1. **Introduction**

The US equity market is a place where investors all over the world turn to invest their capital. The objective is simple; make money. In almost all instances, this means buying a security for one price, and selling it for a higher price at a later date. The golden ticket to making money in the markets is to buy low, and sell high. However, timing these swings is exceedingly difficult even with ML/AI helpers. Instead of trying to time the market, another investing philosophy is that over the longer term, well run companies do well. To identify well-run companies, one can turn to company fundamentals that can be pulled off of balance sheets, income statements, and earnings reports. If fundamental indicators are related to returns in the US equity market, unsupervised clustering of these indicators should yield clusters of companies that perform better or worse than other clusters.

1. **Data**

The primary data source for this project comes from the AlphaVantage python API. Because the free version of the API is limited to 25 calls per day, I opted to pay $25 for a higher tier so I could download my data all in one go. Even with the paid tier, I was running into throttling restrictions, so I had to add a one second sleep component in the loop I used to download all the data.

*2.1 Data Cleaning/Imputation/EDA*

The data returned from AlphaVantage was far from complete. Upon inspection, there were quite a few missing data points across various features with varying degrees of missingness.

*Figure 1: Count of missing features*



The missing data points were then filled in using the ‘yfinance’ python library. This free library is similar to AlphaVantage, except it does not include as many features, and the features are even more likely to be missing. The features that were available were either imputed programmatically, or by hand. If a feature was unavailable from both data sources, this feature was dropped from the analysis. After all data was either imputed, or dropped, the data set is left with 20 different features to cluster on.

*Figure 2: Data description*

RangeIndex: 500 entries, 0 to 499

Data columns (total 22 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Unnamed: 0 500 non-null int64

1 Symbol 500 non-null object

2 MarketCapitalization 500 non-null int64

3 BookValue 500 non-null float64

4 EPS 500 non-null float64

5 GrossProfitTTM 500 non-null int64

6 DilutedEPSTTM 500 non-null float64

7 QuarterlyEarningsGrowthYOY 500 non-null float64

8 QuarterlyRevenueGrowthYOY 500 non-null float64

9 Beta 500 non-null float64

10 fiscalDateEnding 500 non-null object

11 totalRevenue 500 non-null float64

12 operatingIncome 500 non-null int64

13 operatingExpenses 500 non-null float64

14 netIncome 500 non-null int64

15 totalAssets 500 non-null int64

16 totalLiabilities 500 non-null int64

17 totalShareholderEquity 500 non-null int64

18 DividendPerShare 500 non-null float64

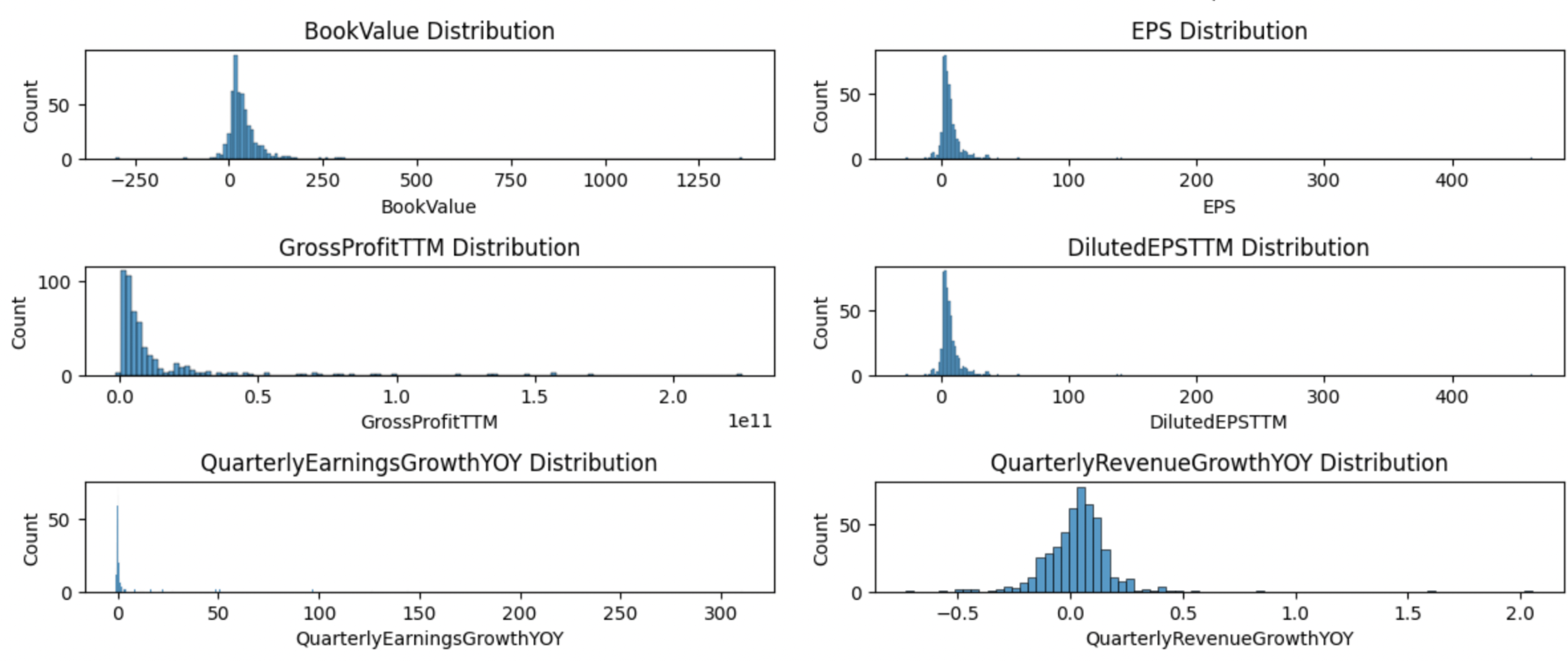
19 DividendYield 500 non-null float64

20 PEGRatio 500 non-null float64

21 currentDebt 500 non-null float64

Checking the distribution of the variables shows that there is ample variation to exploit during clustering. This is important because the variation is what will allow different equities to be assigned to different clusters. If there were no variation, all equities would be assigned to the same cluster.

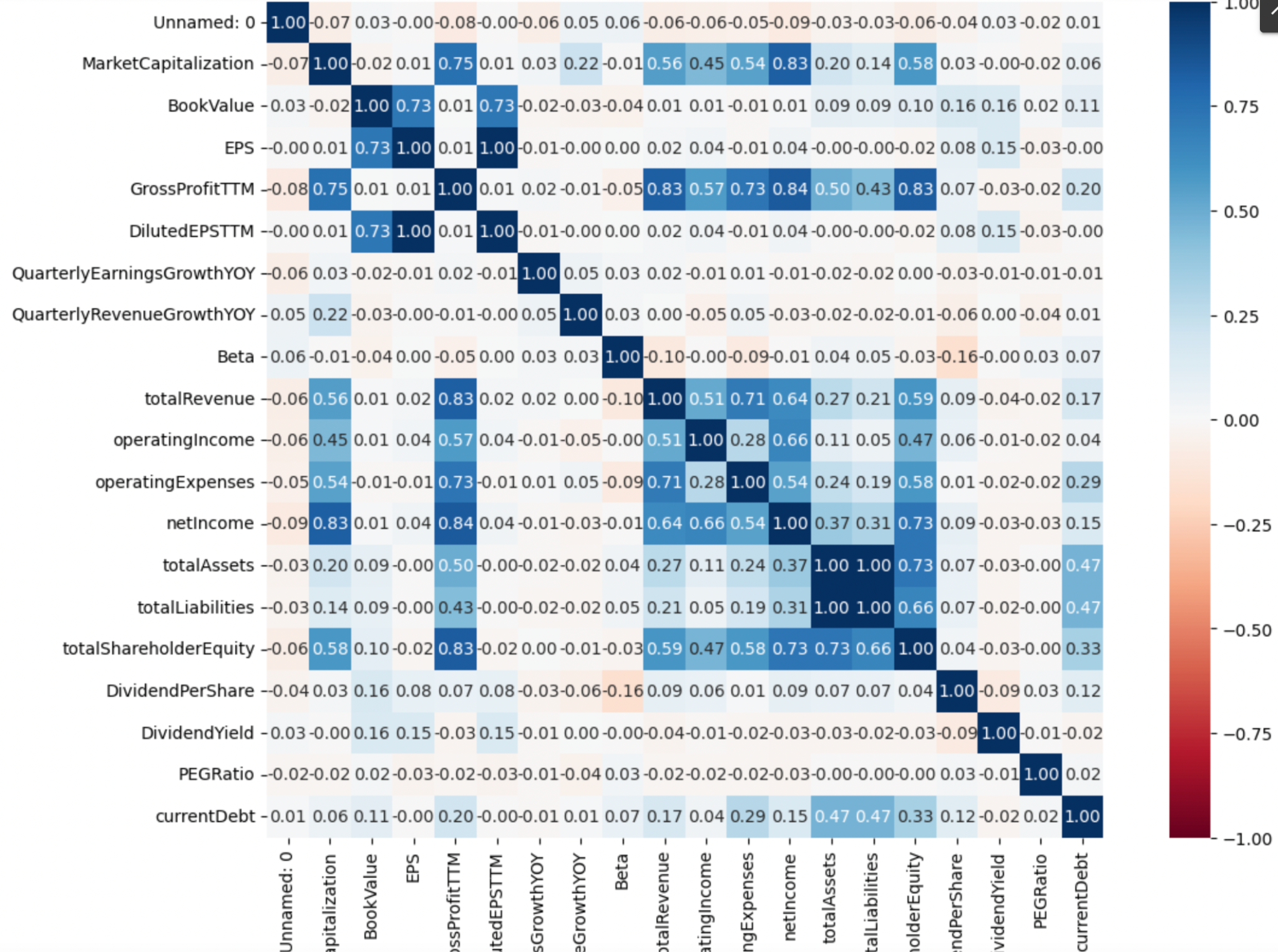
*Figure 3: Distributions of selected data points*



Although unsupervised clustering algorithms do not make the same assumptions about collinearity as some of the more familiar linear models, it is still a good idea to check the correlations between variables to see if there are any redundancies in the data. If two variables are highly correlated, the information they provide would essentially be counted twice, this is especially true after normalization, and the clusters might not split appropriately on other

features.

*Figure 4: Correlation matrix*



Overall, there are not too many strong correlations present in the data (only a couple of points over .8), and most points seem to have almost zero correlation with the exception of the financial features. Because there might be slight redundancies in the data, it will be interesting to see how reducing the dimensionality of the features with something like PCA might change the clusters.

*2.2 Data pre processing*

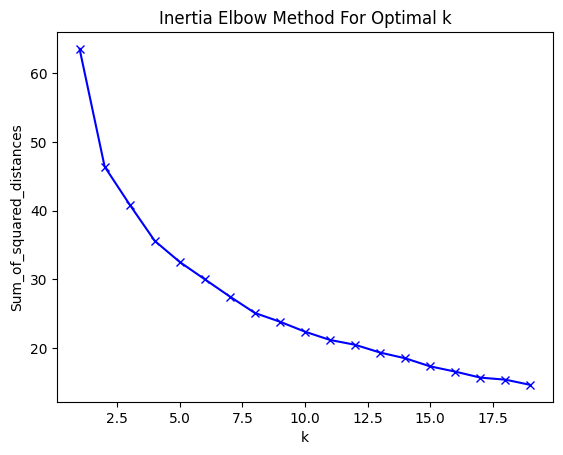
The only pre-processing step being taken at this time is to scale the data using a MinMax scaler. This will bound the data between 0 and 1, and ensure that features on a larger scale are not considered more important than features on a smaller scale. The scaling of the data is a crucial step in all distance based algorithms. To understand why this is the case, consider the formula for euclidean distance:

Now assume X ranges from (0,10) and Y ranges from (0,100). If X2 = 5 and X1 = 10, then X1 is five units away from X2, but twice as large. If Y2 = 25, and Y1 = 30 then Y1 is still five units away from Y2, but it is not twice as large. Paradoxes like this can cause clustering models to fail.

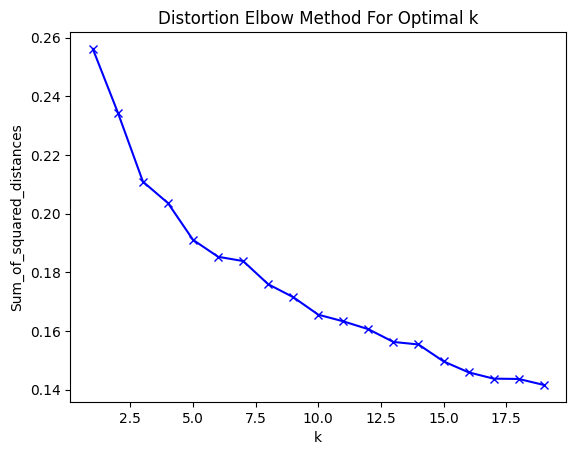
1. **Choosing the optimal number of clusters**

One of the reasons that clustering with K means is so attractive is because it is relatively simple. There is only one hyperparameter the researcher needs to concern themselves with, and that is the number of clusters, K. Choosing the appropriate K is non-trivial, and typically something that is not known a priori. The most popular method for choosing K is known as the ‘Elbow Method’, which is a kind of crude heuristic. The basic idea is to measure the distance of each point to their cluster center, with different numbers of clusters, and choose the number of clusters which minimizes each point’s distance to the cluster center, without diminishing returns. When these points are plotted out, the optimal number of clusters appears as an ‘elbow’ in the graph. The two most common measures used in the elbow method are *Distortion* and *Inertia* [1]*.* Where distortion is the average of the sum of squared distances to the cluster center, and inertia is just the sum of squared distances to the cluster center.

*Figure 5: Inertia elbow method*

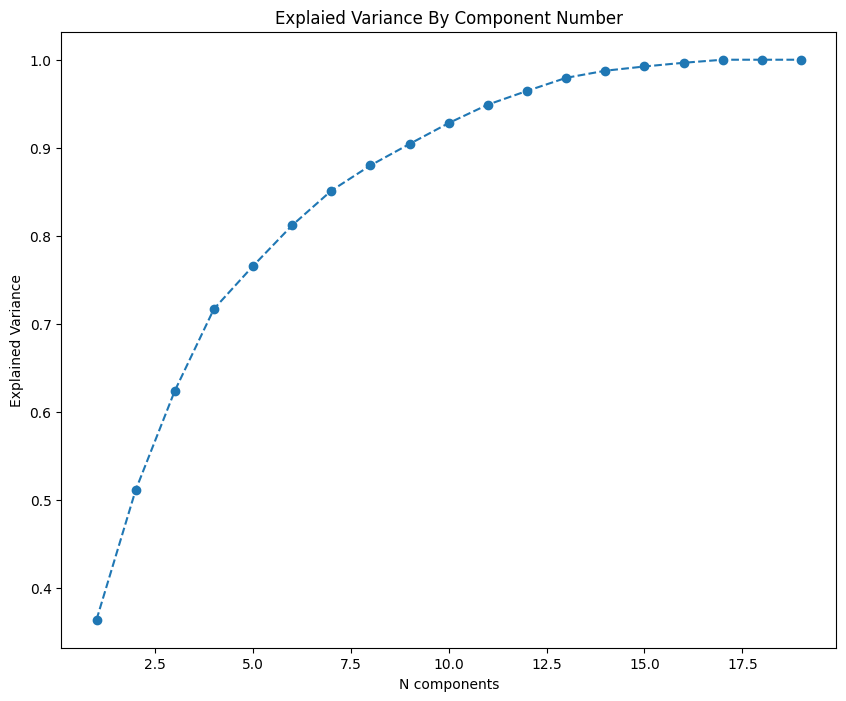


*Figure 6: Distortion elbow method*



Based on both methods, the optimal number of clusters is a bit ambiguous. However, there appear to be slight elbows at around 3 and 8 clusters. For the purposes of this analysis, the appropriate number of clusters will be assumed to be 8.

In addition to the ‘normal’ Kmeans, I will look at Kmeans with data that has gone through PCA to reduce the dimensionality of the data. This is often an important step in clustering because distance calculations in high dimensional spaces become ambiguous [2]. PCA offers a way to reduce the dimensionality of the data, without having to compromise by tossing out features. The first step in doing the PCA is to choose the number of components. Similar to Kmeans, a visual heuristic is helpful here. The general idea is to choose the number of components that explain ~80% of the variation in the data.



Based on the chart above, the number of components that explains 80% of the data is either 7 or 8 components. For this analysis 7 components will be used.

The next step is to choose the number of K for the data that has gone through PCA. This is also accomplished by the elbow method, and it has been found that 5 clusters is appropriate for Kmeans with PCA.

1. **Model Results**

The Kmeans model and the PCA Kmeans model produced distinct clusters with the following counts:

*Figure 7: Kmeans*

| Kmeans | N |
| --- | --- |
| 0 | 220 |
| 1 | 33 |
| 2 | 8 |
| 3 | 61 |
| 4 | 168 |
| 5 | 4 |
| 6 | 1 |
| 7 | 5 |

*Figure 8: Kmenas withe PCA*

| kmeans\_pca | N |
| --- | --- |
| 0 | 192 |
| 1 | 219 |
| 2 | 9 |
| 3 | 6 |
| 4 | 74 |

One of the most interesting aspects of the clusters is how uneven the cluster numbers are. There is nothing within Kmeans that specifies clusters have similar numbers, but the balance is surprising either way. Looking into Kmeans Cluster 5, the companies are JMP, C, BAC, and WFC. So, it would appear that the model clustered all big bank stocks together. This is great news, because all of these companies are structured similarly and have similar fundamentals.

Similarly, the PCA cluster 2 contains the companies META, GOOG, GOOGL, AMZN, AAPL, MSFT, WMT, CVX, and XOM. What is fascinating about this cluster is that it included big oil with big tech. Exploring this relationship further, this appears to be due to with the sheer size of these companies. Again, it seems that the clusters are grouping similar companies as we would hope.

Manually looking at clusters is a great way to get a feel for the data, but doesn’t really tell anything about how ‘good’ the clustering results are. For this, the Silhouette score can be used. The Silhouette score ranges from -1 to 1 where 1 means the clusters are separate with no overlap or missclassified points. Both models (Kmeans and PCA kmeans) yielded Silhouette scores of .22. Implying that the clusters are not very well segmented. A different number of K, different clustering algorithm, and/or removing/adding features to the analysis would be logical next steps to make the clustering better.

1. **Back Testing**

The ultimate goal of this analysis was to see if clustering equities based on their fundamentals would yield clusters of companies that perform better or worse than other clusters. To analyze this, I calculated the percent returns for each company from Jan 1 2024 - March 1 2024, and aggregated the returns by cluster. The results are as follows:

*Figure 9: Kmenas % Return*

Kmeans Cluster % Return

0 4%

1 3.6%

2 4.3%

3 5.5%

4 6.4%

5 7.8%

6 8.9%

7 2.8%

*Figure 10: Kmeans with PCA Return*

Kmeans w/ PCA Cluster % Returns

0 6.5%

1 4.2%

2 4.2%

3 5.5%

4 3.8%

Over this same time period, the S&P 500 had a return of 5.5%. There are three clusters from straight Kmeans that beat the average, and one cluster from the PCA analysis that beats the average. Therefore, based on these results I fail to reject the null hypothesis, and conclude that clustering can identify companies that perform better than other companies based on their fundamental indicators.

1. **Conclusion**

This analysis has shown that clustering equities on their fundamental indicators could yield an investment strategy that beats a broader index. It is important to note that the testing period for these clusters is rather short, and the strategy might play out differently over a longer time horizon. For those wishing to continue this line of work, different clustering algorithms might yield better results. Kmeans is a standard algorithm that is easy to understand, but hierarchical clustering is better suited to these types of problems.This is because Kmenas works best with spherical clusters, and the likelihood of the clusters being spherical is low given the number of features being considered.

**References**

1. Yse, D. L. (n.d.). *Introduction to K-means clustering*. Pinecone. https://www.pinecone.io/learn/k-means-clustering/
2. Wikimedia Foundation. (2024, February 27). *Clustering high-dimensional data*. Wikipedia. https://en.wikipedia.org/wiki/Clustering\_high-dimensional\_data