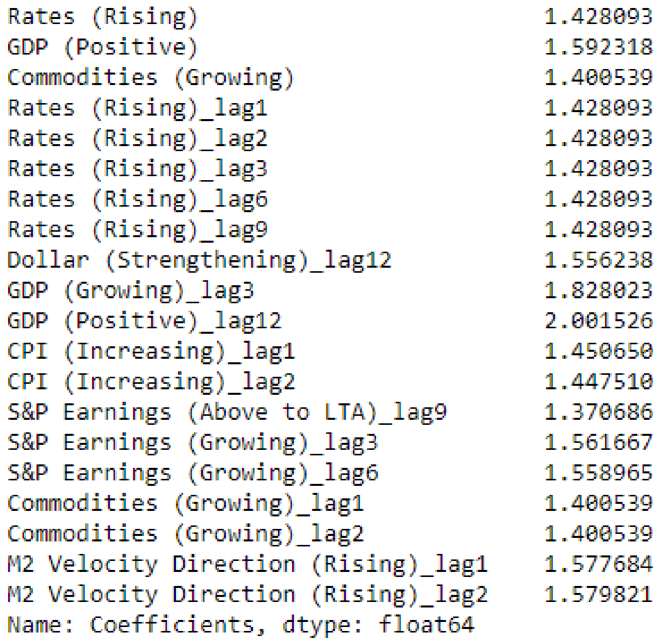
For this procedure we also utilize a modified version of the Concordance Rate that assigns greater significance to series with Hamming distances approaching 1.

$$Custom\ Concordance\ Rate = \min\left[\ Concordance\ Rate, \qquad 1 - Concordance\ Rate\ \right]$$

Logistic Regression

In this process, we will construct a logistic regression model with 'l1' penalty for each of the macroeconomic variables. Subsequently, we will extract the variables chosen by each individual model and perform separate logistic regressions between these selected variables and the target variable. The raw scores for this procedure will be determined as the absolute values of the coefficients derived from these individual regressions.

Below you can see the raw scores estimated using the the GDP (Growing) model.



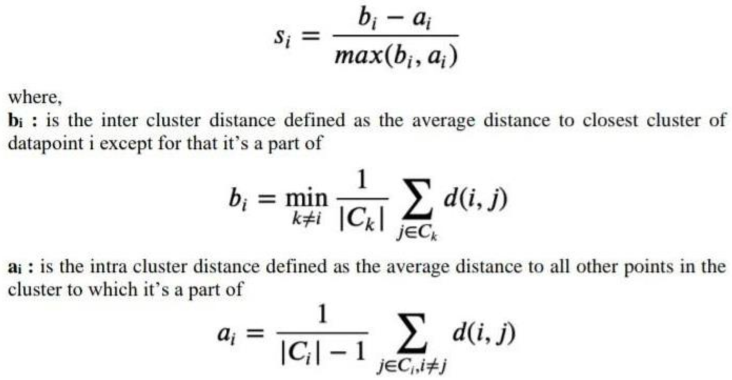
```
Rates (Rising)                           1.428093
GDP (Positive)                           1.592318
Commodities (Growing)                    1.400539
Rates (Rising)_lag1                      1.428093
Rates (Rising)_lag2                      1.428093
Rates (Rising)_lag3                      1.428093
Rates (Rising)_lag6                      1.428093
Rates (Rising)_lag9                      1.428093
Dollar (Strengthening)_lag12             1.556238
GDP (Growing)_lag3                       1.828023
GDP (Positive)_lag12                     2.001526
CPI (Increasing)_lag1                    1.450650
CPI (Increasing)_lag2                    1.447510
S&P Earnings (Above to LTA)_lag9         1.370686
S&P Earnings (Growing)_lag3              1.561667
S&P Earnings (Growing)_lag6              1.558965
Commodities (Growing)_lag1               1.400539
Commodities (Growing)_lag2               1.400539
M2 Velocity Direction (Rising)_lag1      1.577684
M2 Velocity Direction (Rising)_lag2      1.579821
Name: Coefficients, dtype: float64
```

Clustering

We execute the Agglomerative Clustering and DBscan procedures using the distance matrix that incorporates the Modified Hamming measure. We obtain two different sets of clusters and identify the common elements between the two clusters. We arrange the common elements based on their proximity to the target variable using the Modified Hamming measure. The score will be the Hamming raw score.

CLUSTERING PROCEDURES

In our dataset of Boolean transformations we implemented various clustering procedures. To assess the quality of the obtained cluster sets, we use the silhouette coefficient. This coefficient measures the cohesion within the same cluster and the separation from other clusters' data. It is calculated using the following formula:



$$s_i = \frac{b_i - a_i}{max(b_i, a_i)}$$

where,
b$_i$: is the inter cluster distance defined as the average distance to closest cluster of datapoint i except for that it’s a part of

$$b_i = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i, j)$$

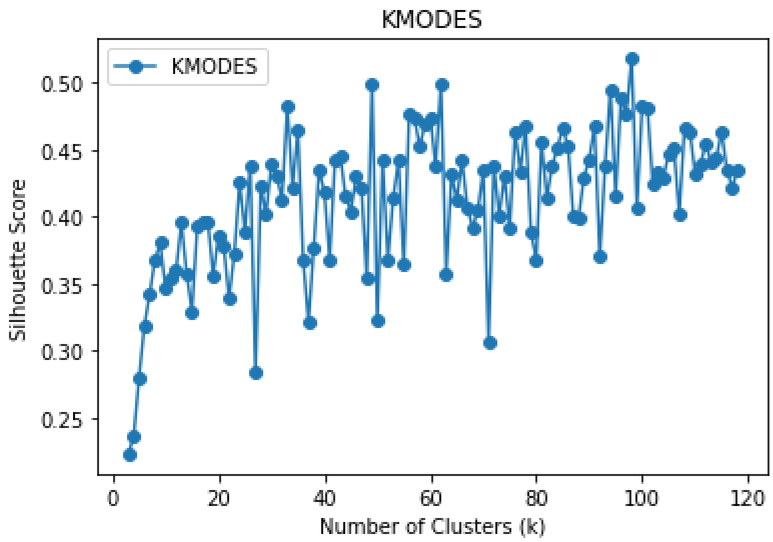
a$_i$: is the intra cluster distance defined as the average distance to all other points in the cluster to which it’s a part of

$$a_i = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i, j)$$

The distance measure used in the calculation of the silhouette coefficient is the Custom Hamming Distance. The silhouette coefficient ranges from -1 to 1, where values close to 1 indicate better clustering quality, and values of 0 or less than 0 indicate poor grouping.

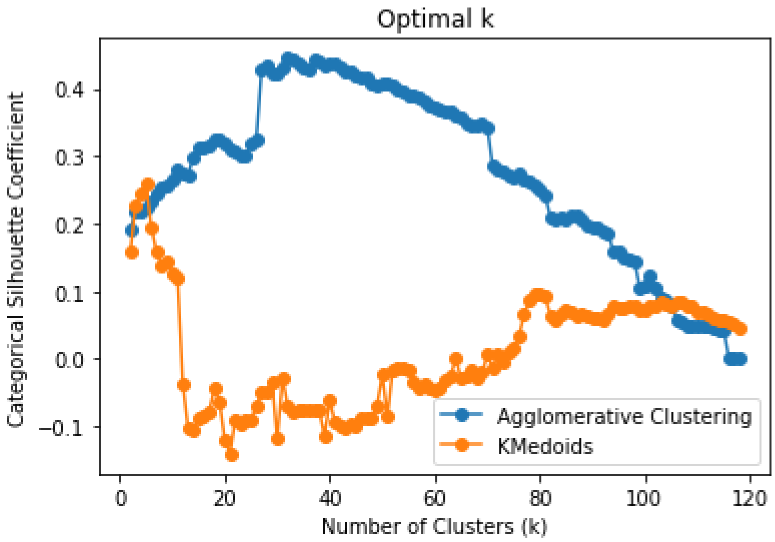
KMODES

We executed the KModes algorithm from KModes on our dataset. Similar to KMeans, this procedure requires specifying the number of clusters as an input parameter. Additionally, it's necessary to specify the distance measure the algorithm will use. In this algorithm, the Custom Hamming Distance is not available as distance measure, therefore we opted for Hamming distance instead. However, to capture the inverse relationship between the variables we also included in the procedure their inverted series. We implemented the procedure with varying values of K and estimated the silhouette score for each cluster set obtained. The results obtained following this methodology are depicted in the lower graph.



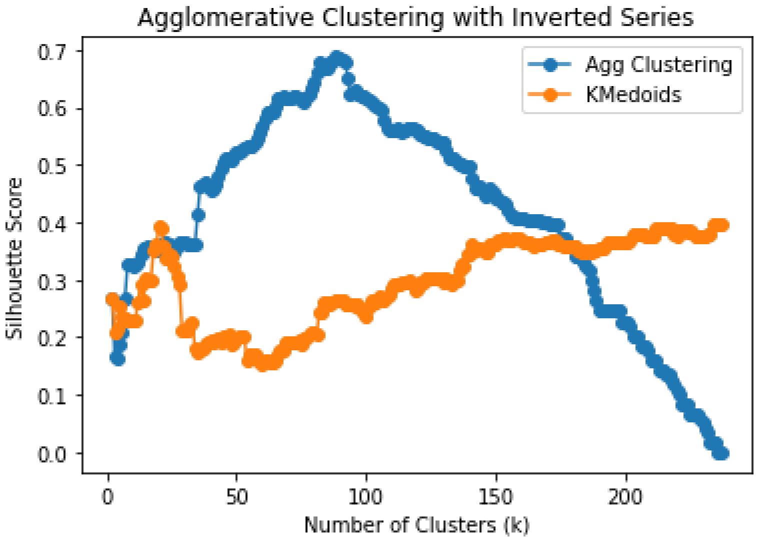
AGGLOMERATIVE CLUSTERING

To implement the Agglomerative Clustering procedure from Scikit-learn, the required input parameters are the distance matrix and the number of clusters. The distance matrix corresponds to the one obtained using the Custom Hamming Distance measure. We ran this procedure with varying values of K and estimated the silhouette coefficient for each cluster set obtained. The lower graph illustrates the results obtained using this and KMedoids methodologies.



From the graph, it's evident that the cluster set achieved with K=39 obtains the highest silhouette score.

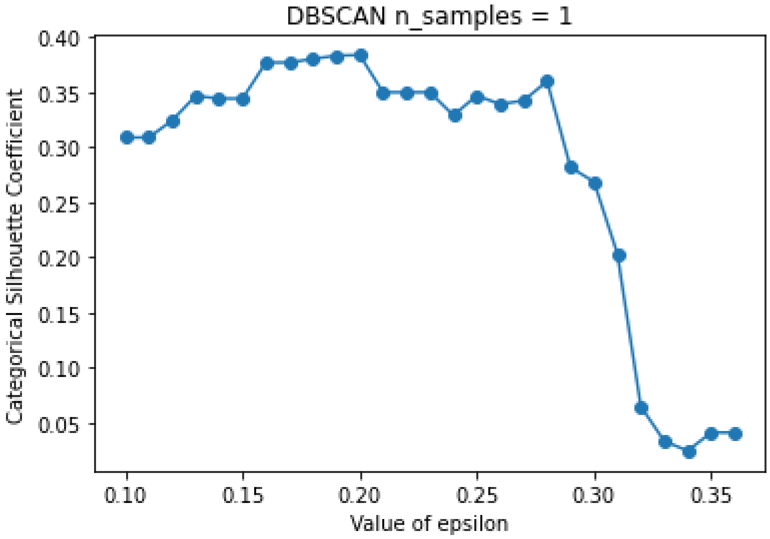
In order to compare the results with KModes we also implemented the procedure including the inverted series. You can see the results in the lower graph. It is evident from the graph that the cluster sets obtained following Agglomerative Clustering procedure have better quality.

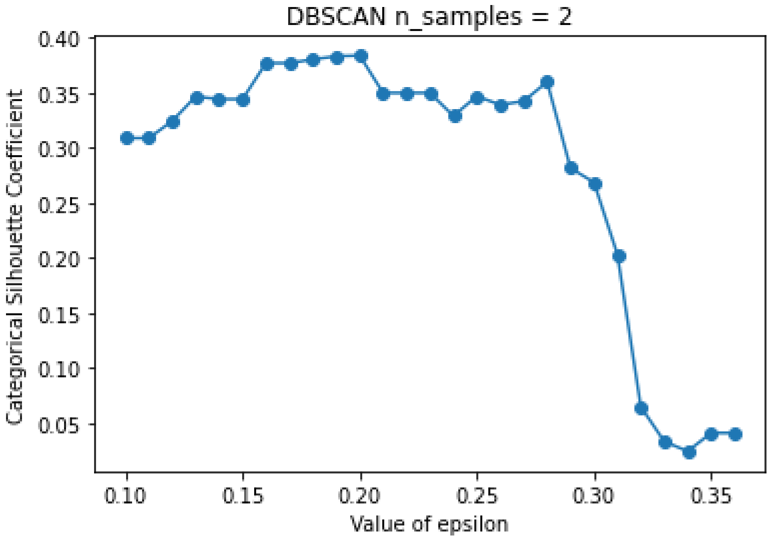


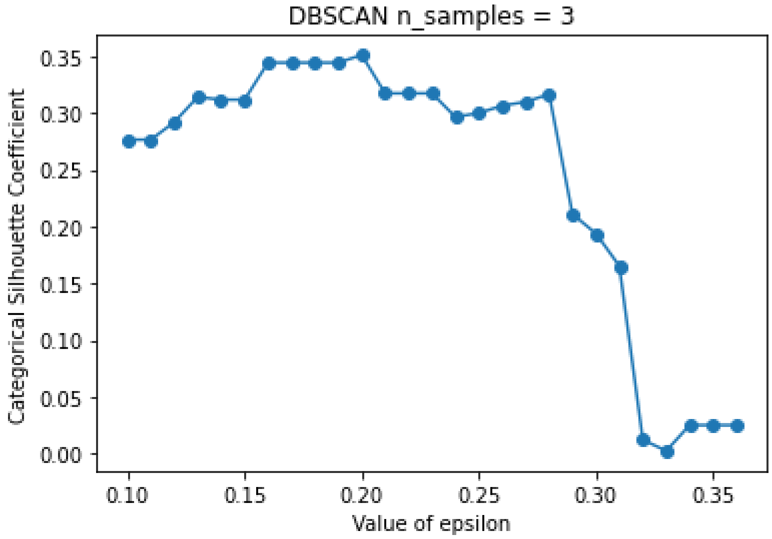
DBSCAN

The implementation of the DBSCAN procedure from scikit learn requires three input parameters: the distance matrix, epsilon (ε), and the minimum number of points required to consider a point as a nucleus.

We executed the procedure using varying values of epsilon (ε) and three different values (1,2 and 3) for n, which represent the minimum number of points around a point to be considered a core point. For each cluster set obtained, we estimated its silhouette score. Below, you can observe the results obtained.



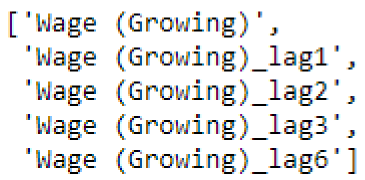




From the graphs, it becomes evident that the optimal parameter combination is epsilon=0.19 and n=2.

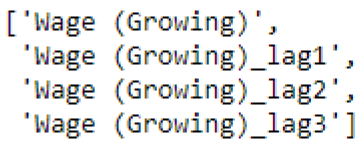
AGGREGATE RESULTS

We found the best clusters using both DBSCAN and Agglomerative Clustering methods. We will then match the variables grouped with each main variable. Our final results will include what both methods agree upon.



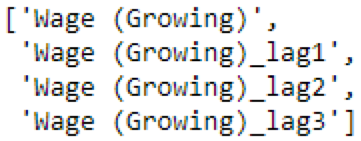
```
['Wage (Growing)',
 'Wage (Growing)_lag1',
 'Wage (Growing)_lag2',
 'Wage (Growing)_lag3',
 'Wage (Growing)_lag6']
```

Wage (Growing) cluster determined by Agglomerative Clustering.



```
['Wage (Growing)',
 'Wage (Growing)_lag1',
 'Wage (Growing)_lag2',
 'Wage (Growing)_lag3']
```

Wage (Growing) cluster determined by DBSCAN.



```
['Wage (Growing)',
 'Wage (Growing)_lag1',
 'Wage (Growing)_lag2',
 'Wage (Growing)_lag3']
```

Wage (Growing) cluster aggregating results.

COMPARATIVE ANALYSIS BETWEEN CLUSTER SETS

The purpose of this section is to determine whether the clusters obtained using the previous methodology remain consistent across different time periods. To achieve this, we implement clustering methodologies in different time spans and compare the resulting sets of clusters in consecutive periods. Our analysis focuses on three main aspects. Firstly, we delineate various time periods using an expanding window that extends by one year between consecutive time periods. Secondly, we employ a rolling window of 15-year duration to identify different time periods, with a one-year difference between successive windows. The final analysis involves dividing the entire study period into three equal parts.

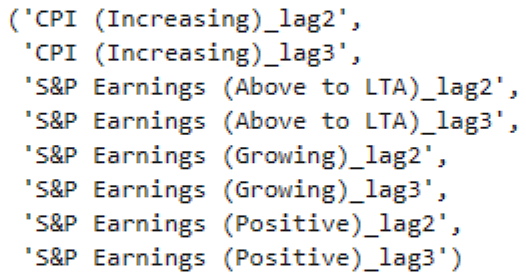
This comparative analysis requires determining the corresponding pairs of clusters between the two sets being compared. To find these corresponding cluster pairs, we follow the following methodology: first, we estimate the similarity matrix between the two sets of clusters using the Jaccard measure.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

We will identify the two clusters for which this measure is the largest among all clusters and set these clusters as the first pair of corresponding clusters. Next, we remove from the matrix all values from the rows and columns associated with these clusters. With the new matrix, we again identify the maximum similarity value. We set this cluster pair as the second pair of corresponding clusters and remove from the similarity matrix all values from the rows and columns associated with these clusters. We repeat this process to obtain all the pairs of corresponding clusters, continuing until the similarity matrix has no rows or no columns left. Clusters that do not have any correspondences at this point will be assigned an empty set as correspondence.

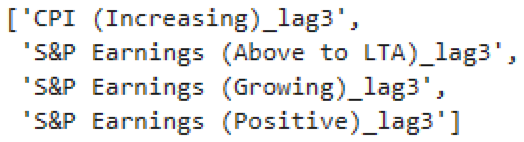
Once we obtain the correspondences between the sets of clusters, we compare the corresponding clusters. Our analysis will examine the elements that compose the clusters and their centroids. We extract the dissonant elements between the corresponding clusters and store them in a separate list. The dissimilarity between the two sets of clusters will be defined as the percentage of mismatching elements relative to the total number of elements.

$$dissimilarity\ between\ clusters = \frac{\#\ missmatching\ elements}{\#\ total\ elements}$$



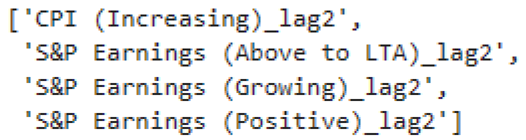
```
('CPI (Increasing)_lag2',
 'CPI (Increasing)_lag3',
 'S&P Earnings (Above to LTA)_lag2',
 'S&P Earnings (Above to LTA)_lag3',
 'S&P Earnings (Growing)_lag2',
 'S&P Earnings (Growing)_lag3',
 'S&P Earnings (Positive)_lag2',
 'S&P Earnings (Positive)_lag3')
```

Cluster from Cluster set 2007-2012



```
['CPI (Increasing)_lag3',
 'S&P Earnings (Above to LTA)_lag3',
 'S&P Earnings (Growing)_lag3',
 'S&P Earnings (Positive)_lag3']
```

Corresponding cluster from Cluster set 2008-2013



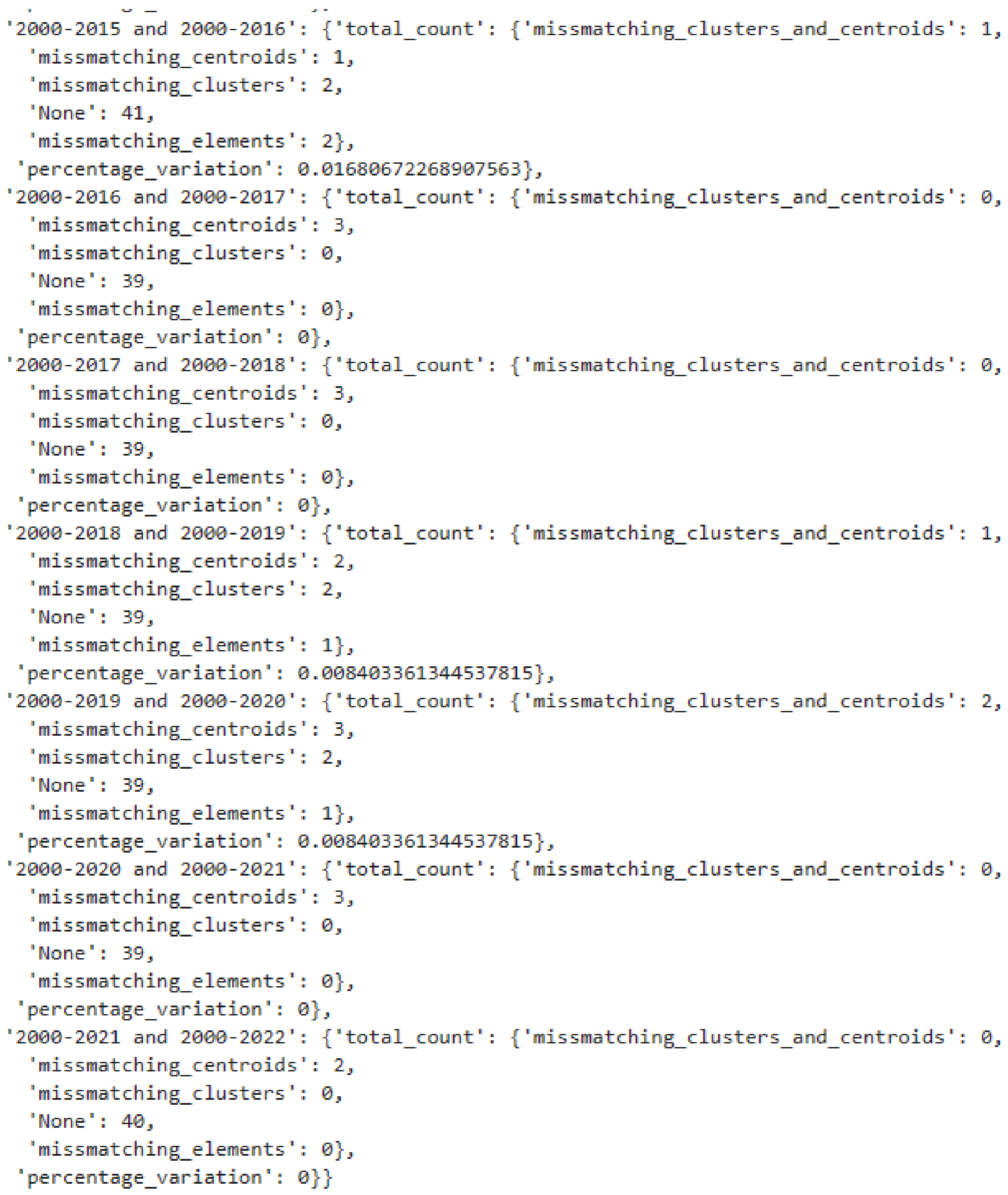
```
['CPI (Increasing)_lag2',
 'S&P Earnings (Above to LTA)_lag2',
 'S&P Earnings (Growing)_lag2',
 'S&P Earnings (Positive)_lag2']
```

Dissonant elements between cluster from 2007-2012 and its corresponding cluster from 2008-2013.

To broaden our analysis, we also compare the centroids between the two sets of clusters and determine the number of mismatching centroids between the two sets of clusters. We also determine the number of mismatching clusters and the number of clusters in which there is dissonance in both their components and their centroid.

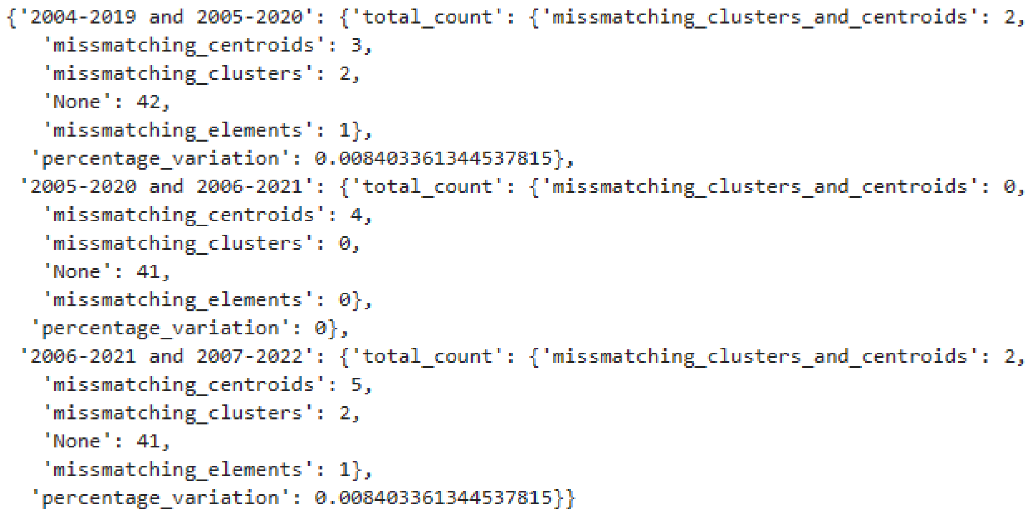
At the bottom section, the summarized results of our analysis are presented.

Expanding Window



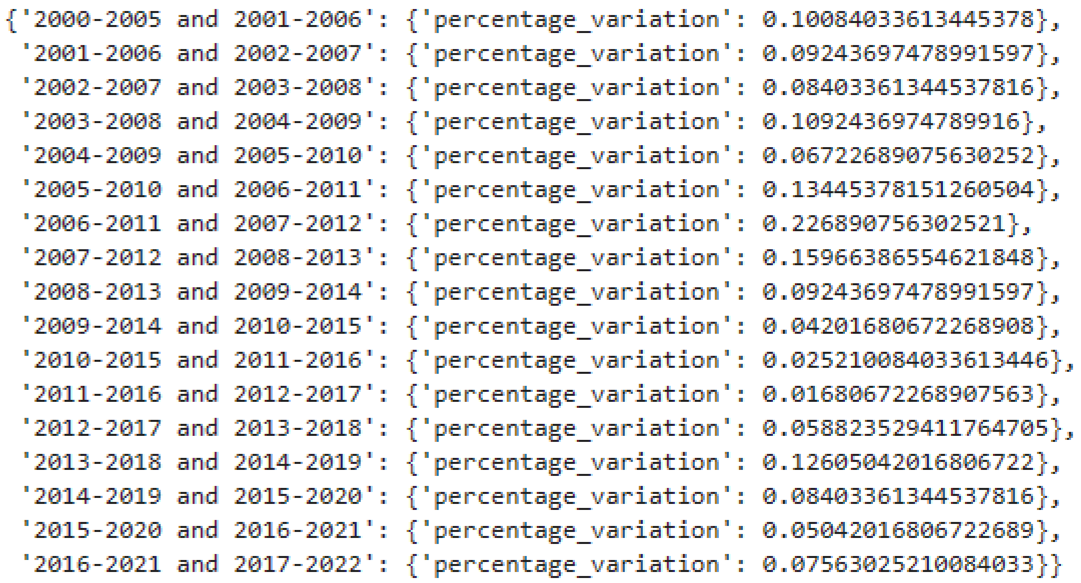
```
'2000-2015 and 2000-2016': {'total_count': {'missmatching_clusters_and_centroids': 1,
   'missmatching_centroids': 1,
   'missmatching_clusters': 2,
   'None': 41,
   'missmatching_elements': 2},
  'percentage_variation': 0.01680672268907563},
 '2000-2016 and 2000-2017': {'total_count': {'missmatching_clusters_and_centroids': 0,
   'missmatching_centroids': 3,
   'missmatching_clusters': 0,
   'None': 39,
   'missmatching_elements': 0},
  'percentage_variation': 0},
 '2000-2017 and 2000-2018': {'total_count': {'missmatching_clusters_and_centroids': 0,
   'missmatching_centroids': 3,
   'missmatching_clusters': 0,
   'None': 39,
   'missmatching_elements': 0},
  'percentage_variation': 0},
 '2000-2018 and 2000-2019': {'total_count': {'missmatching_clusters_and_centroids': 1,
   'missmatching_centroids': 2,
   'missmatching_clusters': 2,
   'None': 39,
   'missmatching_elements': 1},
  'percentage_variation': 0.008403361344537815},
 '2000-2019 and 2000-2020': {'total_count': {'missmatching_clusters_and_centroids': 2,
   'missmatching_centroids': 3,
   'missmatching_clusters': 2,
   'None': 39,
   'missmatching_elements': 1},
  'percentage_variation': 0.008403361344537815},
 '2000-2020 and 2000-2021': {'total_count': {'missmatching_clusters_and_centroids': 0,
   'missmatching_centroids': 3,
   'missmatching_clusters': 0,
   'None': 39,
   'missmatching_elements': 0},
  'percentage_variation': 0},
 '2000-2021 and 2000-2022': {'total_count': {'missmatching_clusters_and_centroids': 0,
   'missmatching_centroids': 2,
   'missmatching_clusters': 0,
   'None': 40,
   'missmatching_elements': 0},
  'percentage_variation': 0}}
```

Rolling Window (15 years)



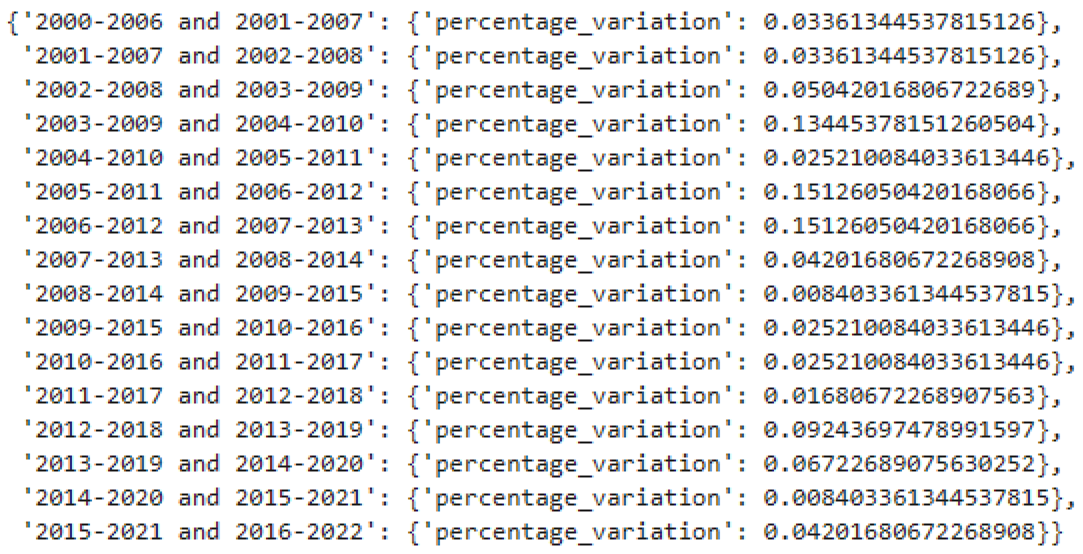
```
{'2004-2019 and 2005-2020': {'total_count': {'missmatching_clusters_and_centroids': 2,
   'missmatching_centroids': 3,
   'missmatching_clusters': 2,
   'None': 42,
   'missmatching_elements': 1},
  'percentage_variation': 0.008403361344537815},
 '2005-2020 and 2006-2021': {'total_count': {'missmatching_clusters_and_centroids': 0,
   'missmatching_centroids': 4,
   'missmatching_clusters': 0,
   'None': 41,
   'missmatching_elements': 0},
  'percentage_variation': 0},
 '2006-2021 and 2007-2022': {'total_count': {'missmatching_clusters_and_centroids': 2,
   'missmatching_centroids': 5,
   'missmatching_clusters': 2,
   'None': 41,
   'missmatching_elements': 1},
  'percentage_variation': 0.008403361344537815}}
```

Rolling Window (5 years)



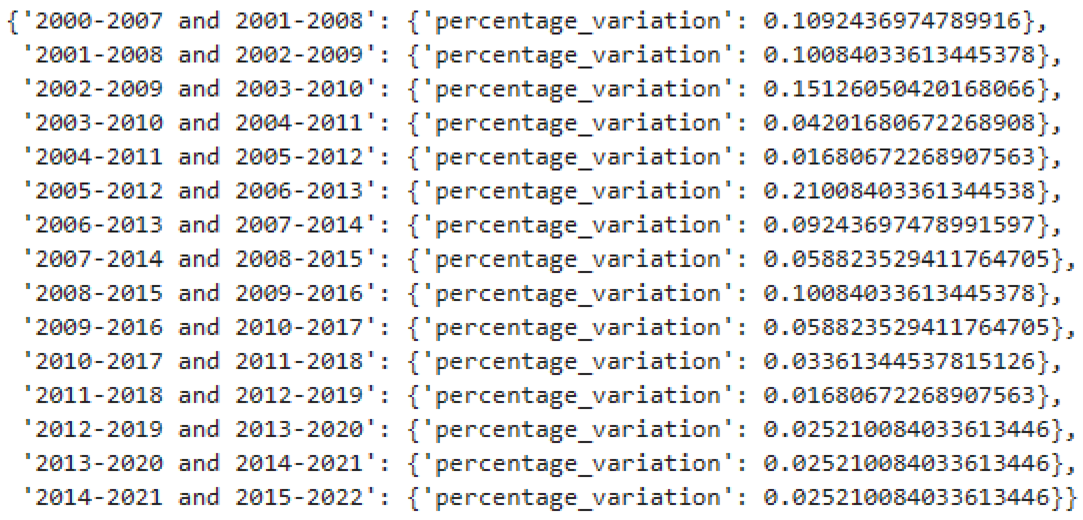
```
{'2000-2005 and 2001-2006': {'percentage_variation': 0.10084033613445378},
 '2001-2006 and 2002-2007': {'percentage_variation': 0.09243697478991597},
 '2002-2007 and 2003-2008': {'percentage_variation': 0.08403361344537816},
 '2003-2008 and 2004-2009': {'percentage_variation': 0.1092436974789916},
 '2004-2009 and 2005-2010': {'percentage_variation': 0.06722689075630252},
 '2005-2010 and 2006-2011': {'percentage_variation': 0.13445378151260504},
 '2006-2011 and 2007-2012': {'percentage_variation': 0.226890756302521},
 '2007-2012 and 2008-2013': {'percentage_variation': 0.15966386554621848},
 '2008-2013 and 2009-2014': {'percentage_variation': 0.09243697478991597},
 '2009-2014 and 2010-2015': {'percentage_variation': 0.04201680672268908},
 '2010-2015 and 2011-2016': {'percentage_variation': 0.025210084033613446},
 '2011-2016 and 2012-2017': {'percentage_variation': 0.01680672268907563},
 '2012-2017 and 2013-2018': {'percentage_variation': 0.058823529411764705},
 '2013-2018 and 2014-2019': {'percentage_variation': 0.12605042016806722},
 '2014-2019 and 2015-2020': {'percentage_variation': 0.08403361344537816},
 '2015-2020 and 2016-2021': {'percentage_variation': 0.05042016806722689},
 '2016-2021 and 2017-2022': {'percentage_variation': 0.07563025210084033}}
```

Rolling Window (6 years)



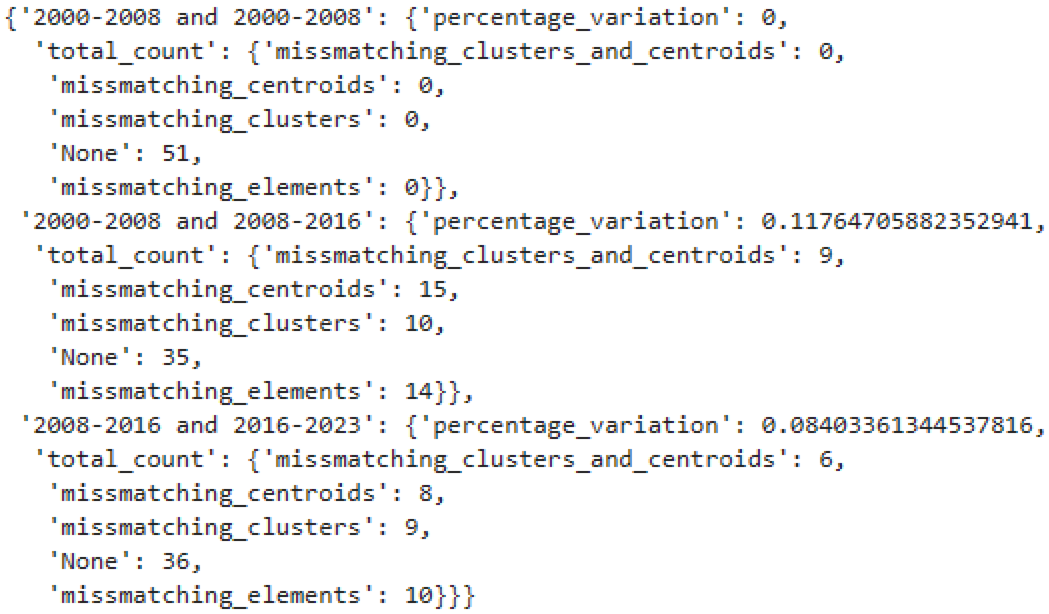
```
{'2000-2006 and 2001-2007': {'percentage_variation': 0.03361344537815126},
 '2001-2007 and 2002-2008': {'percentage_variation': 0.03361344537815126},
 '2002-2008 and 2003-2009': {'percentage_variation': 0.05042016806722689},
 '2003-2009 and 2004-2010': {'percentage_variation': 0.13445378151260504},
 '2004-2010 and 2005-2011': {'percentage_variation': 0.025210084033613446},
 '2005-2011 and 2006-2012': {'percentage_variation': 0.15126050420168066},
 '2006-2012 and 2007-2013': {'percentage_variation': 0.15126050420168066},
 '2007-2013 and 2008-2014': {'percentage_variation': 0.04201680672268908},
 '2008-2014 and 2009-2015': {'percentage_variation': 0.008403361344537815},
 '2009-2015 and 2010-2016': {'percentage_variation': 0.025210084033613446},
 '2010-2016 and 2011-2017': {'percentage_variation': 0.025210084033613446},
 '2011-2017 and 2012-2018': {'percentage_variation': 0.01680672268907563},
 '2012-2018 and 2013-2019': {'percentage_variation': 0.09243697478991597},
 '2013-2019 and 2014-2020': {'percentage_variation': 0.06722689075630252},
 '2014-2020 and 2015-2021': {'percentage_variation': 0.008403361344537815},
 '2015-2021 and 2016-2022': {'percentage_variation': 0.04201680672268908}}
```

Rolling Window (7 years)



```
{'2000-2007 and 2001-2008': {'percentage_variation': 0.1092436974789916},
 '2001-2008 and 2002-2009': {'percentage_variation': 0.10084033613445378},
 '2002-2009 and 2003-2010': {'percentage_variation': 0.15126050420168066},
 '2003-2010 and 2004-2011': {'percentage_variation': 0.04201680672268908},
 '2004-2011 and 2005-2012': {'percentage_variation': 0.01680672268907563},
 '2005-2012 and 2006-2013': {'percentage_variation': 0.21008403361344538},
 '2006-2013 and 2007-2014': {'percentage_variation': 0.09243697478991597},
 '2007-2014 and 2008-2015': {'percentage_variation': 0.058823529411764705},
 '2008-2015 and 2009-2016': {'percentage_variation': 0.10084033613445378},
 '2009-2016 and 2010-2017': {'percentage_variation': 0.058823529411764705},
 '2010-2017 and 2011-2018': {'percentage_variation': 0.03361344537815126},
 '2011-2018 and 2012-2019': {'percentage_variation': 0.01680672268907563},
 '2012-2019 and 2013-2020': {'percentage_variation': 0.025210084033613446},
 '2013-2020 and 2014-2021': {'percentage_variation': 0.025210084033613446},
 '2014-2021 and 2015-2022': {'percentage_variation': 0.025210084033613446}}
```

Fixed Window (3 periods)



```
{'2000-2008 and 2000-2008': {'percentage_variation': 0,
  'total_count': {'missmatching_clusters_and_centroids': 0,
   'missmatching_centroids': 0,
   'missmatching_clusters': 0,
   'None': 51,
   'missmatching_elements': 0}},
 '2000-2008 and 2008-2016': {'percentage_variation': 0.11764705882352941,
  'total_count': {'missmatching_clusters_and_centroids': 9,
   'missmatching_centroids': 15,
   'missmatching_clusters': 10,
   'None': 35,
   'missmatching_elements': 14}},
 '2008-2016 and 2016-2023': {'percentage_variation': 0.08403361344537816,
  'total_count': {'missmatching_clusters_and_centroids': 6,
   'missmatching_centroids': 8,
   'missmatching_clusters': 9,
   'None': 36,
   'missmatching_elements': 10}}}
```

The comparative analysis using expansive time windows does not reveal high percentage variations between sets of clusters. The maximum value, 1.68%, is reached between the periods 2000-2015 and 2000-2016. For fixed windows, the maximum value obtained in the period 2000-2008 and 2008-2016 does not exceed 11.76%.

The sets of clusters obtained using 15-year rolling windows are consistent over time, with no instance exceeding a 0.8% variation. As the duration of the rolling window decreases, the variability between the cluster sets increases, reaching a maximum value of 22.68% for 5-year rolling windows, 15.12% for 6-year rolling windows, and 21% for 7-year rolling windows.

Examining the results with 5-year, 6-year, and 7-year rolling windows, it can be noted that the following periods record a significant percentage of variation in all three analyses: 2010, 2013, 2019.

Furthermore, it can be observed that the following elements are commonly found among the dissonant elements: Rates (Rising)_lag12, Yield Curve (Inverted)_lag1, Yield Curve (Inverted)_lag3, Dollar (Strengthening)_lag2, Dollar (Strengthening)_lag3 and Asset Inflation (Positive).

Finally, the findings related to changes in centroids may not be as significant, as many clusters have a considerable number of variables, and their centroids can be very close to other variables, such that a not-so-substantial change can result in a shift in the centroid.

PROCEDURES NOT INCLUDED

VARIABLE REPLACEMENT

LASSO

For this procedure we followed a similar methodology as with Logistic Regression. However, we estimated our raw score using 3 values extracted from the single regressions: single regression coefficients, p-value and R-Squared. The model in this is not linear however, therefore it is not ideal to implement it with Boolean values. Frequently, the results were inconsistent compared to the outcomes of the other procedures. For this reason, we decided not to include it in our aggregated scoring mechanism.

PCA

We ran Principal Component Analysis in our Macroeconomic Boolean data. We determined the contributions of the variables for the first 7 principal components, which explain 80% of the variation. However, these results are not included in our aggregated scoring mechanism, since PCA is suitable to be used in continuous data.

MCA

We ran Multiple Correspondance Analysis in our Macroeconomic Boolean data. We found the weights of the variables in the first 9 principal coordinates. These explain almost 88% of the variation. The procedure is more appropriate for categorical data, however, it is more appropriate to use for data with 3 categories or more. Consequently, we decided not to include the results in our aggregated scoring scheme.

CLUSTERING

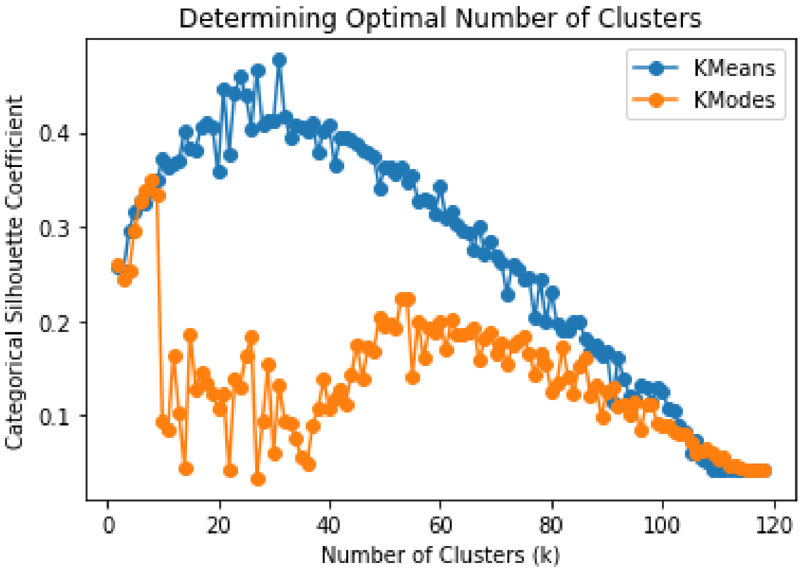
KMEANS

We utilized Scikit-learn's K-means procedure on our Boolean dataset. To execute this procedure, it is necessary to specify the number of clusters. We carried out the procedure with diverse K values (representing the number of clusters) and assessed the silhouette score for each resulting cluster set.

The distance measure employed in the K-means implementation is the Euclidean distance. Although it may seem logical to use this measure with Boolean data, it's not the most suitable choice for this data type. Consequently, the results of this procedure will not be included in the aggregated results.

KMODES

We executed the KModes algorithm from KModes on our dataset. Similar to KMeans, this procedure requires specifying the number of clusters as an input parameter. Additionally, it's necessary to specify the distance measure the algorithm will use. In this algorithm, the Custom Hamming Distance is not available as distance measure, therefore we opted for Hamming distance instead. We implemented the procedure with varying values of K and estimated the silhouette score for each cluster set obtained. The results obtained following this methodology and the one of KMeans are depicted in the lower graph.



Despite Hamming distance being a valid measure for Boolean data, it falls short in capturing the similarity between inversely related variables. Hence, the results of this procedure will not be included in our aggregated findings.

KMEDOIDS

Similarly to KMeans and KModes, implementing KMedoids requires specifying the number of clusters as an input parameter. Additionally, it is necessary to define the distance matrix the algorithm will utilize. In this case, we opted for the Custom Hamming Distance to estimate the distance matrix. We carried out the procedure with various K values and computed the silhouette score for each resulting cluster set.

As evident from the graph, the scores are rather subpar. Consequently, this approach will not be considered in the aggregation of results.

NEXT STEPS

It is also important to point out in our variable replacement results the type of relationship there exists. This can be easily extracted from the Hamming Distance: values of this measure that are close to 0 reveal a positive relationship, while values close to 1 reveal an inverse relationship. Regarding the Clustering section, the results obtained were based solely in the optimization of a coefficient: silhouette score. It would be interesting and necessary to implement the clustering procedures with other type of objectives such as minimum number of variables per cluster, cluster limitations and introduce other type of variables into the clustering universe.