Clustering and Deep Learning in Portfolio Construction

Harvard University - Extension School
CS-82 Advanced Machine Learning
Nitesh Kurma - Janice Pham

I. Introduction

Equity market is one of the most active markets in the financial world and draws a lot of attention and research from both academics and industry experts. Most of the work and research have been focused on large cap stocks which typically already reflect fair market values thanks to the market efficiency. This market efficiency partially comes from active trading activities by financial market participants. In addition, the amount of information and news coverage for these large cap stocks is huge compared to mid and small cap counterparts which helps facilitate market efficiency. However, the market is never in a perfect efficiency, i.e., the stock price does not reflect all the company's public and private information, hence, leaves room for profit-making potential.

Recent developments in machine learning have led to more novel methods that could be applied in finance. This project aims to explore both supervised and unsupervised machine learning techniques with the application in portfolio construction. On the unsupervised side, we applied various clustering methods including, but not limited to, K-Means, Hierarchical Clustering, Gaussian Mixture Model (GMM) using quarterly data of more than 270 large and midcap stocks from 2002 to 2022. On the supervised side, we implemented deep learning based on LSTM model on the same dataset to predict quarterly returns, rank stocks, and construct long-short portfolio based on these rankings.

The two main goals are as follows:

- Explore if clustering methods can be used as a feasible stock selection method in the portfolio construction.
- Explore if LSTM model can be used together with clustering method to enhance the portfolio construction strategy

- Explore visualization in clustering to see how well our clustering algorithm is doing.

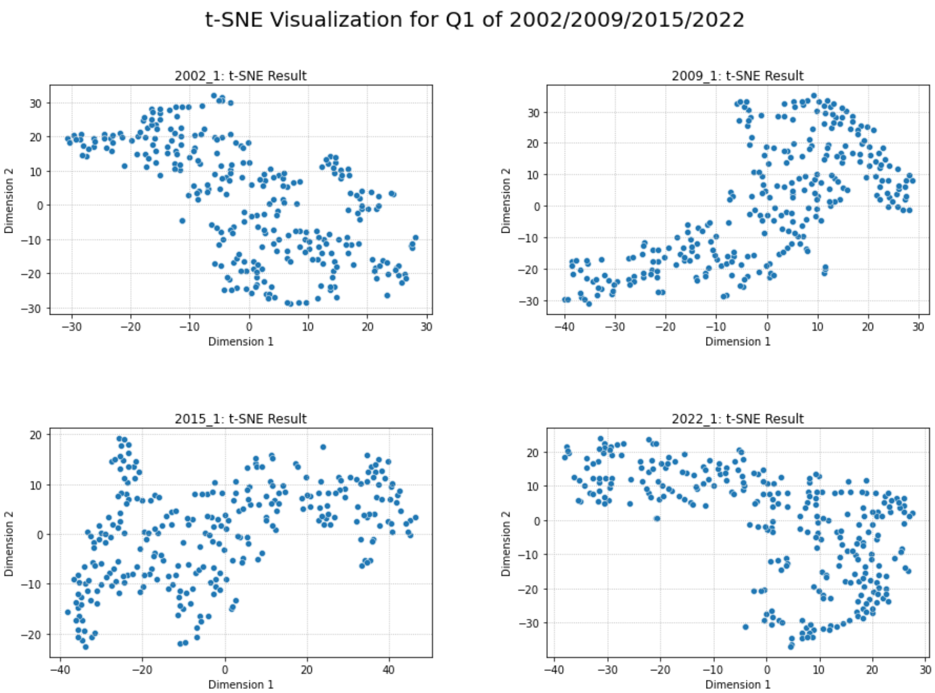
II. Data Collection, Exploration, and Cleaning

1. Data Collection

We used tiingo.com API to retrieve 1342 tickers from both large cap and mid cap companies from 2002 to 2020. However, since a lot of companies did not exist at that time, the number of stocks in the list was trimmed down to match this criterion. Furthermore, because some companies did not report the same statistics as other companies, we experienced a lot of missing values for these companies. Therefore, we excluded them from our stock selection. The final list contains 293 stocks in both large and midcap categories (classified based on their capitalizations as of December 2022) across how many features. We looked at each features, and found missing values, and each feature. We also found few stocks that did not have completed fundamental data. Our final stock universe was 278 stocks with both fundamental and daily data.

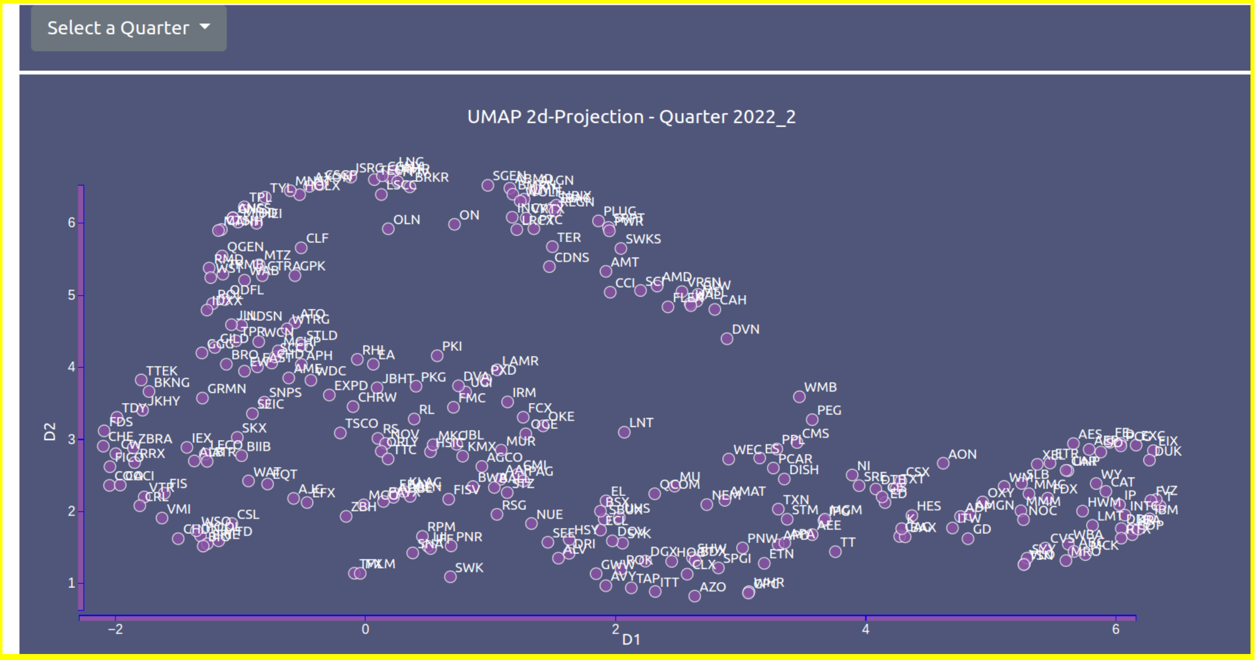
2. Exploratory Data Analysis (EDA)

The collected dataset contains 48 features from income statements and balance sheets across 278 stocks. We removed overlapped features based on our domain knowledge and based on the data cleaning process of missing values. The final dataset includes 17 features across 278 stocks. The principal component analysis across the stocks for each feature indicated that many stocks tend to move together (see notebook for detail). In addition, t-SNE visualization also suggested that there be no real clusters among these stocks or features. However, this is difficult to confirm because it only provides the visualization for two dimensions.



t-SNE Visualization for Q1 of 2002/2009/2015/2022

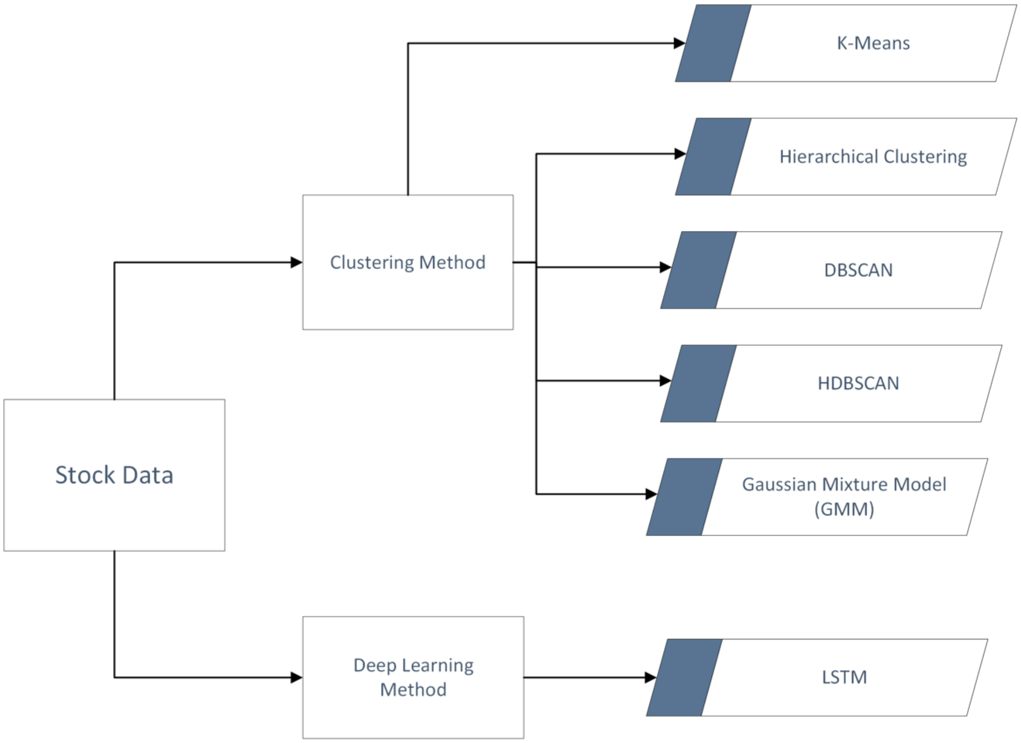
UMAP worked the best for our visualization of clusters, and well as looking at 278 stocks. We created a visualization tool for our EDA and for verification of our clustering methods. Here is a visualization of all 278 stocks using UMAP for 2022 Quarter 2.



UMAP 2d-Projection - Quarter 2022_2

III. Methodology

Following is the structure of our modeling process:



1. Clustering method

We performed 4 different clustering methods including K-Means, Hierarchical Clustering, DBScan, and Gaussian Mixture Model (GMM)

The modeling strategy is as follows for each clustering method:

- Step 1: We use the first quarter to perform the respective clustering method and fine-tune the model based on knee plot or Silhouette score plot (K-Means, Hierarchical Clustering, DBSCAN, HDBSCAN) and BIC ,AIC and Log Likelihood score plot (GMM). Based on the result of the fine-tuned model, we obtain the optimal number of cluster that we will apply for later quarters. Note that we do not fine-tune subsequent quarters but only use the optimal number of clusters and apply it.

- Step 2: For each subsequent quarter, we perform clustering technique on a set of 17 features and 278 stocks. The result of this process is a set of different clusters per quarter. Each cluster contains different stocks and may or may not have the same number of stocks compared to other clusters.

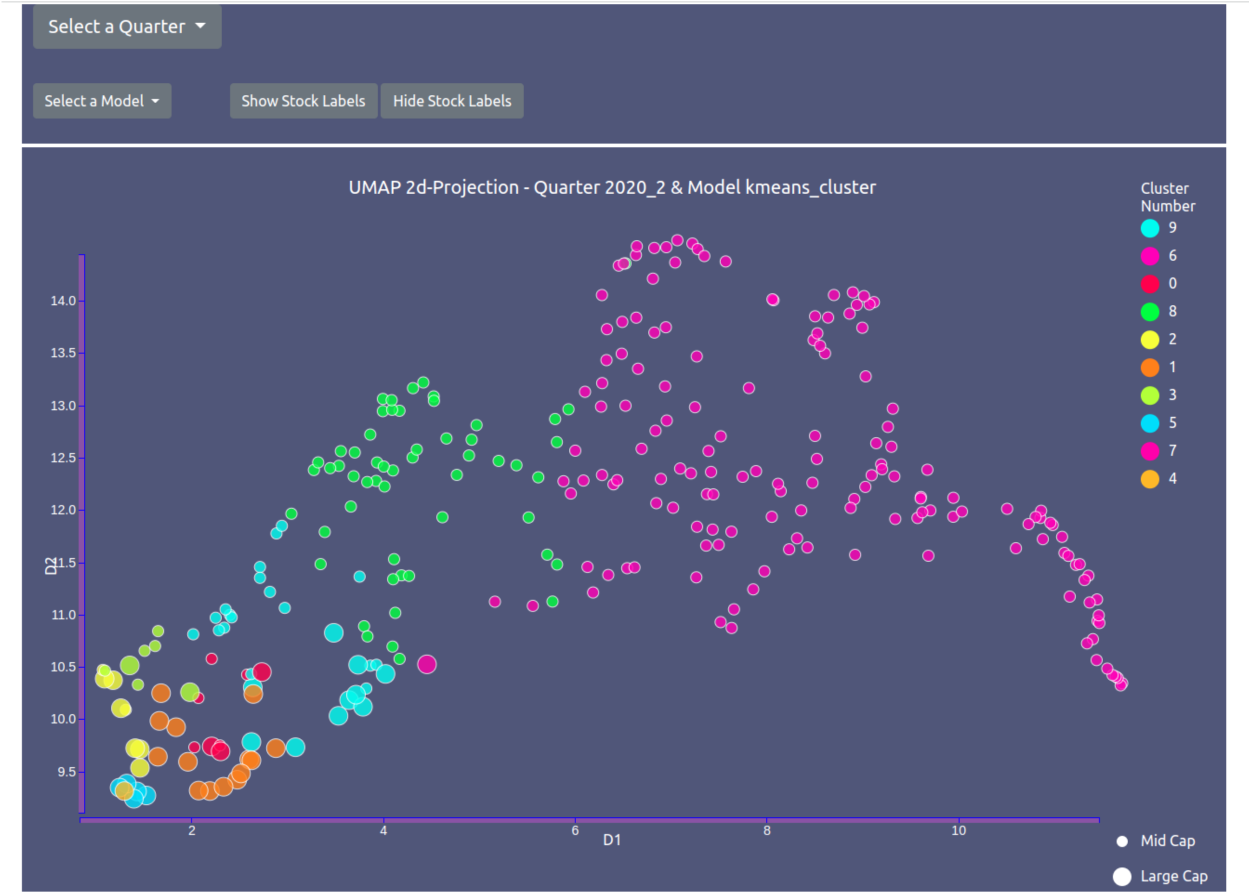
- Step 3: We compute the quarterly return of each cluster based on the quarterly stock returns and equal weighting scheme. We also compute the cluster's volatility (or standard deviation) based on the volatility of its stocks. Using return and volatility (assuming risk-free rate of zero), we can compute the Sharpe ratio of each cluster.

- Step 4: Based on the Sharpe ratios obtained for all clusters of each quarter, we rank the clustered portfolio and select the top 3 clusters with the highest Sharpe ratios per quarter. We repeat this process to obtain the cluster portfolio rankings across all quarters.

- Step 5: For each quarter, we compare the top 3 clustered portfolio's Sharpe ratios with that of the benchmark S&P 500. We also compare and count how many times the top performing clustered portfolio outperforms S&P 500.

This procedure is repeated for all clustering methods under consideration to obtain the sets of clustered portfolio's performances, compare and evaluate the effectiveness of each clustering method in constructing a clustered portfolio.

2. **Visualization :** We looked at different visualization techniques for visualizing our stocks, and how each of the clustering algorithms are clustering our data. However, We had a large universe of stocks, and there were 83 quarters. We created a visualization tool in d3.js to visualize our data, and clusters across all features in 2D-Projection using UMAP. Our visualization tool can be accessed at https://dashing-crisp-38c329.netlify.app/ . Here is an example screenshot for Covid period quarter in 2022 Quarter 2.



3. Deep learning method

The LSTM modeling procedure is as follows:

- <u>Step 1:</u> We prepare the data structure for LSTM model by scaling and reshaping to match with LSTM model architecture (see the notebook for details). In addition, we also split the data into train and test datasets. Since the goal is to predict next quarter stock return, the data is shifted by 1 period. In other words, 2002 Q1 features are used to predict 2002 Q2 stock return.

- <u>Step 2:</u> We use LSTM to train and predict the quarterly stock returns for each stock in the dataset. The result is a set of all stocks' quarterly returns (across 278 stocks) from 2002 to 2022. The LSTM model architecture includes 2 layers of LSTM with the following hyperparameters:

Learning rate	0.01
Hidden size	64
Dropout rate	0.3
Epoch	50
Batch size	32
Patience (for Early Stopping)	5

We do not go deeper in terms of the model architecture or hyperparameters because with the limited amount of quarterly data we have, this would risk overfitting.

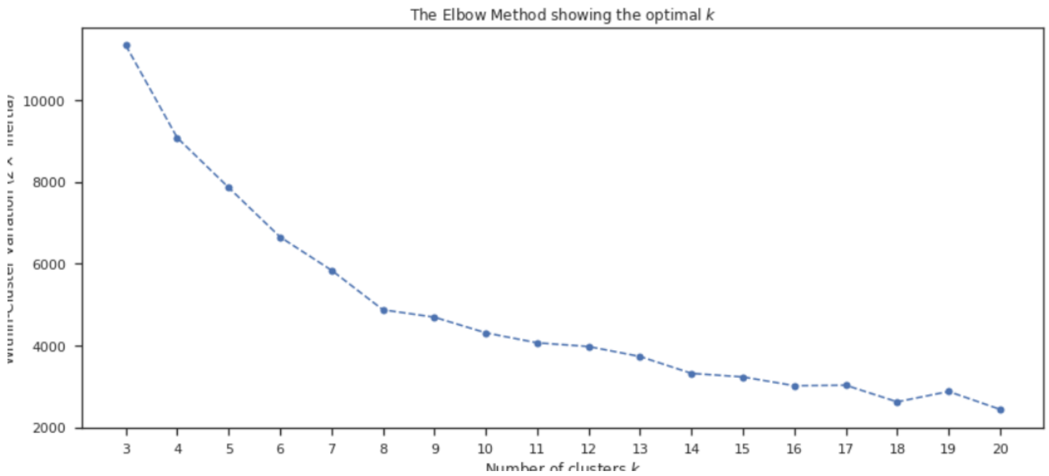
- Step 3: We rank the stocks based on the quarterly returns and select the top n’th stocks to long and the bottom n’th stocks to short. This provides us with the long-short portfolio for each quarter.

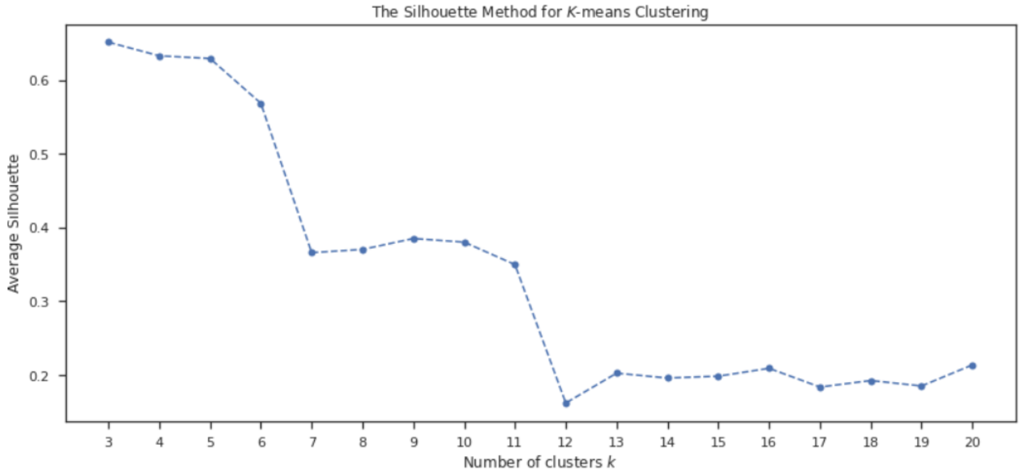
- Step 4: We compute the Sharpe ratio of the above long-short portfolio strategy obtained by the LSTM model and compare it with the benchmark S&P 500.

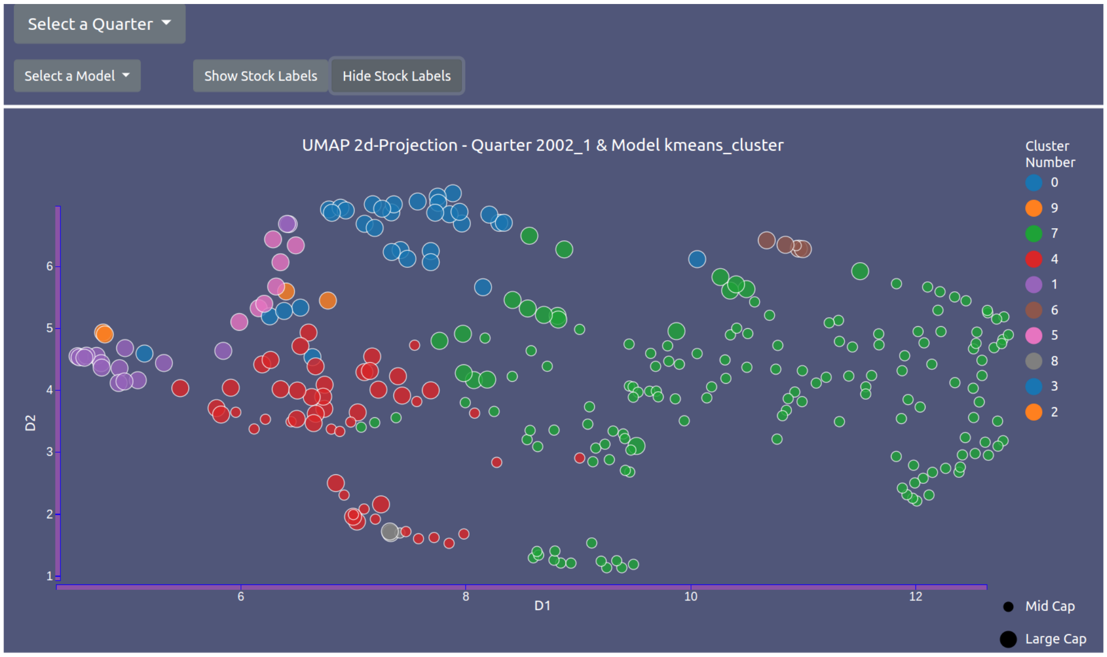
IV. Result

1. Clustering method

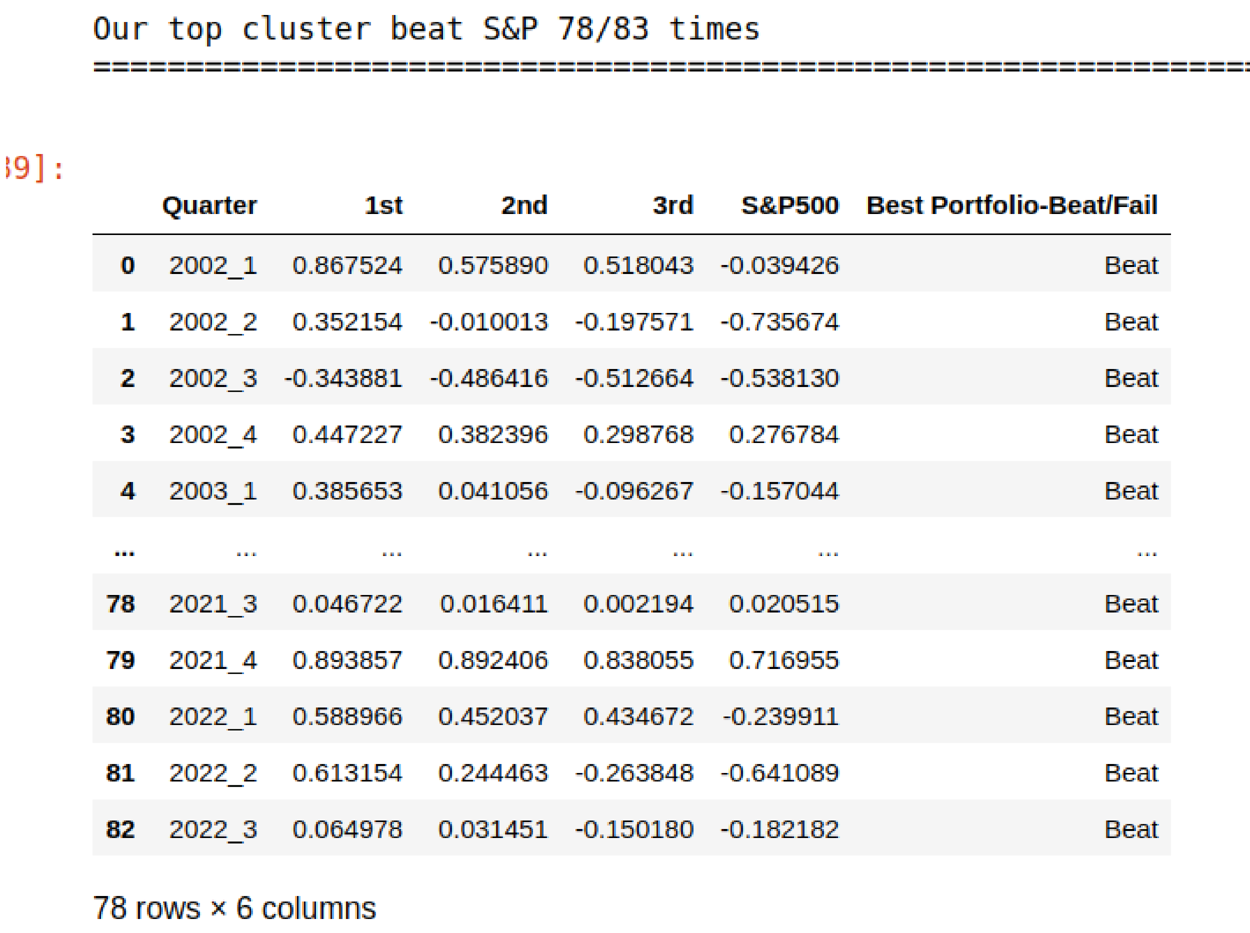
a. K-Means We trained Kmeans cluster from 3 cluster to 30 clusters. For model selection we used both elbow method of inertia, and silhouette method. Both gave us conflicting number of clusters. We tried in the range of 8-10 clusters, and inspected how well they were doing for our 2002 quarter 1 data. 10 cluster seemed appropriate. We also looked at the cluster through the tool we developed for visualization. We found that with 10 clusters stocks were properly grouped according to their financial performance for 2002 Quarter 1.


The Elbow Method showing the optimal k
10000
8000
6000
4000
2000
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Number of clusters k


The Silhouette Method for K-means Clustering
Average Silhouette
0.6
0.5
0.4
0.3
0.2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Number of clusters k


Select a Quarter
Select a Model
Show Stock Labels
Hide Stock Labels
UMAP 2d-Projection - Quarter 2002_1 & Model kmeans_cluster
D2
1
2
3
4
5
6
D1
6
8
10
12
Cluster Number
● 0
● 9
● 7
● 4
● 1
● 6
● 5
● 8
● 3
● 2
● Mid Cap
● Large Cap

The top 3 best performing clustered portfolios from the hierarchical clustering method appeared to work well. In fact, they outperformed the S&P500 most of the time and the best clustered model only failed in 5 periods out of 83 periods. During financial downturns, this top clustered portfolio’s loss was lessened while during economic boom, it performed much better.

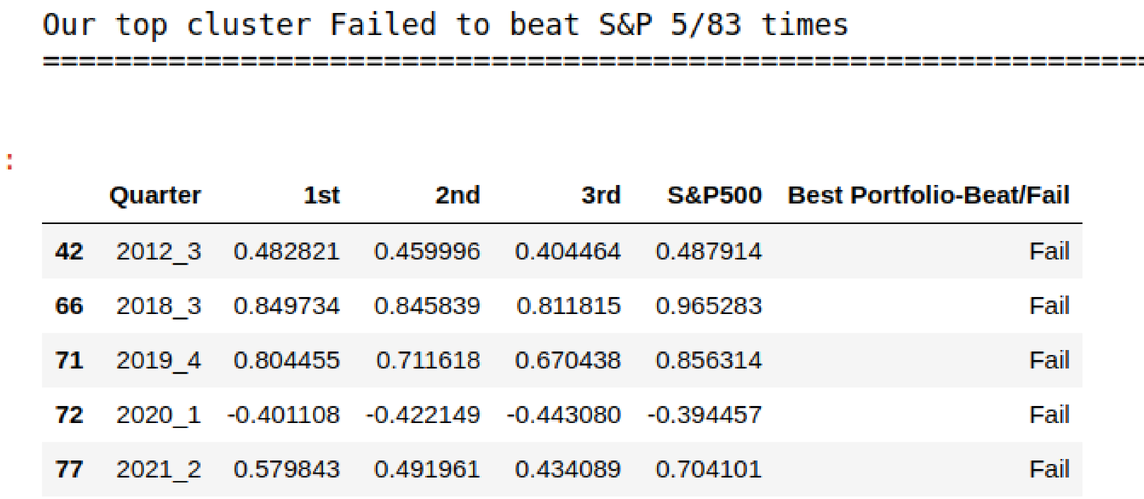


```
Our top cluster beat S&P 78/83 times
================================================================================
```

9]:

	Quarter	1st	2nd	3rd	S&P500	Best Portfolio-Beat/Fail
0	2002_1	0.867524	0.575890	0.518043	-0.039426	Beat
1	2002_2	0.352154	-0.010013	-0.197571	-0.735674	Beat
2	2002_3	-0.343881	-0.486416	-0.512664	-0.538130	Beat
3	2002_4	0.447227	0.382396	0.298768	0.276784	Beat
4	2003_1	0.385653	0.041056	-0.096267	-0.157044	Beat
...	...	...	...	...	...	...
78	2021_3	0.046722	0.016411	0.002194	0.020515	Beat
79	2021_4	0.893857	0.892406	0.838055	0.716955	Beat
80	2022_1	0.588966	0.452037	0.434672	-0.239911	Beat
81	2022_2	0.613154	0.244463	-0.263848	-0.641089	Beat
82	2022_3	0.064978	0.031451	-0.150180	-0.182182	Beat

78 rows × 6 columns



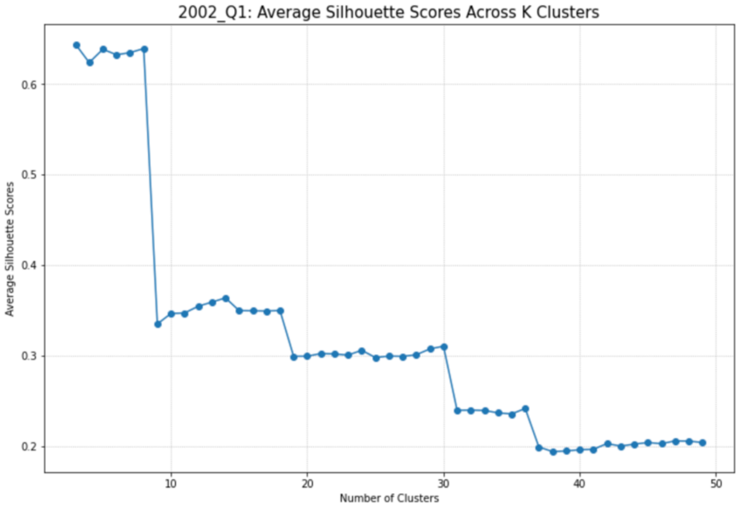
```
Our top cluster Failed to beat S&P 5/83 times
=============================================================
```

:

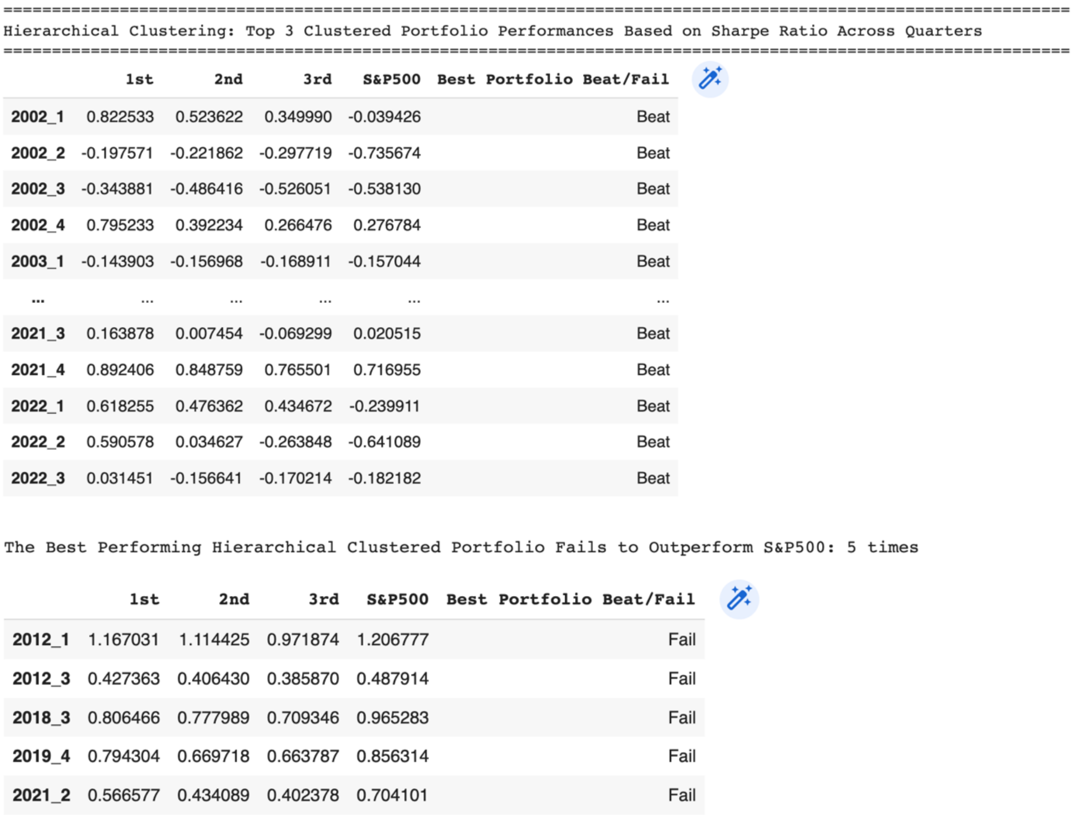
	Quarter	1st	2nd	3rd	S&P500	Best Portfolio-Beat/Fail
42	2012_3	0.482821	0.459996	0.404464	0.487914	Fail
66	2018_3	0.849734	0.845839	0.811815	0.965283	Fail
71	2019_4	0.804455	0.711618	0.670438	0.856314	Fail
72	2020_1	-0.401108	-0.422149	-0.443080	-0.394457	Fail
77	2021_2	0.579843	0.491961	0.434089	0.704101	Fail

b. Hierarchical clustering

We performed hierarchical clustering starting from 3 clusters up to 50 clusters with increment of 1 cluster on the first quarter. The average Silhoutte score plot suggested the optimal number of clusters is 8 because there is a big drop after this point and the Silhouette scores no longer improves beyond this point.



The top 3 best performing clustered portfolios from the hierarchical clustering method appeared to work well. In fact, they outperformed the S&P500 most of the time and the best clustered model only failed in 5 periods out of 83 periods. During financial downturns, this top clustered portfolio's loss was lessened while during economic boom, it performed much better.



```
====================================================================================================
Hierarchical Clustering: Top 3 Clustered Portfolio Performances Based on Sharpe Ratio Across Quarters
====================================================================================================
```

	1st	2nd	3rd	S&P500	Best Portfolio Beat/Fail
2002_1	0.822533	0.523622	0.349990	-0.039426	Beat
2002_2	-0.197571	-0.221862	-0.297719	-0.735674	Beat
2002_3	-0.343881	-0.486416	-0.526051	-0.538130	Beat
2002_4	0.795233	0.392234	0.266476	0.276784	Beat
2003_1	-0.143903	-0.156968	-0.168911	-0.157044	Beat
...	...	...	...	...	...
2021_3	0.163878	0.007454	-0.069299	0.020515	Beat
2021_4	0.892406	0.848759	0.765501	0.716955	Beat
2022_1	0.618255	0.476362	0.434672	-0.239911	Beat
2022_2	0.590578	0.034627	-0.263848	-0.641089	Beat
2022_3	0.031451	-0.156641	-0.170214	-0.182182	Beat

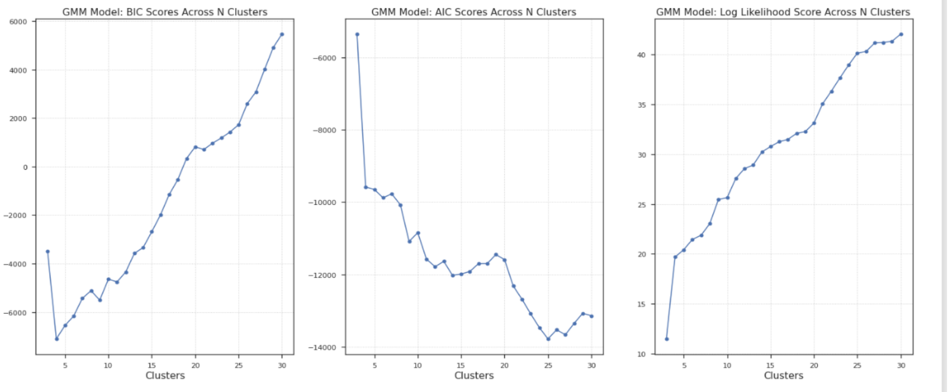
```
The Best Performing Hierarchical Clustered Portfolio Fails to Outperform S&P500: 5 times
```

	1st	2nd	3rd	S&P500	Best Portfolio Beat/Fail
2012_1	1.167031	1.114425	0.971874	1.206777	Fail
2012_3	0.427363	0.406430	0.385870	0.487914	Fail
2018_3	0.806466	0.777989	0.709346	0.965283	Fail
2019_4	0.794304	0.669718	0.663787	0.856314	Fail
2021_2	0.566577	0.434089	0.402378	0.704101	Fail

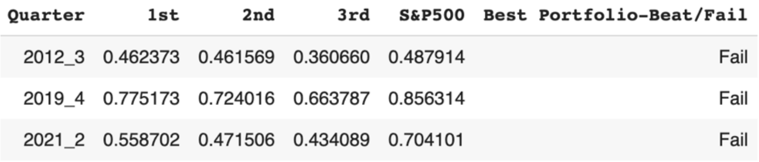
c. DBSCAN Unlike K-Means and hierarchical clustering, DBSCAN did not work for our data and prediction goal. DBSCAN resulted in 1 cluster across all values of epsilon and min_samples. As we see from the visualization, this seems to make sense as these clustering algorithms methods are based on density and the t-SNE & UMAP suggested that the data points are quite close to one another (high density). Therefore, we cannot use this method to group the stocks and construct the clustered portfolio.

d. Gaussian mixture model (GMM)

The last clustering model we tested was GMM and we used 3 metrics to select the optimal number of clusters: BIC, AIC, and the likelihood score obtained from sklearn GMM model. We selected n_cluster = 9 based on the AIC/BIC plot and when we observed the GMM likelihood scores, we added another optimal n_cluster = 12 for portfolio construction and comparison with the benchmark. Based on the amount of data, any number between 7 and 15 seems to be a good candidate for the optimal number of clusters. As the number of clusters increases, BIC starts to go up, particularly after n_cluster = 15. Both models performed well and similar to K-Means and hierarchical clustering, it only failed less the 10 times out of the entire period.

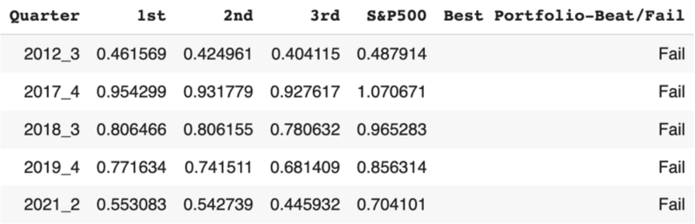


GMM-9



Quarter	1st	2nd	3rd	S&P500	Best Portfolio-Beat/Fail
2012_3	0.462373	0.461569	0.360660	0.487914	Fail
2019_4	0.775173	0.724016	0.663787	0.856314	Fail
2021_2	0.558702	0.471506	0.434089	0.704101	Fail

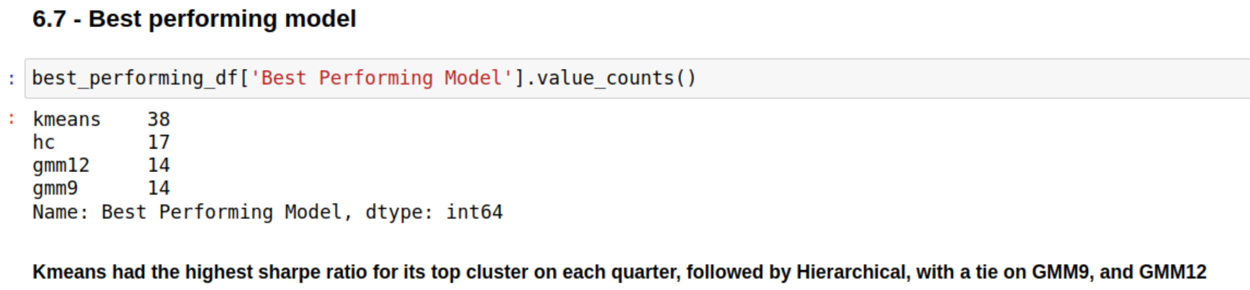
GMM-12



Quarter	1st	2nd	3rd	S&P500	Best Portfolio-Beat/Fail
2012_3	0.461569	0.424961	0.404115	0.487914	Fail
2017_4	0.954299	0.931779	0.927617	1.070671	Fail
2018_3	0.806466	0.806155	0.780632	0.965283	Fail
2019_4	0.771634	0.741511	0.681409	0.856314	Fail
2021_2	0.553083	0.542739	0.445932	0.704101	Fail

e. Overall model comparison

Comparing across all clustering methods, K-Means is the best performing model since out of all periods, K-Means' top clustered portfolio achieved the highest number of times that it outperformed the other models. This is followed by hierarchical clustering and GMM.



6.7 - Best performing model

```
: best_performing_df['Best Performing Model'].value_counts()
```

```
: kmeans    38
  hc        17
  gmm12     14
  gmm9      14
  Name: Best Performing Model, dtype: int64
```

Kmeans had the highest sharpe ratio for its top cluster on each quarter, followed by Hierarchical, with a tie on GMM9, and GMM12

2. **Deep learning method (LSTM)**

Clustering method is more about grouping stocks that can take advantage of the synergy and correlations among the stocks. This synergy and correlations generate values that when these stocks are grouped together, they can achieve a better performance. The clustering method is not meant to predict the stock returns but rather a test to see which stocks should go together.

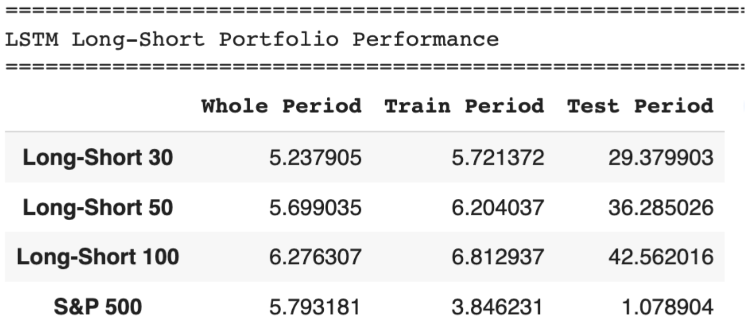
On the other hand, LSTM can be used to predict stock returns thanks to the nature of its architecture that makes it suitable for time series data. This in turns can be used as a subsequent process to the clustering step above to select and predict the stock components of the best performing cluster.

The most heavy-lifting work in our LSTM modeling is the data pre-processing process. Since we are interested in predicting the stock's return in the next period based on the current period's features, we firstly need to shift our response variable by a period and shorten our feature period by 1 period. In other words, 2022 Q1 features will be used to predict stock's returns in 2022 Q2. Moreover, we perform scaling and reshape the data to match with the model architecture. As we have 278 stocks and 17 features for 82 quarters (after shortening by 1 quarter for predicting purpose), we perform LSTM model separately on each stock to obtain the predicted stock returns.

To construct the portfolio, we rank these stocks based on the quarterly returns to decide the long and short position. There are 3 long-short portfolios that we explore: the long-short of top 30/50/100 and bottom 30/50/100 stocks. Similar to the clustering method, we evaluate the portfolio performance based on Sharpe ratio and also compare it with the benchmark. In addition, unlike unsupervised learning like clustering, since we split the data into train and test datasets, we can compute the Sharpe ratio for train, test, and whole period. Because the entire dataset contains more than 200 stocks, we cannot evaluate the individual stock's LSTM model performance and in fact this is not correct either as the goal is to construct and evaluate the whole portfolio. Therefore, we do not evaluate a single stock performance. In addition, a back test is a conventional way in finance to evaluate the model performance.

Based on the following table, the long-short portfolio's annualized Sharpe ratios for all three positions all outperformed that of S&P 500. It is interesting to see the Sharpe ratio in the test period is apparently higher than the train period, suggesting the model seems to generalize well. However, this very high Sharpe ratio in the test period also raises concern because it could be due to some special events in the period under consideration that has led to this abnormally high Sharpe ratio. Another reason could be the model especially only works well for this specific period and if we consider a different time range, it may become unstable and cannot generate similar results.

Moreover, we also notice here the effect of diversification: as we add more stocks, Sharpe ratio still increases but the increase appears to be at diminishing returns. With additional stock, one can reduce the risk; however, this also means the high return that is usually associated with high risk also balances with other low risk low return stocks. This is the trade-off between risk and return in portfolio construction.



```
============================================================
LSTM Long-Short Portfolio Performance
============================================================
```

	Whole Period	Train Period	Test Period
Long-Short 30	5.237905	5.721372	29.379903
Long-Short 50	5.699035	6.204037	36.285026
Long-Short 100	6.276307	6.812937	42.562016
S&P 500	5.793181	3.846231	1.078904

V. Conclusion

This project explores a novel way to construct a portfolio based on both supervised and unsupervised machine learning and deep learning methods. With respect to unsupervised methods, aside from the density-based clustering method DBSCAN, all the methods we have tested (K-Means, Hierarchical Clustering, GMM) outperform the benchmark S&P 500 most of the time. This suggests that clustering can be used as part of the stock selection process. While there is no ground truth when it comes to unsupervised learning, the clustered portfolio's performance is already an indication to the success of the model. The clustered portfolio's stock components may change with each run, this is both an advantage and disadvantage, i.e., we can identify new stocks that could contribute to the portfolio performance, but we cannot replicate the result by including the similar stocks in the cluster in the next period. Therefore, the clustering method serves as a tool to learn the interactions among the stocks and why some stocks tend to move together or against one another. On the other hand, LSTM targets predicting the stock returns which can help support the result from clustering step. In other words, one can use the stocks selected by the top performing cluster and apply LSTM to predict the returns and see if the clustered portfolio could achieve persistent and consistent performance over the time.