Power of Autoencoder and t-SNE in Stock Deep Clustering and its Application to Fama-French Factor Model

Bonwoo Koo[1], Kyuho Lee[2], Seongtag Sin[3], Bosung Park[2]

[1]Department of Industrial and Systems Engineering, Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea
[2]School of Computer Science, Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea
[3]School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea

Abstract. This paper examines advanced machine learning techniques to improve stock clustering based on financial data and price movements. Initially, simple clustering techniques such as K-means clustering yielded suboptimal and unexplainable results, evidenced by low silhouette scores and high Davies-Bouldin indices. By incorporating additional financial features in high dimension and using an autoencoder for dimensionality reduction followed by t-SNE, we capture more complex and non-linear relationships more effectively than with PCA. The optimal results were achieved using combined fundamental and extended financial data processed through an autoencoder and t-SNE, followed by K-means clustering. This approach produced distinct clusters with lower intra-cluster variance, confirmed by improved silhouette scores and Davies-Bouldin indices. Key financial features, such as market capitalization, current assets, and liabilities, were critical in defining clusters. These clustering results suggested new factors, such as liquidity and leverage, which can be incorporated into the Fama-French factor model to enhance the explanation and prediction of stock returns based on financial properties. This study highlights the potential of integrating deep learning and machine learning techniques with traditional financial analysis to uncover new explanatory factors, providing valuable insights for financial engineering and investment strategies.

Keywords: *Clustering, Autoencoder, Dimensionality Reduction, Fama-French Model, AI in Finance*

1 Introduction

In the previous AI practice assignment, we utilized the financial fundamental data of stocks to segment them into clusters using simple clustering techniques like K-means clustering. Stocks grouped in the same cluster would exhibit similar characteristics, such as price movements and financial properties. From the assignment results, clusters were primarily formed based on market capitalization values, which is a historically renowned factor for segmenting stocks. However, other factors apart from market capitalization did not significantly contribute to the clustering. Additionally, the visualization of clustering in two and three dimensions was unsatisfactory, as the clusters were not distinctly formed.

Figure 1: Visualization of clusters of raw data and the PCA-reduced data from AI Practice 2

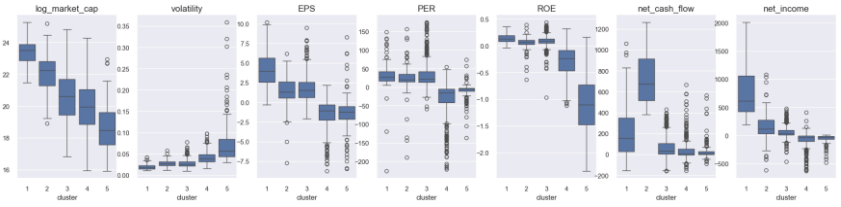


Figure 2: Financial properties analysis of clusters from AI Practice 2

To address these limitations, we incorporated several improvements to ensure better differentiation between clusters and better cohesion among stocks within clusters. We first merged all available financial properties datasets and decided to utilize price data to incorporate mean return and standard deviation as additional features for clustering. Subsequently, we applied an autoencoder to extract the latent vector with lower dimension and applied clustering after processing the results through additional dimension reduction technique of TSNE. Unlike PCA, which is linear and preserves global structure, t-SNE is non-linear, preserving local structures and better capturing complex data relationships. This combination addresses PCA's limitations by providing a more nuanced visualization of the clustered data. It was predicted that utilizing a larger dataset and employing an autoencoder would ultimately yield better clustering results. To

support this hypothesis, we conducted experiments by controlling the use of each dataset and the application of the autoencoder.

In this experiment, our goal is to explore the potential of deep & machine learning techniques in analyzing the statistical implications of the time-series data hidden within each financial information and stock price movement. We aim to divide stocks into several clusters based on our clustering methodology. To have significance in terms of financial engineering, it is crucial to examine whether the stocks within these groups exhibit similar returns and risks. As a solution, we intend to interpret our experimental results from the perspective of factor models like the Fama-French factor Model. We aim to identify new factors or refine existing factors that might explain stock returns based on our clustering results. This will provide us an insight of uncovering new factors that might not be immediately evident through traditional analysis.

2 Related Work

Clustering techniques have been widely utilized in finance, ranging from classifying securities to segmenting financial customers for services like portfolio optimization and management. These methods have evolved significantly over time, starting from basic classification and genetic algorithms to more sophisticated approaches like K-means and hierarchical clustering, enabling more precise and insightful analyses.

In 2004, Pattarin et al. explored the use of clustering for mutual funds style analysis in their paper "Clustering Financial Time Series: An Application to Mutual Funds Style Analysis." They introduced a robust evolutionary clustering methodology combined with principal component analysis for dimensionality reduction and a constrained regression model for style identification. This approach successfully aligned with existing classification schemes and explained out-of-sample variability in fund returns, showcasing the effectiveness of advanced clustering techniques in financial contexts (Pattarin et al., 2004).

Building on these foundations, Bini B. Sa and Tessy Mathew in 2015 investigated the integration of clustering and regression techniques for stock prediction. They employed K-means clustering to categorize stocks with similar behaviors, which were then analyzed using various regression models to forecast future stock prices. Their hybrid approach leveraged the strengths of both clustering and regression to enhance prediction accuracy by addressing the non-linearity and volatility inherent in stock market data (Bini & Mathew, 2015).

More recently, in 2023, Hwang et al. extended the application of clustering techniques to household finance in their paper "Identifying Household Finance Heterogeneity via Deep Clustering." This study emphasizes the use of deep clustering to improve the performance of clustering compared to traditional methods. The authors used a deep learning-based clustering method to analyze high-dimensional balance sheet data of

approximately 50,000 households. By employing advanced dimension-reduction techniques, they managed to incorporate the full joint distribution of high-dimensional data in the clustering process. This approach significantly enhances the understanding of household finance heterogeneity by identifying crucial asset and debt variables and their correlations with sociodemographic factors such as age, education, and family size (Hwang et al., 2023).

Motivated by the significant performance improvements and detailed insights achieved in Hwang et al.'s work, we propose applying similar methodologies to stock data. We also aim to enhance our previous stock classification assignment by incorporating deep clustering methods. This approach will involve analyzing high-dimensional stock data with advanced dimension-reduction techniques, considering the variety of financial data. By conducting rigorous clustering experiments with various techniques, we seek to explore the potential of deep & machine learning methods in finance. This could facilitate uncovering new financial factors that can be integrated into traditional financial engineering models, like the Fama-French factor model, to achieve more precise predictions and deeper insights into stock market dynamics.

3 Data

The data provided in the previous assignment consisted of three files: *financial_statements.csv, financial_statements_extension.csv*, and a file containing price data of individual stocks named *security_daily.ftr*. These data are obtained from WRDS (Wharton Research Data Services). The financial_statements.csv file contains fundamental metrics such as market cap, PBR, PER, EPS, ROE, commonly used to compare the financial characteristics of stocks, whereas the extension data includes more detailed fundamental metrics. In the previous assignment, clustering was performed using each dataset individually. However, in this project, the goal is to merge the two data files to conduct clustering on all available fundamentals and compare the clustering results. Moreover, we added price data of each stock which corresponds to the period of financial statement data obtained: mean returns and standard deviation of daily, weekly, monthly, three-month, and six-month periods. For subsequent experiments, clustering would be conducted using the same methodology on three datasets: one containing only the fundamentals from *financial_statements.csv* labeled as *'original*,' another combining the original data with the extension data labeled as *'original + extension*,' and the third combining the previous dataset with price-derived mean return and standard deviation data labeled as *'original + extension + price*.' The results of each clustering would be compared and analyzed accordingly.

4 Methodology

4.1 Data Preprocessing

4.1.1 Data Merge

As mentioned before, we merge three types of data regarding company stocks: financial statements, extension financial statements, and mean return and standard deviation of stock prices. During this process, tickers that did not appear in all three datasets were excluded, and NaN values were imputed with the closest preceding value.

4.1.2 Data Scaling

As a data scaling technique, we implement log transformation and standardization on the raw dataset. Log transformation applies the natural logarithm to the values in a dataset to stabilize the variance and make the large scaled financial data more normally distributed. Standardization transforms the data to have a mean of 0 and a standard deviation of 1. This ensures that all features contribute equally to the model's clustering performance and that the model is not biased towards features with larger scales.

4.2 Autoencoder

Autoencoder is a dimension reduction technique based on neural networks and is composed of two parts, encoder and decoder. Encoder compresses high-dimensional data into a lower-dimensional latent space, and decoder takes the latent space and tries to reconstruct the original input data as closely as possible, and this whole process is optimized by minimizing the reconstruction error. Latent vectors obtained from this process can capture the essential patterns and structures of the input data layer. Thus, application of autoencoder in stock clustering is expected to improve the performance of clustering algorithms such as K-means clustering and spectral clustering since the latent space often highlights the natural groupings in the data better than the original high-dimensional space. Moreover, the fact that autoencoders can model non-linear relationships, which are often exhibited by financial data, is expected to give comparative advantage over traditional linear dimensionality reduction techniques like PCA. When an autoencoder is used for the 'original' data or price data, the input data layer is compressed to 4 latent vectors since 'original' data and price data contains 8 columns of stock features only. However, when 'original + extension' data or 'original + extension + price' data are used as input data, they are reduced to 8 latent vectors since a larger number of latent vectors would better preserve essential patterns and those two dataset contain 32 and 24 features respectively.

4.3 t-SNE

t-SNE is a non-linear dimensionality reduction technique designed to preserve the local structure of data while embedding it in a lower-dimensional space to typically two or three dimensions. This process is achieved by minimizing the divergence between two probability distributions through optimization, where one measures pairwise similarities in the high-dimensional space using Gaussian distribution and the other one measures pairwise similarities in the low-dimensional space using Student's t-distribution. Application of t-SNE has an advantage over PCA in that it can capture non-linear relationships in financial data that exhibit complex, intricate, non-linear relationships. It is used as a preprocessing step to reduce dimensionality before applying clustering algorithms such as K-means clustering and spectral clustering. By reducing the dimensionality in a way that maintains the distances between similar points, t-SNE can create a representation where clusters are more distinct and separated. This makes it easier for clustering algorithms to identify and separate these clusters. Moreover, since high-dimensional data often contains noise that can interfere with clustering performance, t-SNE can reduce the influence of noise by emphasizing the most relevant structures. This preprocessing step can lead to more robust and accurate clustering results.

4.4 Clustering

4.4.1 K-means Clustering

K-means clustering is a clustering technique used to find patterns from unlabeled data to organize similarity groups of clusters. The algorithm works by initializing k centroids in the data and repeating two steps until convergence: assigning each data point to its closest centroid and moving each centroid to the center of data points assigned to it. Silhouette score and Davies-Bolding index are used for the performance evaluation of clustering. It is used to identify clusters where each cluster feature explains different classes of stock. K-means clustering is applied after dimension reduction through t-SNE to four different datasets, 'original' data, 'original + extension' data, 'original + extension + price' data and price data. These four experiments are repeated using an autoencoder before application of t-SNE. The clustering results of four experiments not using autoencoder and four experiments using autoencoder are compared by determining and comparing Silhouette score and Davies-Bolding index for each experiment. Then the experiments with best and worst clustering results are further studied by analyzing financial properties box plot, mean table and correlation table.

4.4.2 Spectral Clustering

Spectral clustering is a technique used to find patterns in unlabeled data by leveraging the eigenvalues of similarity matrices to perform dimensionality reduction before clustering in fewer dimensions. The algorithm works by constructing a similarity graph from the data, computing the Laplacian of the graph, and then using the eigenvalues of

the Laplacian to reduce dimensions. The reduced dimensionality data is then clustered using traditional methods like K-means. Performance evaluation of clustering results is conducted using Silhouette score and Davies-Bouldin index. Spectral clustering is applied after dimension reduction through t-SNE to four different datasets, 'original' data, 'original + extension' data, 'original + extension + price' data, and price data. These four experiments are repeated using an autoencoder before the application of t-SNE. The clustering results of four experiments not using an autoencoder and four experiments using an autoencoder are compared by determining and comparing Silhouette score and Davies-Bouldin index for each experiment. Then, the experiments with the best and worst clustering results are further studied by analyzing financial properties box plot, mean table, and correlation table.

5 Experiments

The experiments comprise eight distinct scenarios, each involving one of four different datasets, with and without the application of an autoencoder. All scenarios include the use of t-SNE dimension reduction before clustering.

Table 1: Data and Methodology Combination of Each Experiment

Experiment	Data	Use of Autoencoder
Experiment 1	Original Financial Statements	X
Experiment 2	Original + Extension Financial Statements	X
Experiment 3	Original + Extension Financial Statements + Price	X
Experiment 4	Only Price	X
Experiment 5	Original Financial Statements	O
Experiment 6	Original + Extension Financial Statements	O
Experiment 7	Original + Extension Financial Statements + Price	O
Experiment 8	Only Price	O

6 Results

6.1 Effect of Autoencoder

6.1.1 K-means Clustering

Silhouette score and Davies-Bouldin index obtained after K-means clustering using 4 different datasets are compared to the experiment done using autoencoder before dimension reduction by t-SNE. Experiments 1 and 5 use 'original' data, 2 and 6 use 'original + extension' data, 3 and 7 use 'original + extension + price' data, and 4 and 8 use price data. For each experiment, cluster size is varied from 3 to 6 and Silhouette score and Davies-Bouldin index are recorded as shown in table below.

Table 2: Silhouette score and Davies-Bouldin index in K-means Clustering

K-means clustering								
Cluster size	n = 3		n= 4		n= 5		n= 6	
Score	Silhou	Davies	Silhou	Davies	Silhou	Davies	Silhou	Davies
Without Autoencoder								
Exp 1	0.352	1.0	0.353	0.983	0.361	0.931	0.372	0.923
Exp 2	0.431	0.828	0.369	0.975	**0.35**	1.037	0.373	0.968
Exp 3	0.353	**1.071**	0.361	1.039	0.369	0.969	0.383	0.968
Exp 4	0.422	0.833	0.399	0.884	0.408	0.879	0.417	0.829
With Autoencoder								
Exp 5	0.478	1.038	0.509	0.947	0.558	0.888	**0.576**	0.847
Exp 6	0.386	1.049	0.445	0.963	0.479	0.809	0.521	**0.679**
Exp 7	0.415	0.975	0.407	0.947	0.475	0.756	0.493	0.736
Exp 8	0.428	0.847	0.417	0.897	0.478	0.688	0.468	0.685

In general, experiment 5 to 8 show higher Silhouette score and lower Davies-Bouldin index compared to that of experiment 1 to 4. This matches with earlier expectations and supports an argument that the use of autoencoder before dimension reduction improves the result of clustering. As marked in the table above, experiment 5 with n=5 showed highest Silhouette score and experiment 6 with n=6 showed lowest Davies-Bouldin index. Moreover, experiment 2 with n=5 showed lowest Silhouette score and experiment 3 with n=3 showed highest Davies-Bouldin index.

Clustering with the use of the autoencoder results in much clearer and distinguishable clusters in a 3 dimension scatter plot with 3 features.

Table 3: 2D and 3D Scatter plot of 6 clusters in K-means Clustering

Dimension	2 features	3 features

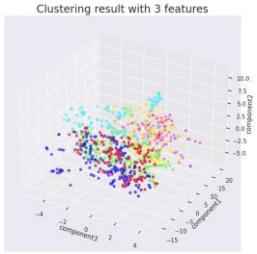
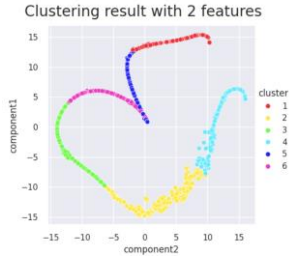
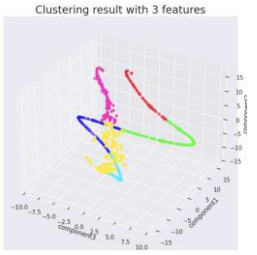
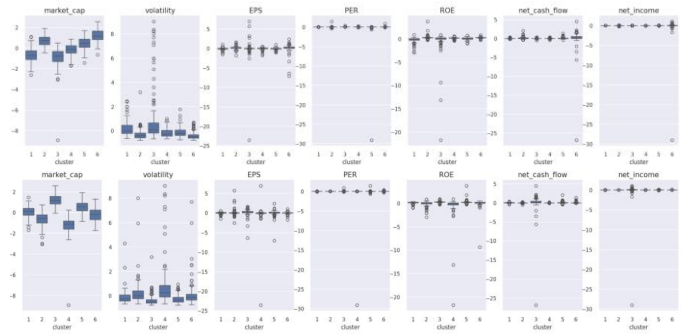
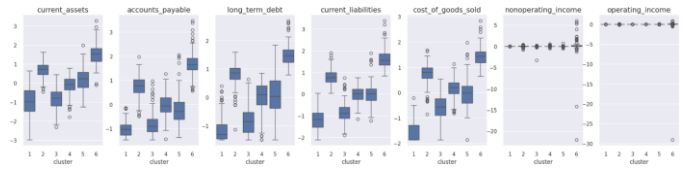
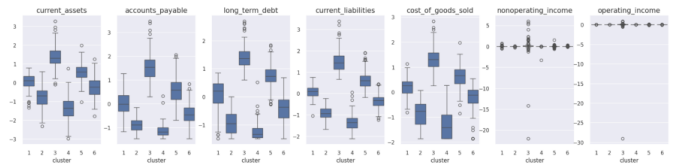
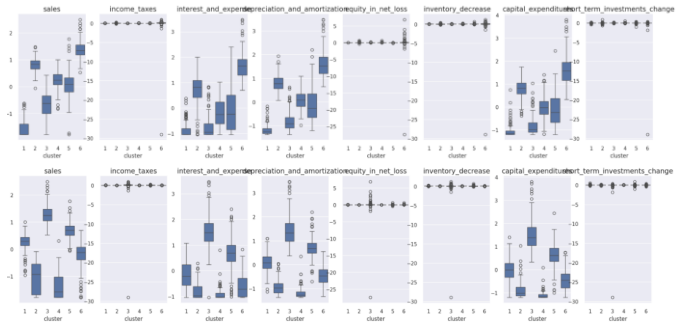
Experiment 2 (n=6)		
Experiment 6 (n=6)		

Table 4: Comparison of Box plot of experiment 2 and 6 in K-means

Comparison of box plot for experiment 2 (above) and experiment 6 (below)	
Original data	
Extension data 1	

	
Extension data 2	

Clustering with use of an autoencoder results in clusters that show more distinguishable features in long term debt and current liabilities among the original dataset. Among the extension data, clusters show better separation in features such as sales and depreciation and amortization. Although it does not show significant difference, clusters obtained with the use of autoencoder show smaller width of box plot, meaning that stocks exhibit smaller data variance within the same cluster. Thus, interpretation of boxplot leads to the conclusion that the use of autoencoder improves clustering through more precise data gathering which results in lower data variance of clusters. This interpretation in turn explains the higher Silhouette score obtained when autoencoder was used.

6.1.2 Spectral Clustering

Similar to how K-means Clustering experiments were conducted, Silhouette score and Davies-Bouldin index obtained after Spectral Clustering using 4 different datasets are compared to the experiment done using autoencoder before dimension reduction by t-SNE.

Total of 8 experiments were conducted, and their compositions are the same as K-mean's case.

Table 5: Silhouette score and Davies-Bouldin index in Spectral Clustering

Spectral Clustering								
Cluster size	n = 3		n= 4		n= 5		n= 6	
Score	Silhou	Davies	Silhou	Davies	Silhou	Davies	Silhou	Davies
Without Autoencoder								
Exp 1	0.076	**1.398**	0.17	1.144	0.171	1.265	0.225	0.982
Exp 2	-0.013	1.074	**-0.053**	0.981	0.258	0.765	0.199	0.799
Exp 3	0.154	0.831	0.13	0.799	0.286	0.829	0.173	0.892
Exp 4	0.372	0.838	0.365	0.833	0.398	0.862	0.368	0.869
With Autoencoder								
Exp 5	0.057	1.508	0.084	1.351	0.168	1.259	0.274	0.979
Exp 6	0.341	1.19	0.385	1.01	0.435	0.837	0.387	0.808
Exp 7	0.291	1.176	0.353	1.133	0.426	0.915	**0.436**	0.85
Exp 8	0.372	0.838	0.404	0.879	0.429	0.819	0.362	**0.754**

In terms of Silhouette score, experiment 5~8 has a higher score compared to experiment 1~4. Highest Silhouette score is observed at experiment 7 with 6 clusters, while lowest Silhouette score is observed at experiment 2 with 4 clusters: negative silhouette score signifies that it is almost misclassified. In terms of Davies Bouldin Index, the lowest result is obtained at experiment 8 with 6 clusters, and highest score was observed at experiment 1 with 3 clusters. Lower value means better classification, but in general, result of the Davies Bouldin index of experiment 1~4 was lower then experiment 5~8, which might signifies the inefficiency of autoencoder when used with Spectral Clustering, since in K-means case, experiments with autoencoder tend to had better result.

As before, the comparison of autoencoder results and non-autoencoder results in a 3 dimensional scatter plot reveal a clear difference between the clustering accuracy of the experiments.

Table 6: 2D and 3D Scatter plot of 6 clusters in Spectral Clustering

Dimension	2 features	3 features

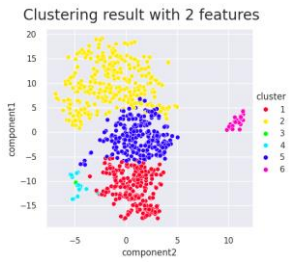
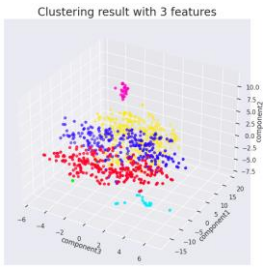
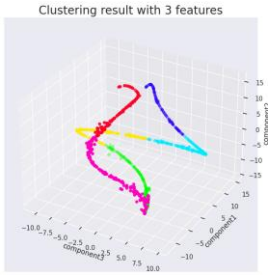
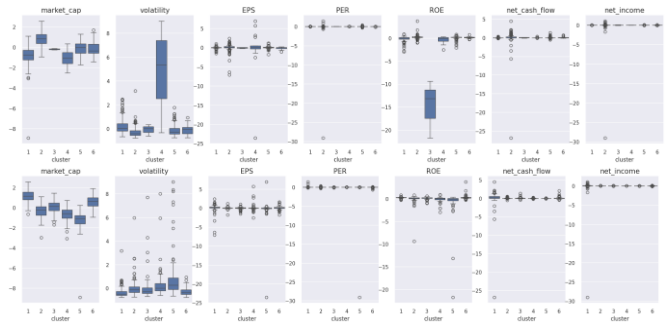
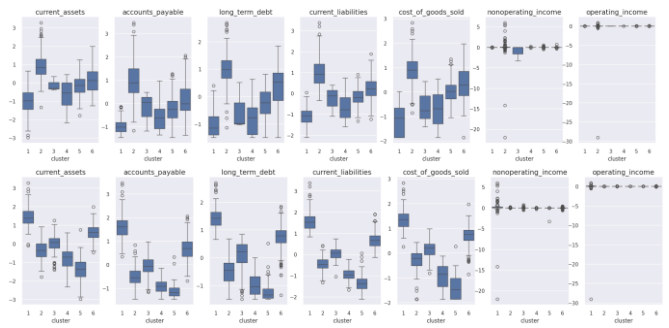
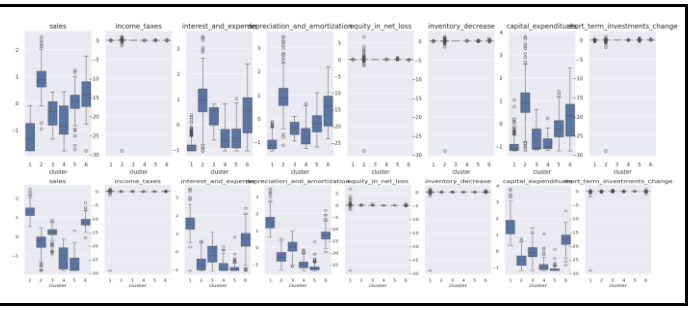
Experiment 2 (n=6)		
Experiment 6 (n=6)		

Table 7: Comparison of Box plot of experiment 2 and 6 in Spectral Clustering

Comparison of box plot for experiment 2 (above) and experiment 6 (below)	
Original data	
Extension data 1	

Extension data 2	

Similar to the K-means experiment's result, Clustering with use of an autoencoder showed better results in most of the features, as width of box plot significantly decreased compared to the experiments without autoencoder. This signifies variance of the clusters about each feature is comparably lower. However, unlike K-means clustering result, extension data2 features were not well separated - some of the clusters had similar values along the same features. For example, sales feature and interest_and_expense feature showed poor separation result, compared to K-means clustering result. This explains why the Highest silhouette score of Spectral clustering was lower than the Highest silhouette score of the K-means clustering result.

6.2 Effect of Price Data

For the experiments that did not use autoencoder, experiment 4 which uses price data as input data shows best clustering result with highest Silhouette score and lowest Davies-Bouldin index. However, clustering result of experiment 3 which uses both price data and financial data is not distinguishably better compared to the clustering result of experiment 2 which does not include price data as input features. This result supports an argument that features of price data itself result in good clustering results but, it does not improve clustering result when it is used together with other financial features.

7 Discussion of Best Model

7.1 Cluster Definition

After conducting experiment 1 to 8 using different sets of data with and without autoencoder, experiment 6 turned out to show the best clustering result among the 8 experiments. Considering the metrics scores, visualization results, and financial properties analysis of optimal models between K-means and Spectral clustering, the model with K-means clustering demonstrated more reliable and explanatory results. Thus, our optimal model would be using **'original + extension' data** as input data and incorporating **autoencoder** before dimension reduction through **t-SNE** and clustering by **K-means clustering** method while setting **the number of clusters to 6**.

Table 8: Box plot of experiment 6 of K-means

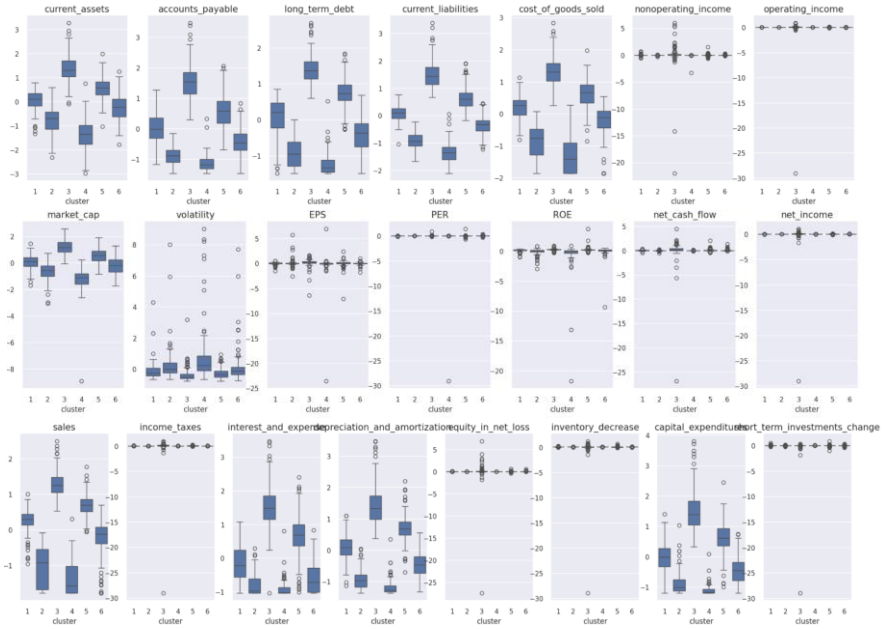


Table 9: Mean Tables of experiment 6 of K-means

Mean table of features	

cluster	current_assets	accounts_payable	long_term_debt	current_liabilities	cost_of_goods_sold	nonoperating_income	operating_income
1	0.037906	0.002118	0.072963	0.101732	0.207565	0.004668	0.020756
2	-0.760228	-0.878160	-0.918654	-0.926043	-0.920285	-0.008629	0.018722
3	1.393411	1.559173	1.403776	1.512259	1.328691	0.022839	-0.113308
4	-1.366077	-1.129168	-1.243556	-1.348411	-1.362018	-0.039339	0.020271
5	0.552285	0.559399	0.736254	0.625038	0.620003	0.008889	0.029704
6	-0.241535	-0.435747	-0.440832	-0.343592	-0.248360	0.001773	0.018654

cluster	market_cap	volatility	EPS	PER	ROE	net_cash_flow	net_income
1	0.025912	-0.136017	0.000655	0.032517	0.089443	-0.015666	0.028573
2	-0.637653	0.190321	0.035451	0.031770	-0.112121	-0.038436	0.029033
3	1.206830	-0.398936	0.149044	0.041216	0.211885	0.039372	-0.150377
4	-1.244792	0.778396	-0.223062	-0.230549	-0.546103	-0.041396	0.031108
5	0.522884	-0.292176	0.008009	0.034634	0.193992	0.030650	0.029360
6	-0.232829	0.074734	-0.018216	0.032797	0.008286	0.007213	0.029013

sales	income_taxes	interest_and_expense	depreciation_and_amortization	equity_in_net_loss	inventory_decrease	capital_expenditures	short_term_investments_change
0.254624	0.025313	-0.142676	0.090137	0.003717	0.031723	-0.012298	2.9752
-0.999846	0.023938	-0.814018	-0.953449	0.001967	0.026900	-0.890060	3.9367
1.290545	-0.132382	1.513109	1.468679	-0.029256	-0.158106	1.499009	-3.9367
-1.371534	0.024049	-0.934140	-1.179742	0.002096	0.026251	-1.100186	3.9367
0.698656	0.030044	0.828241	0.890094	0.009971	0.040203	0.823150	1.9367
-0.262337	0.024339	-0.580850	-0.471960	0.008500	0.028049	-0.455965	1.9367

Table 4 summarizes the box plot for features of 6 different clusters and table 5 is a mean table of each feature for 6 different clusters. It is observable that cluster 3, which has the largest market cap among 6 clusters, also possesses the largest value for most of the remaining features compared to other clusters. Thus, it leads to the conclusion that each cluster has a varying size of each feature in sequential order. From cluster 3 being the cluster with the greatest size of all features, clusters possess features with

decreasing amounts in order of 3, 5, 1, 6, 2, 4, starting from the largest to the smallest. Each clusters can be named according to their features as following:

Table 10: Labeling Each Cluster

Cluster 1	Mid-cap stocks
Cluster 2	Mid to small-cap stocks
Cluster 3	Large-cap stocks with most leverage
Cluster 4	Small-cap stocks with least leverage
Cluster 5	Large to mid-cap stocks
Cluster 6	Mid-cap stocks with lower leverage

7.2 Each Cluster Financial Properties

Additionally, the correlation table of clusters and the return and risk plot show greater diversification between clusters. The correlation table at the bottom is from the HW assignment, while the table at the top is from our model. Similarly, the return and risk plot on the left represents our model's results, whereas the plot on the right is from the HW assignment. The correlations between clusters in our model ranged from 0.1 to 0.2, which are significantly lower than the HW assignment results, which ranged from 0.2 to 0.4. Moreover, our optimal clusters exhibited more diverse mean returns and risks compared to the HW assignment results.

Table 11: Comparison of Correlation Tables and Return and Risk plot with HW Assignment

Correlation table of clusters	

	Cluster_1	Cluster_2	Cluster_3	Cluster_4	Cluster_5	Cluster_6
Cluster_1	0.206170	0.169396	0.199515	0.153944	0.221766	0.184044
Cluster_2		0.195575	0.116259	0.186751	0.160127	0.179416
Cluster_3			0.279008	0.104370	0.245787	0.153454
Cluster_4				0.190602	0.142932	0.161276
Cluster_5					0.248158	0.186413
Cluster_6						0.178608

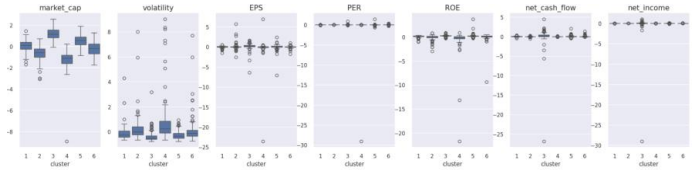
	Cluster_1	Cluster_2	Cluster_3	Cluster_4	Cluster_5
Cluster_1	0.255259	0.273909	0.219487	0.128095	0.081560
Cluster_2		0.392159	0.309855	0.186202	0.125555
Cluster_3			0.254206	0.162585	0.115306
Cluster_4				0.179116	0.176731
Cluster_5					0.206887

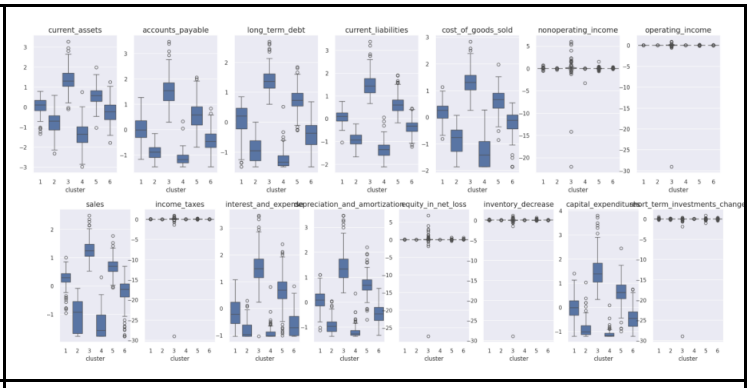
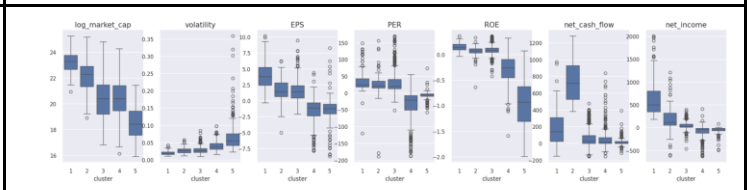
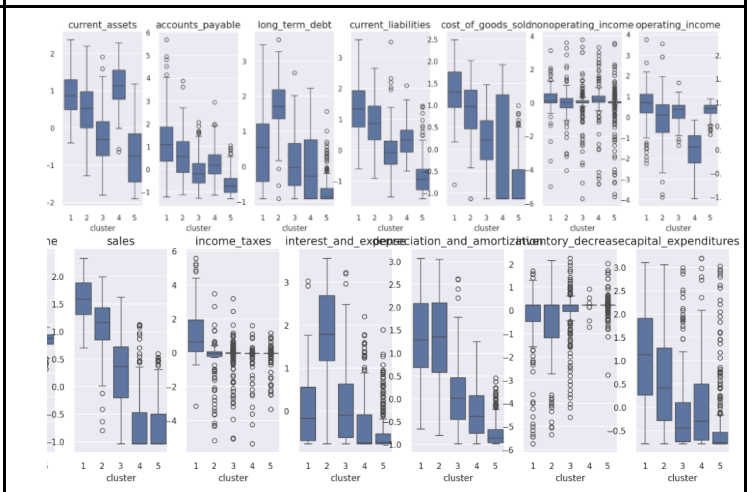
Return and risk characteristics	

7.3 Comparison with HW Assignment Result

We compared our optimal model (Exp 6 Model) with the results from the previous AI Practice HW assignment. As shown in Table 11, our optimal clusters demonstrate greater distinction between clusters. In the previous HW assignment, log_market_cap was the only feature that differed significantly among clusters, while other financial features exhibited similar values across clusters. Even when using PCA for dimension reduction, the clustering results displayed high variation within clusters and high similarities between clusters. However, the use of an autoencoder and t-SNE in our current model has led to better clustering performance, as indicated by both clear visualization and improved Silhouette and Davies-Bouldin scores.

Table 12: Box plot comparison between our model and HW model

Experiment 6	

	
AI Practice 2 (Original data)	
AI practice 2 extension PCA + K-means clustering (Extension data)	

8 Application

8.1 Fama-French Factor Model

Unlike the traditional Capital Asset Pricing Model (CAPM), which asserts that the returns of individual assets are influenced solely by market returns, the factor model proposed by Professors Eugene Fama and Kenneth French in 1992 argues that stock

returns are actually influenced by a wider range of internal and external factors. This factor model theory gained significant prominence, starting with the introduction of the 3-factor model, which incorporated size (SMB) and value (HML) factors in addition to the market factor. Subsequent research built upon this foundation, with Carhart's model adding momentum as a factor to create the 4-factor model. Professors Fama and French further expanded their 3-factor model by including profitability (RMW) and investment (CMA) factors to introduce the 5-factor model. Beyond these, numerous other factors are being researched by investors and scholars to predict the price movements of individual assets in the market. We aim to explore how our clustering experiment results can be integrated into these factor models.

8.2 Application to Fama-French Factor Model

Deep learning technology is particularly useful for clustering because deep learning excels in identifying complex, non-linear patterns in data that traditional statistical methods might miss. The complexity of financial markets is very high, and especially in the case of traditional theories, their signals can fade and disappear as they are applied to the market. In such situations, using deep learning, which is useful for handling high-dimensional financial data, to discover new factors and signals has significant financial engineering implications.

As mentioned above, various traditional factor models, including the Fama-French factor model, are based on predefined economic theories. By applying deep clustering in various ways to new types of data, it is possible to derive entirely new factors that are not based on traditional economic theories. This experiment aims to demonstrate the validity of this methodology by finding new groupings in traditional financial data.

Additionally, deep learning techniques, including deep clustering, can provide foundational theories that can respond more quickly to financial markets that are constantly and unpredictably changing, compared to traditional theories. One significant advantage is the ability to quickly detect meaningful factors from data patterns that are subtly and constantly evolving. In practice, proprietary trading firms sometimes use factor timing techniques to capture factors that work well at specific times. Deep learning could also be used as a practical tool for such factor timing.

From the clustering results, distinct clusters are based on the scale of several financial factors, especially from market cap, current assets, accounts payable, long-term debt, current liabilities, and cost of goods sold. Considering the characteristics of each factor, we can propose three potential additional factors in the Fama-French model: **liquidity, leverage, operational efficiency**.

First, the clustering based on **current assets and accounts payable** suggests that liquidity is a significant distinguishing factor. Liquidity indicates how easily a company can meet its short-term obligations, impacting its financial stability and risk profile.

Firms with higher liquidity are generally seen as less risky, which could influence their returns.

$$R_i - R_f = \alpha_i + \beta_{MKT} \cdot (R_M - R_f) + \beta_{SMB} \cdot SMB + \beta_{HML} \cdot HML + \beta_{LIQ} \cdot LIQ + \epsilon_i$$

Second, the differentiation based on **long-term debt and current liabilities** points to leverage as a potential factor. Leverage measures the extent of a company's financing through debt, which can affect its risk and return characteristics. Higher leverage implies higher risk due to the obligation to meet debt repayments, potentially leading to higher returns demanded by investors.

$$R_i - R_f = \alpha_i + \beta_{MKT} \cdot (R_M - R_f) + \beta_{SMB} \cdot SMB + \beta_{HML} \cdot HML + \beta_{LEV} \cdot LEV + \epsilon_i$$

Lastly, the distinct clusters based on **cost of goods sold** suggest operational efficiency as a factor. Cost of goods sold is a direct measure of a company's production efficiency and cost management. Companies with lower costs relative to their sales are more profitable and efficient, which can be a critical factor influencing their stock returns.

$$R_i - R_f = \alpha_i + \beta_{MKT} \cdot (R_M - R_f) + \beta_{SMB} \cdot SMB + \beta_{HML} \cdot HML + \beta_{OE} \cdot OE + \epsilon_i$$

8.3 Robustness check on newly defined factors via regression analysis

When defining these new factors, it is crucial to determine how well the new factors explain the returns of individual assets. This is because we can verify how factor extraction based on our deep clustering results applies to the actual financial market. To validate the appropriateness of the factors, we need to perform regression analysis using the formula of the Fama-French model presented above with the actual time-series returns of each portfolio and the risk-free rate.

The required data are as follows:
- Returns of individual assets
- Market returns
- Risk-free rate (e.g., US 3-year Treasury Bill, T-Bill)
- Returns of small-cap and large-cap portfolios (SMB)
- Returns of high-value and low-value portfolios (HML)
- Portfolio returns based on the newly defined factors (NF)

Based on the above data, we calculate the excess returns of individual assets and the market premium, and then define the following formula.

$$R_i - R_f = \alpha_i + \beta_{MKT} \cdot (R_M - R_f) + \beta_{SMB} \cdot SMB + \beta_{HML} \cdot HML + \beta_{new\_factor} \cdot NF + \epsilon_i$$

By performing regression analysis on the above formula, we can estimate the coefficients (betas) and statistical metrics. For instance, we can use Python's 'statsmodels' library. Using this library, we can estimate each coefficient and determine the model's fit through the R squared value. For example, if the R squared value is 0.85, it means that the model explains 85% of the variation in the returns of this asset. Additionally, this library provides statistical indicators such as t-statistics and p-values, which allow us to check the statistical significance.

However, in this experiment, since each fundamental data was provided only for a single day, we could not calculate the daily returns of the portfolios classified by these factors, which limited the regression analysis.

9 Limitation

In our research, several limitations were identified. First, the presence of outliers in PER and ROE box plots distorted the scale, making it difficult to compare how well each feature was clustered. To address this, it may be necessary to consider a process of removing outliers, as clustering results could change after adequately trimming these outliers. Also, the data used was limited; despite using data from around 2,000 different stocks, the evaluation period was restricted. Thus, it is essential to conduct the same experiments over different time periods to test whether the formation of clusters and the derived factors remain consistent.

Second, the dataset could be expanded to include additional relevant factors such as PBR, which were not part of the current analysis. Including these additional firm-specific data points could provide a more comprehensive view. Additionally, incorporating external factors like GNR or GDP of the respective time periods might impact the classification results. Adding these extra datasets could further improve the robustness and accuracy of the clustering analysis.

Another limitation of our research involves the hyperparameter tuning of the autoencoder. The layer depth and latent space size of the autoencoder could significantly influence the clustering results. Our study might have benefited from a more thorough exploration of these parameters, as different configurations could yield different outcomes. Additionally, the choice of optimizer plays a crucial role in training the autoencoder. By experimenting with various optimizers and fine-tuning the hyperparameters, we might have achieved more optimal clustering results, potentially improving the overall performance and accuracy of our model.

10 Conclusion

After conducting eight different experiments to perform deep clustering using various datasets and dimensionality reduction techniques, the analysis of clustering results led to a reasonable conclusion: the application of autoencoders significantly improved the clustering of stocks using various financial properties. This demonstrates the power of advanced machine learning techniques in uncovering hidden patterns in complex financial datasets. Furthermore, the experiments provided another significant finding regarding price data. Contrary to initial expectations, the use of price data alongside financial data as input did not enhance clustering results. This insight highlights the necessity of carefully selecting relevant data features when applying AI in financial contexts, as not all data types contribute equally to model performance. Lastly, and most importantly, the analysis of each feature from the obtained clusters and the application to the traditional factor model revealed an interesting result: the characteristics of the clusters are distinguished based on three notable factors—liquidity, leverage, and operational efficiency, in addition to the traditional size factor. This finding underscores the potential of AI to identify key financial indicators that are crucial for risk management and investment strategies.

The most important lesson from this experiment is that the application of AI in finance, such as using sophisticated algorithms like autoencoders can lead to more accurate and insightful analysis, ultimately resulting in better decision-making processes. Although some AI algorithms may not fit several finance related problems, the application of AI, particularly deep-learning and clustering techniques, holds transformative potential for the finance sector if reasonable financial knowledge is combined in the analysis.

References

1. Bini, B. S., & Mathew, T. (2016). Clustering and regression techniques for stock prediction. *Procedia Technology*, *24*, 1248-1255.
2. Hwang, Y., Lee, Y., & Fabozzi, F. J. (2023). Identifying household finance heterogeneity via deep clustering. *Annals of Operations Research*, *325*(2), 1255-1289.
3. Pattarin, F., Paterlini, S., & Minerva, T. (2004). Clustering financial time series: an application to mutual funds style analysis. *Computational Statistics & Data Analysis*, *47*(2), 353-372.