COMP0050 - Machine Learning with Applications in Finance Coursework

Leo-Paul Le Meur

April 2, 2023

1 Predicting banks default

1.1 Introduction

The data provided takes has 14 attributes and one binary target variable. In order to tackle this classification problem, I will compare the performance of three models. I will first fit a logistic regression to the data. Then, I will train a Random Forest, and I will finally train a neural network. I will also include some optimization techniques to better fit the data.

1.2 Methodology

1.2.1 Pre-Processing and Over-Sampling

After loading the data, I shuffled the rows and divided them into 80% training and a 20% test set.

Since our data has a distribution of 97% of companies not defaulting and 3% of companies defaulting, I will aim to re-balance this ratio to inprove our models' performance. 5 shows that a smaller ratio of companies not defaulting over companies defaulting (referred to as the CND/CD Ratio) will result in a more efficient model.

There are two ways to deal with data imbalanced data in a classification problem: Undersampling is the process of removing randomized data from the majority category to even the ratio. We can perform undersampling clustering to regroup highly correlated samples. The problem with undersampling is that it leads to data loss. The alternative is to oversample our dataset. Oversampling is a method that increases the number of minority category samples. Here, I am oversampling using the SMOTE algorithm. SMOTE creates instances of samples by using k-nearest neighbor to find and create new values (K-Nearest Neighbor will also be used in the clustering data processing).

1.2.2 Dimensional reduction with Principal Component Analysis

Let us now evaluate the statistical significance of each feature of the model. Principal Component Analysis (now referred to as PCA) reduces the number of dimensions in our data by finding the most important patterns or structures in the data. They can be identified by finding the directions in the feature space that have the highest variance.

Here, I chose to fit data with a PCA explaining 95% of the variance. The PCA analysis reduced my data to 6 dimensions.

1.2.3 Accessing Model Performance

There are several ways to access the performance of a classification model. The simplest one is the accuracy score. The Accuracy Score gives the percentage of time the model is right on its prediction. The problem is that when faced with imbalances in the test set (oversampling only fixes imbalances in the training set!), the accuracy can be misleading as it may give high values for a model that predicts that the company defaults 0% of the time.

Another efficient metric to evaluate performance is the F1 score. The F1 score uses the confusion matrix (True Positive, False Positive, False Negative, and True Negatives) and calculates the precision

p of a model with its recall r. p and r are defined as:

$$p = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

$$r = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

The F1 is the harmonic mean of both scores.

Finally, the Area Under the Curve (now referred to as AUC) is derived from the ROC graph. The ROC graph plots the recall r with the specificity s defined as:

$$s = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}}$$

Since both values range from 0 to 1, the AUC also takes in a value of 0 to 1.

I will be using both the F1 Score and the AUC in order to compare model performance.

1.2.4 Logistic Regression

The first model to consider is logistic regression. A Logistic Regression allows the classification of data through a Cumulative Density Function (now referred to as CDF).

An n parameters logistic Regression fits a sigmoid function to data. The function can be written as:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n)}}$$

with $\beta_0, \beta_1, \cdots, \beta_n$ the coefficients to be estimated to fit our data and $X_1, X_2, \cdots, X_n$ the variables.

It is a CDF since:

$$\lim_{x \to \infty} \frac{1}{1 + e^{-x}} = 1$$

and:

$$\lim_{x \to -\infty} \frac{1}{1 + e^{-x}} = 0$$

We will estimate the parameters of the logistic regression through a maximum likelihood estimation. The likelihood function, $L(\beta)$ is a function that tracks the probability of all events in the training set happening through the probability computed by the different β. Maximizing this probability means maximizing the performance of the model on the training set. For computational reasons, I am maximizing the log of the likelihood function $log(L(\beta))$.

The function to maximize becomes:

$$log(L(\boldsymbol{\beta})) = \sum_{i=1}^{n} [y_i \log(P(Y_i = 1|\mathbf{X}_i)) + (1 - y_i) \log(P(Y_i = 0|\mathbf{X}_i))]$$

With n the number of data points in the dataset, y_i is the binary outcome of the training point (in this case 0 for no-default and 1 if the company defaults).

The algorithm used to fit this model is a Gradient Ascent.

1.2.5 Neural Network: Multi-Layer Perceptron

A Neural Network is an algorithm that tries to simulate the human brain by connecting a series of cells. A Perceptron is a single-layer neural network. The cells are divided between input layers, hidden layers, and output layers. The input layer has n neurons (or nodes), where n corresponds to the number of features. The next layer is the hidden layer. It helps in the transformation of input into output. A Perceptron without a hidden layer would have limited performance and would only be successful at capturing linear combinations in data. The last layer is the output layer. In our classification case, the output cell labels data as a probability, interpreted as 0 or 1 (default or no default).

The different possible parameters for a model are the number of cells n, and the α term, the regularization threshold. To find the optimal number of cells for my model, I used Grid Search (Bergstra Bengio, 2012). Grid search iterates through a list of possible parameters for the model data and finds the highest-performing one. The grid search function gave me an optimal number of cells $n = 256$ and the regularization term $\alpha = 0.001$. This gives me a network with a total of 261 neurons (256 in the hidden layer, 6 in the input layer, and 1 in the output layer).

	With PCA		Without PCA	
	AUC	F1 Score	AUC	F1 Score
Logistic Regression	0.80	0.28	0.80	>0.01
Neural Network	0.80	0.26	0.82	0.05
Random Forest	0.81	0.25	0.78	0.03

Table 1: F1 Score and AUC score by the model and whether PCA was used or not

1.2.6 Random Forest and Decision Tree

A decision tree is an algorithm that separates data based on its attributes. It will make decisions from attributes of the data (such as Cash $> 1,000.00$£) to classify the instance.

A decision tree gets trained through the principle of maximum entropy. To train a decision tree, we must find the cutoff maximizing the information gain.

The downside of a decision tree is that it is highly prone to its training set and is likely to overfit its data.

A Random Forest, however, divides its features under c random categories. Each sub-division of our features will be attributed to a decision tree.

I am also using grid sampling for hyperparameter tuning to test for various number and depths of trees.

1.3 Results

5 accesses the impact of undersampling. It clearly shows that a model with a more equal data repartition tends to outperform. It is however worth mentioning that undersampling can lead to information losses, which would risk underfitting the model, hence my use of oversampling.

"Dealing with imbalanced data is a crucial step in building a reliable model, as learning from imbalanced data can lead to suboptimal performance (He Garcia, 2009) [3].1 Also shows the large impact that oversampling has had on the performance of our model. The F1 Score difference is highly significant and shows how models with imbalances risk underperforming.

Overall, the best model is the Logistic Regression, with data rebalancing, since it holds the best F1 score and more or less the same AUC as other models. The difference in AUC between the two Random Forests shows that a Random Forest performs better in higher dimensions.

2 Predicting the industry of a stock through clustering

2.1 Introduction

The data provided is given through a CSV file with 48,200 daily returns (since July 1926). I will then perform a K-means clustering analysis, on the industries, that I will compare with the results of a hierarchical clustering analysis. I am assuming that the clustering of industries depends on the economic period, and will therefore perform a clustering analysis on various economic periods (WW2, Cold War, Tech boom). I took the following economic periods:

- Great Depression (1929-39) A period of low economic growth
- Stagflation (1970)
- Recession Recovery (2009 - 2019)

Unfortunately, I encountered an error for World War 2 and the data histogram did not work... It might have been the most interesting analysis to do since it is the period that differs the most from the others.

2.2 Methodology

2.2.1 Data Cleaning

I first dropped the first column of the data since it presented a DateTime object.

Then, the CSV file has some NaN missing values. Before implementing PCA, I had to remove them.

I, therefore, used imputation, a method to replace missing values with other values by predicting them. According to Azimi and Seifi [3], k-nearest neighbor imputation is effective in handling missing data in complex datasets, hence my use of k-nearest-neighbor imputation.

K-nearest neighbor finds the lowest Euclidian distance d in rows without NaN value and sets the value NaN to the closest neighbor’s. Here the Euclidian distance is defined as:

$$d = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2}$$

with n being the number of features in our sample space, x, and y being the two points.

Now that the data has been loaded properly and has the right format, I could perform the clustering analysis.

2.2.2 K-Means clustering

The first algorithm to consider for clustering is K-Means. It is a supervised learning method since it requires as input the desired number of clusters k.

It operates in the following way:

1. Initialization: K-means selects k random numbers in the set.
2. Assign each point of the dataset to a cluster using Euclidian Distance (defined above)
3. Recalculate cluster centroids using the mean of every data point associated with each cluster
4. Repeat steps 2 and 3 until the clusters converge.

The optimal number of clusters k can be found with an elbow graph, plotting the reduction of the variance of each cluster by the number of clusters.

This elbow plot measures the inertia I or the within-cluster sum of squares in the function of the number of clusters.

$$I = \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||^2$$

The point where the slope slows down the most is the optimal number of clusters. In our case, the graph looks fairly ambiguous, but we may still decide that the optimal number of clusters is 12.

2.2.3 Hierarchical Clustering

In this specific case, hierarchical clustering seems to be appropriate since industries follow hierarchical patterns. For example, ”food”, ”soda” and ”beer” should be regrouped as a cluster of ”Food and Beverages”.

Hierarchical clustering is a model that regroups columns with their closest neighbors. The algorithm continues the cluster until it grouped every column together.

```
Cluster 0: [’Soda ’, ’Hlth ’]
   Cluster 1: [’Food ’, ’Beer ’, ’Smoke’, ’Books’, ’Hshld’, ’Clths’, ’Drugs’, ’FabPr’, ’Guns ’, ’Util
’, ’Telcm’, ’PerSv’, ’BusSv’, ’Boxes’, ’Whlsl’, ’Rtail’, ’Meals’, ’Banks’, ’Insur’, ’Other’]
   Cluster 2: [’Rubbr’]
   Cluster 3: [’Paper’]
   Cluster 4: [’RlEst’]
   Cluster 5: [’Toys ’]
   Cluster 6: [’Cnstr’]
   Cluster 7: [’Fun ’, ’Chems’, ’Txtls’, ’BldMt’, ’Steel’, ’Mach ’, ’ElcEq’, ’Autos’, ’Aero ’, ’Ships’,
’Mines’, ’Oil ’, ’Comps’, ’Chips’, ’LabEq’, ’Trans’, ’Fin ’]
   Cluster 8: [’Agric’]
   Cluster 9: [’Gold ’]
   Cluster 10: [’MedEq’]
   Cluster 11: [’Coal ’]
```

Results of the k-mean clustering

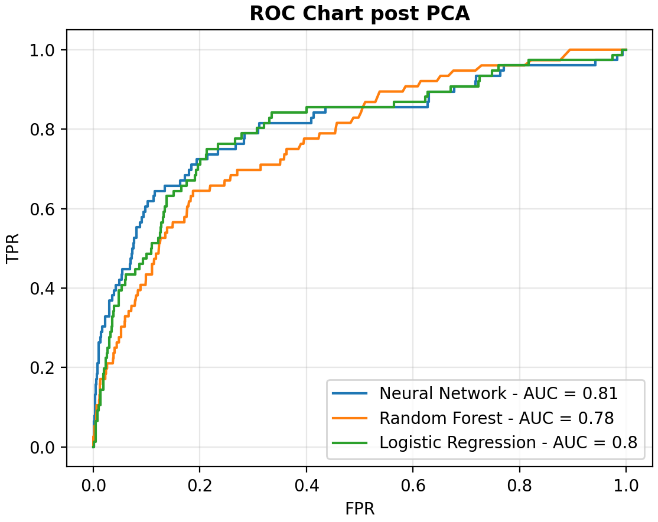


Figure 1: ROC Chart for models with PCA Dimensional Reduction

2.3 Results

In 6,7,9,8, the hierarchical clustering of the different time periods proved my point. Industries cluster differently in different economic periods.

An application of this data could be to classify how the market views our current economic period using the daily returns of each industry.

Some interesting facts about our clusters are that Gold, Coal, and Paper tend to be very uncorrelated to the overall market. An interesting follow-up analysis would be to look at the beta of companies in these industries.

References

[1] Azimi, J., & Seifi, A. R. (2017). Using imputation techniques for handling missing data in complex surveys. Journal of Research in Health Sciences, 17(2)

[2] He, H., Garcia, E. A. (2009). Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 21(9), 1263-1284. DOI: 10.1109/TKDE.2008.239

[3] Bergstra, J., Bengio, Y. (2012). Random search for hyper-parameter optimization. Journal of Machine Learning Research, 13(Feb), 281-305.

3 Appendix

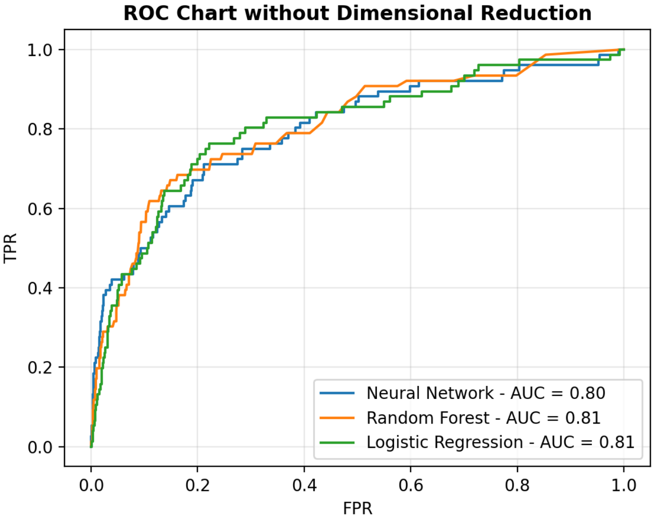


Figure 2: ROC Chart for models without PCA Dimensional Reduction.

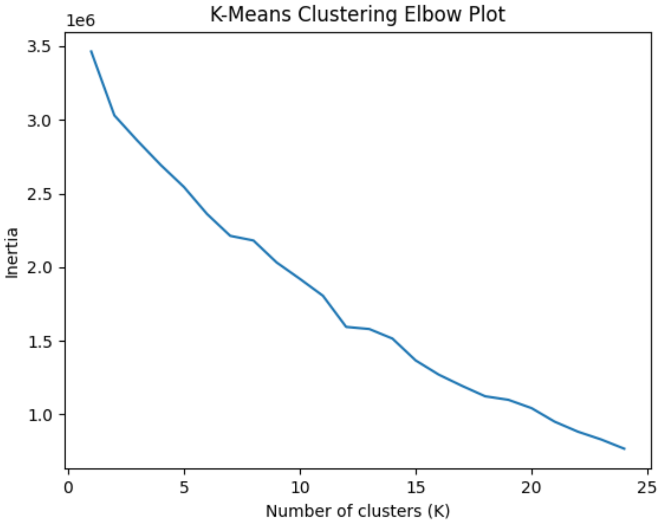
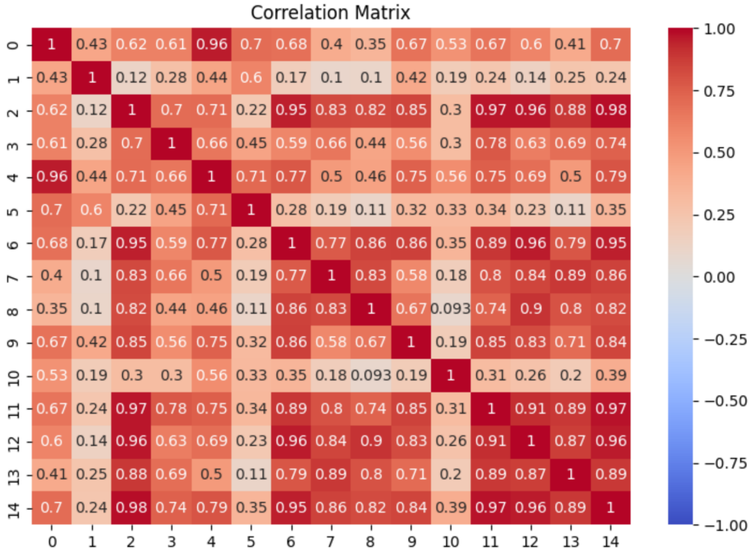


Figure 3: Elbow plot for the K-Means clustering algorithm. Here, the optimal number of clusters seems to be 12



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	1	0.43	0.62	0.61	0.96	0.7	0.68	0.4	0.35	0.67	0.53	0.67	0.6	0.41	0.7
1	0.43	1	0.12	0.28	0.44	0.6	0.17	0.1	0.1	0.42	0.19	0.24	0.14	0.25	0.24
2	0.62	0.12	1	0.7	0.71	0.22	0.95	0.83	0.82	0.85	0.3	0.97	0.96	0.88	0.98
3	0.61	0.28	0.7	1	0.66	0.45	0.59	0.66	0.44	0.56	0.3	0.78	0.63	0.69	0.74
4	0.96	0.44	0.71	0.66	1	0.71	0.77	0.5	0.46	0.75	0.56	0.75	0.69	0.5	0.79
5	0.7	0.6	0.22	0.45	0.71	1	0.28	0.19	0.11	0.32	0.33	0.34	0.23	0.11	0.35
6	0.68	0.17	0.95	0.59	0.77	0.28	1	0.77	0.86	0.86	0.35	0.89	0.96	0.79	0.95
7	0.4	0.1	0.83	0.66	0.5	0.19	0.77	1	0.83	0.58	0.18	0.8	0.84	0.89	0.86
8	0.35	0.1	0.82	0.44	0.46	0.11	0.86	0.83	1	0.67	0.093	0.74	0.9	0.8	0.82
9	0.67	0.42	0.85	0.56	0.75	0.32	0.86	0.58	0.67	1	0.19	0.85	0.83	0.71	0.84
10	0.53	0.19	0.3	0.3	0.56	0.33	0.35	0.18	0.093	0.19	1	0.31	0.26	0.2	0.39
11	0.67	0.24	0.97	0.78	0.75	0.34	0.89	0.8	0.74	0.85	0.31	1	0.91	0.89	0.97
12	0.6	0.14	0.96	0.63	0.69	0.23	0.96	0.84	0.9	0.83	0.26	0.91	1	0.87	0.96
13	0.41	0.25	0.88	0.69	0.5	0.11	0.79	0.89	0.8	0.71	0.2	0.89	0.87	1	0.89
14	0.7	0.24	0.98	0.74	0.79	0.35	0.95	0.86	0.82	0.84	0.39	0.97	0.96	0.89	1

Figure 4: Correlation Matrix of the 14 attributes of the data

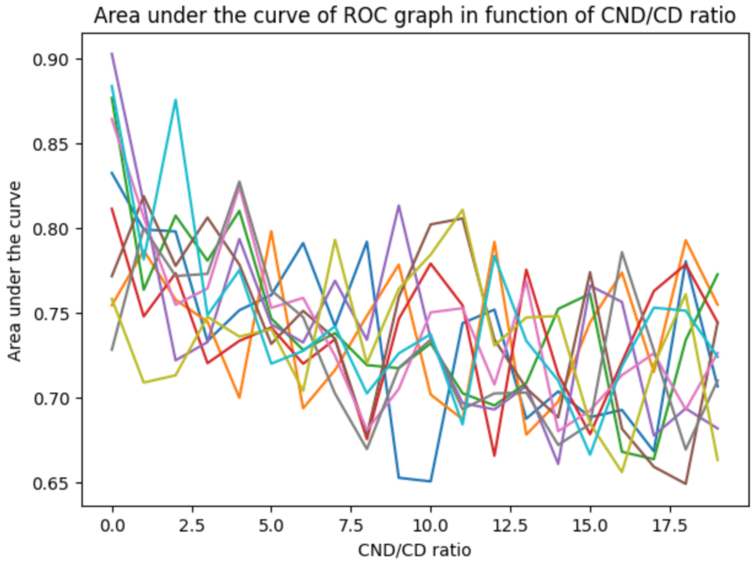


Figure 5: Impact of Under-Sampling Logistic Regressions on the performance of Logistic Regressions

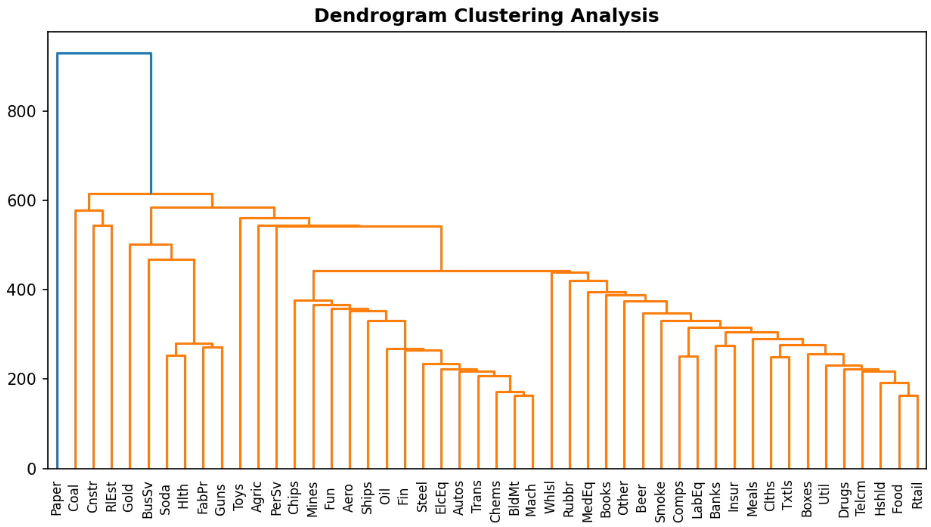


Figure 6: Dendrogram of the 48 industries on the entire dataset

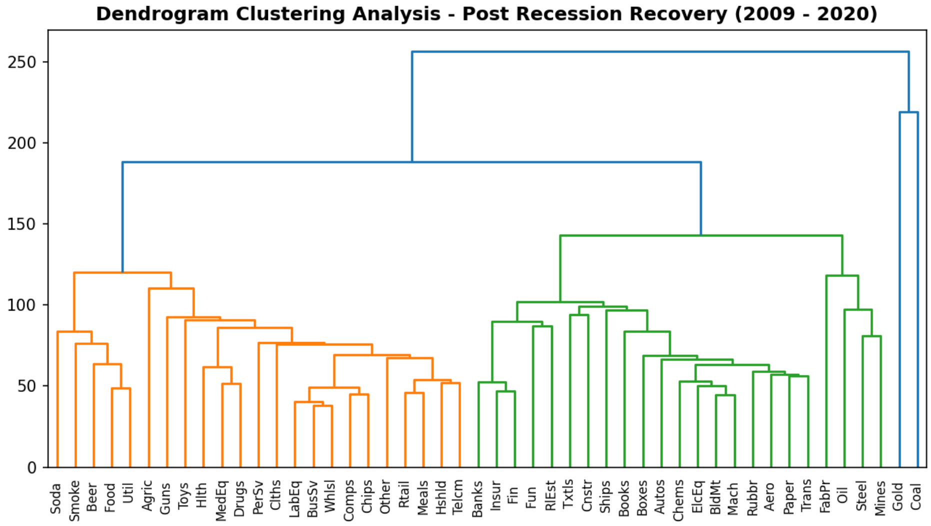


Figure 7: Dendrogram on the Post-Recession Recovery

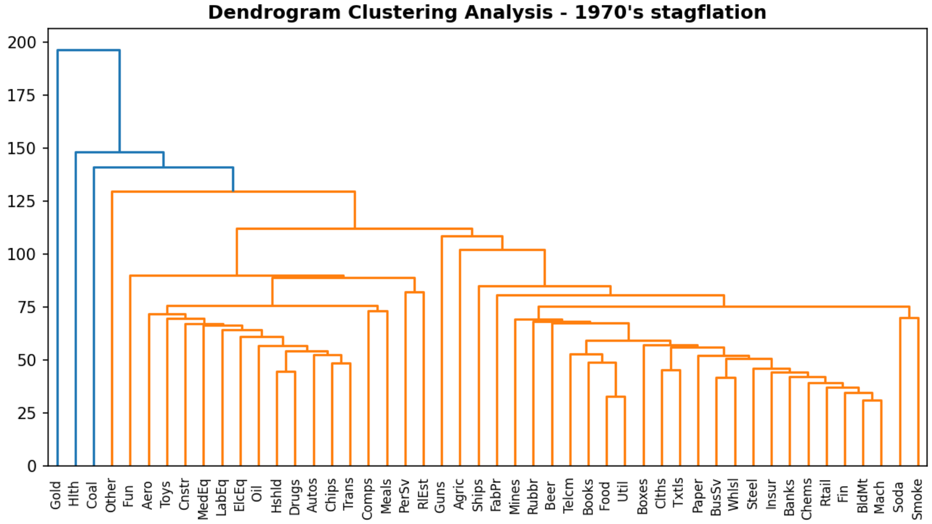


Figure 8: Dendrogram on 1970’s stagflation

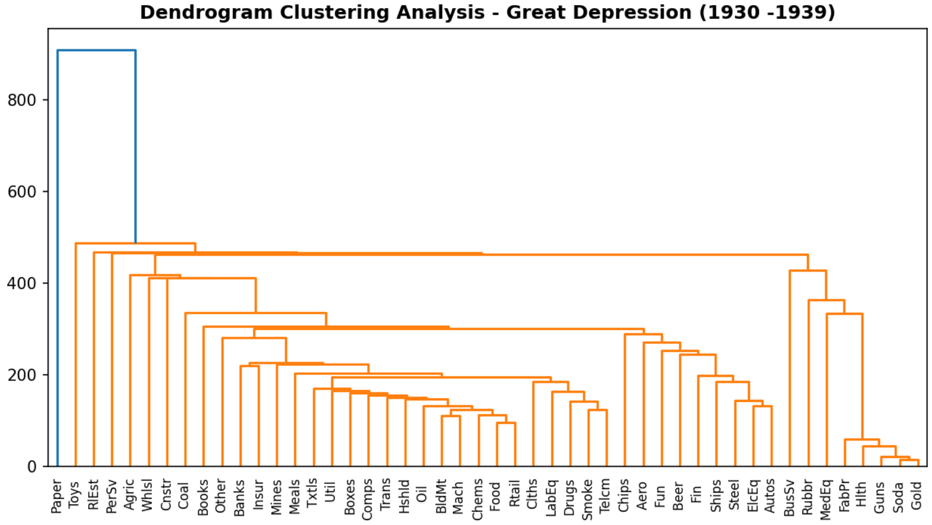


Figure 9: Dendrogram on the Great Recession