SMU Data Science Review

Volume 2 | Number 2 Article 18

2019

A Machine Learning Model for Clustering Securities

Vanessa Torres
Southern Methodist University, vanessat@mail.smu.edu

Travis Deason
Southern Methodist University, tdeason@mail.smu.edu

Michael Landrum
Southern Methodist University, mlandrum22@gmail.com

Nibhrat Lohria
Southern Methodist University, nlohria@mail.smu.edu

Recommended Citation
Torres, Vanessa; Deason, Travis; Landrum, Michael; and Lohria, Nibhrat (2019) "A Machine Learning Model for Clustering Securities," *SMU Data Science Review*: Vol. 2: No. 2, Article 18.
Available at: https://scholar.smu.edu/datasciencereview/vol2/iss2/18

A Machine Learning Model for Clustering Securities

Travis Deason, Michael Landrum, Vanessa Torres, and Nibhrat Lohria

Master of Science in Data Science
Southern Methodist University
Dallas TX 75275 USA
{tdeason, mblandrum, vanessat, nlohria}@smu.edu

Abstract. In this paper, we evaluate the self-declared industry classifications and industry relationships between companies listed on either the Nasdaq or the New York Stock Exchange (NYSE) markets. Large corporations typically operate in multiple industries simultaneously; however, for investment purposes they are classified as belonging to a single industry. This simple classification obscures the actual industries within which a company operates, and, therefore, the investment risks of that company. By using Natural Language Processing (NLP) techniques on Security and Exchange Commission (SEC) filings, we obtained self-defined industry classifications per company. Using clustering techniques such as Hierarchical Agglomerative and *k*-means clustering we were able to identify companies operating in similar industries. We found that the use of NLP to extract features the text was more important to model performance then model selection or optimization.

1 Introduction

As of April 2018, the US stock market was $34 trillion and of those trillions of US dollars, 84% of all those stocks belong to the wealthiest 10% of American families. Markets are highly complicated, and difficult to predict [6]. The concentration of wealth has fueled the growth of powerful predictive analytics. These tools often require access to resources, such as computing power and curated datasets, which are out of reach for the average investor. One of the most difficult things for the average investor, can be selecting stocks which are sufficiently independent from each-other and building a diversified stock portfolio. A diversified stock portfolio hold stocks across most sectors and helps mitigate idiosyncratic or unsystematic risks caused by factors affecting specific industries or companies within an industry. It aims to maximize returns in different areas that would each react differently to the same event, i.e. corporate scandals, natural disasters, or a sudden change in market conditions. And although it does not guarantee against loss, diversification is the most important component of reaching long-range financial goals while minimizing risk.

In addition to providing a risk management tool, clustering based on Natural Language Processing (NLP) analysis of SEC filings is also useful in measuring the footprint of a company or index fund. This can be useful when trying to measure the state of various industries. For example if a decline of stock at Amazon is not correlated with an overall decline of the retail sector, but is better correlated with information technology companies, it would point to the fact that Amazon is not strictly a ”Retail-Catalog & Mail-Order Houses” as is listed in it’s EDGAR filing; or as financial services companies have transformed into ”FinTech” companies, technology companies are also beginning to enter the financial services, calling them ”TechFins. [1]” The Global Industry Classification Standard (GICS) is a classification system for equities developed by Morgan Stanley Capital International and Standard & Poor’s. GICS hierarchy begins with 11 sectors, followed by 24 industry groups, 68 industries, and 157 sub-industries. Furthermore, the GICS strictly follows a coding system that assigns a code from each grouping to every company publicly traded in the market. Therefore, clustering is able to pick up on this distinction; while declarative classifications can be rigid and inaccurate.

The concentration of wealth certainly leaves the other 90% of Americans wishing for a *Magic 8 Ball* (manufactured by Mattel, (MAT)) that could tell them which stocks to buy that will put them into the top 10% and provide them a share of that $34 trillion; or at the very least, which stocks are going to help them better prepare for retirement. Whether it’s from a 401(k) plan or an IRA, or a Roth IRA, the trading of stocks is how most Americans save for retirement. With the growth of artificial intelligence, machine learning, and big data, Americans are now actively engaging in their retirement savings’ landscape and interacting with software programs or smartphone apps when making investment decisions or executing transactions [11].

The two largest publicly traded stock markets in the US, The New York Stock Exchange (NYSE) and The National Association of Securities Dealers Automated Quotation (NASDAQ) are home to over 3300 companies. Stocks are listed on a specific exchange, which bring buyers and sellers together and acts as a market for the shares. Because it could be difficult to monitor each security that trades in a market, stock market indices were created. A stock market index takes a group of stocks and tracks their collective value over time. Figure 1 gives an example of how NASDAQ may move[1].

The market that is representative of the whole and tracks the market’s chan-ges over time. Public stock markets allow institutional investors and private investors to purchase a small, or large, amount of a corporation; which in turn allow corporations to increase the amount of capital available to them. Due to the large amount of companies trading in the public markets, and the nearly unlimited factors which influence a company’s value, it can be difficult for an individual investor to make informed decisions regarding future stock values.

Securities trading has always been fertile ground for machine learning. Many of the largest hedge funds, such as Renaissance Technologies, Bridgewater Associates and Two Sigma have invested heavily in Machine learning to improve their investments. In the case of Bridgewater Associates, the use of machine learning enabled them to beat the Standard and Poor’s 500 index for over 20 years. Many commonplace tools in a Data Scientist’s arsenal were developed with stock trading in mind.

The vast majority of Machine learning models focus around predicting a stock’s future value, or predicting which stocks will have the greatest future value. This is typically a supervised learning and time-series analysis; where the past performance of stocks is utilized to predict their future value. These techniques have been utilized to

[1] Image can be found at https://www.nasdaq.com/symbol/ixic/stock-chart. Last accessed 2/20/2019

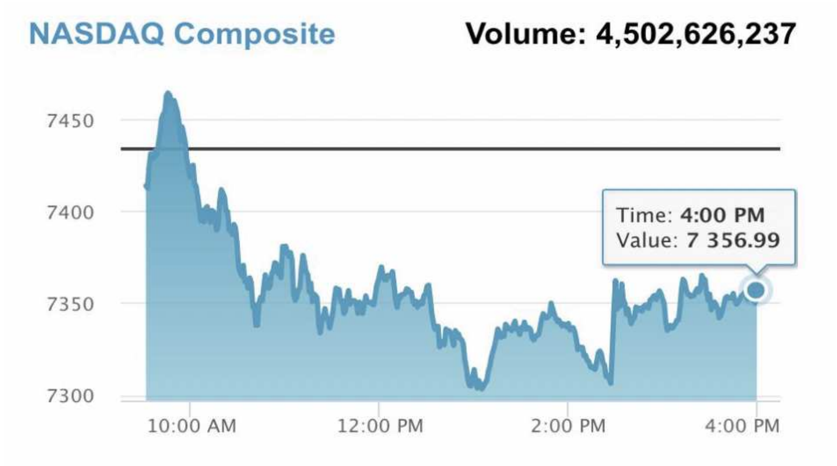


Fig. 1. NASDAQ Composite

varying success, and they are typically closely guarded secrets; that is, until they are no longer competitive.

This paper is focused on a different problem. Stock markets expose an investor to a large amount of risk. Any one stock may lose half, or all, it's value in a single day, and, since stocks are not independent from each other, any change in one stock will impact the value of other stocks. The exposure of stock ”A” to stock ”B” depends on how similar they are. Two companies operating with similar business strategies in similar markets often will rise and fall together. It is a widely accepted risk mitigation policy to diversify investments by owning securities in several uncorrelated market segments, but how correlated a pair of stocks are is typically measured by comparing how closely they track each other, or how closely they track the market. This can be problematic because a company can enter a new industry, change their business focus, or just be correlated to other stocks in a way which has not been reflected by its value.

Our method to solve this problem was to cluster companies based on the raw text information contained in their SEC filings.

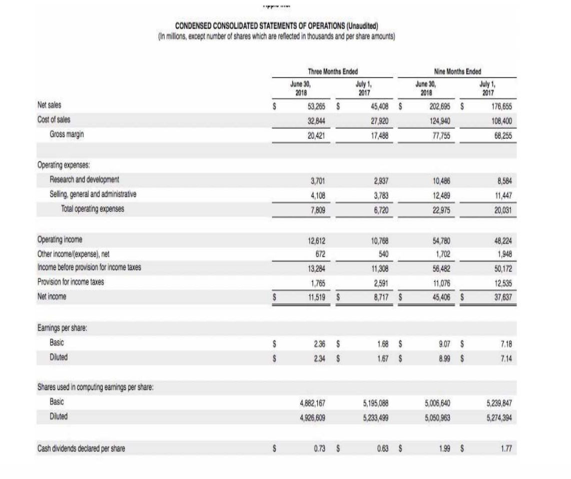
As with most Machine learning projects, data collection and cleaning can be problematic. Professional analysts track thousands of indexes for each stock. Since many of these indexes are industry secrets, we wanted to gather all of our data from free, publicly available, sources; which did not have consistent formatting. This required extensive use of natural language processing for feature extraction.

We evaluated several different model types in clustering securities. Some of those model types include k-means, Hierarchical Agglomerative clustering, and Latent Dirichlet allocation.

2 Dataset Selection

The primary data source for this project is the 10-Q quarterly reports available on Edgar at sec.gov. Since we were looking for industry defining keywords in these reports, we pulled and stored raw text here. The tables, balance sheets, and other financial information were not considered relevant for the purpose of this project. We were interested in creating as comprehensive of a model as possible, so we collected thousands of these reports.

We want our model to be accessible to an amateur investor; where buying and selling the same stock within a single day would incur prohibitively expensive brokerage fees, We were not focused on predicting variations that occur throughout a single day; so we decided to focus on a single price point to use as our dependent variable for our model. For the purposes of our model, it did not matter if we chose the open or closing cost, but using the high/low values introduced greater variance to our model. The closing cost was chosen because it was deduced that an amateur investor is most likely to place orders for stocks after the markets are closed.



CONDENSED CONSOLIDATED STATEMENTS OF OPERATIONS (Unaudited)
(In millions, except number of shares which are reflected in thousands and per share amounts)

	Three Months Ended		Nine Months Ended	
	June 30, 2018	July 1, 2017	June 30, 2018	July 1, 2017
Net sales	$ 53,265	$ 45,408	$ 202,695	$ 176,655
Cost of sales	32,844	27,920	124,940	108,400
Gross margin	20,421	17,488	77,755	68,255
Operating expenses:				
Research and development	3,701	2,937	10,486	8,584
Selling, general and administrative	4,108	3,783	12,489	11,447
Total operating expenses	7,809	6,720	22,975	20,031
Operating income	12,612	10,768	54,780	48,224
Other income/(expense), net	672	540	1,702	1,948
Income before provision for income taxes	13,284	11,308	56,482	50,172
Provision for income taxes	1,765	2,591	11,076	12,535
Net income	$ 11,519	$ 8,717	$ 45,406	$ 37,637
Earnings per share:				
Basic	$ 2.36	$ 1.68	$ 9.07	$ 7.18
Diluted	$ 2.34	$ 1.67	$ 8.99	$ 7.14
Shares used in computing earnings per share:				
Basic	4,882,167	5,195,088	5,006,640	5,239,847
Diluted	4,926,609	5,233,499	5,050,963	5,274,394
Cash dividends declared per share	$ 0.73	$ 0.63	$ 1.99	$ 1.77

Fig. 2. Excerpt from quarterly report for Apple Inc.

Additional features were extracted from SEC filings and quarterly earnings reports. SEC data is publicly available and provides a wealth of information regarding the financial condition of a company at a given time. Since these data-sources are typically in plain text format, this information is not always consistent from report to report. Using natural language processing methods, we were able to group companies into categories. We also could perform sentiment analysis to determine if an earnings report was positive or negative. Figure 2 is an excerpt of the Apple quarterly report from Q42018.

Much of the financial information in this table is unique to Apple and cannot be featured into our model; while some of the information, such as revenue, is required to be included in quarterly reports, and would be highly beneficial to a regression type model. This project will remain focused on unsupervised learning and natural language processing, NLP. For this reason, the unstructured text will form the input for the model.

While the format for these reports is similar, they are not identical, and the information contained therein is critical to the predictive capabilities of our model. This project did not attempt to extract financial information from these tables; because it was found to be redundant. This project focuses on the verbiage used in the text of the reports. This is consistent throughout all the 10-Q reports we analyzed.

In total, our dataset contains ten columns for 7,196 10-Q filings dating back an average of fifteen years, or 431,760 rows of data. We determined this was an adequately sized dataset to build and evaluate our model.

3 Data Preparation

3.1 Pre-processing

All 10-Q filings are public record. We extracted these filings using a web-scraper. These files are downloaded as html documents. We then extract all text which is has been organized into sentences and paragraphs. This information is then saved locally for later processing. We omit text headers, tabular data, and graphs; because they are typically made redundant by the text of the report.

The Security and Exchanges Commission, SEC, does not provide an application protocol interface, API, for 10-q files, but web scraping is permitted. We took care to limit the number of documents being requested at a time; which made scraping take longer, but it was considered necessary to prevent the appearance of a denial of service attack by our server.

3.2 3.2 Text processing

Since this project focuses on NLP, the bulk of the time spent in model optimization went into processing the text. For the purpose of this paper, the term ”document” refers to a single 10-q filing, and the corpus which they will be compared with is all the other 10q documents for the same quarter. Since language evolves over time, this allows each quarter to be observed independently. We used a variety of NLP tools in processing the text, which will be covered in detail throughout this section, the NLP pipeline is as listed below [4]:

- Tokenize sentences
- Apply parts of speech tagger
- Isolate Nouns
- Remove stop words
- Convert words to stems
- Convert words to Features using Words2Vec

The first step of this process, tokenize sentences was completed using the Natural Language Processing Toolkit's, NLTK, sentence tokenizer. To further clean up the sentence tokens, we ran them through a reg-ex filter to remove remaining punctuation and split up hyphenated words.

Once the document has been split up into tokens, we removed all stop words, which are normal every day words that provide no meaning such as the, and, etc. We then used a simple parts of speech tagger to gather all the noun-parts using the NLTK library. Here we are only interested in the nouns, so we drop all words which were not labeled nouns by the part of speech tagger. This process is imperfect, but since these documents are fairly large, we are able to extract enough information to build a useful model.

Once we have eliminated all non-noun words, we applied a lemmatizer to the text to reduce all the words to stems. This part of the process required some tuning, as some of the terms used in financial reports don't fit a standard corpus, but more details on that process can be found in the appendix of this report. Our target here was to decrease the degrees of freedom as much as possible prior to vectorizing the remaining corpus.

Now that we had a list of nouns for each text, we then found the word counts for each word in each text and exported them as individual json files. Due to the size of each quarter (approximately 35 GB per quarter), we were not able to hold all the data at once in memory. Using this process allowed us to shrink the data into about 160 MB worth of information which is much more manageable.

The final step in the pre-processing pipeline is to take the word stems of our noun-parts, and convert them to vectors. There are several tools available for this task, but we had over 10,000 words per model, and we were interested in using as many of the words as possible; so we used two methods to compare their results. The first was a simple TF-IDF (term frequency-inverse document frequency) vectorization. With this method, terms do not appear frequently throughout the corpus are weighted much higher than words that are found in every document. The second method we used was the Words2Vec algorithm. Word2Vec uses a single layer neural network trained on the corpus of words in our dataset. Using Word2Vec we reduce all noun counts down to 40 vectors, where each vector is a weighted grouping of some, or all of the nouns in the corpus. Word2Vec has the added bonus of attempting to add semantic analysis and relating word vectors to each other. For instance, it may notice that the adding the vector for King to the vector for Woman is similar to the vector for Queen [9].

4 Model Determination

4.1 Model Types

The two models we utilized are *k*-Means and Hierarchical agglomerative clustering. These models served different purposes. We found Hierarchical clustering work slightly before we did any extra dimensionality reduction, whereas *k*-means worked significantly better once we reduced the dimensions. For this reason, our final models were *k*-Means. The finite boarder between groups gave the best performance in our correlative score based model evaluation.

4.2 *k*-means

k-means is an unsupervised machine learning tool that has a close relation to *k*-nearest neighbors. *k*-means attempts to classify each document into one of *k* clusters by finding the minimum variance within each cluster. The number *k* is determined by the user of the algorithm.

There are a few methods for helping determine the value for *k* and we chose the Elbow Method. For this method, run *k*-means for all possible values for *k* and compute the total within-cluster sum of square (wss). Plot the values for the wss for each value of *k* and the point where the curve bends (the elbow) is generally considered the value for *k*. Figure 3 is an example of when there is a very obvious elbow.

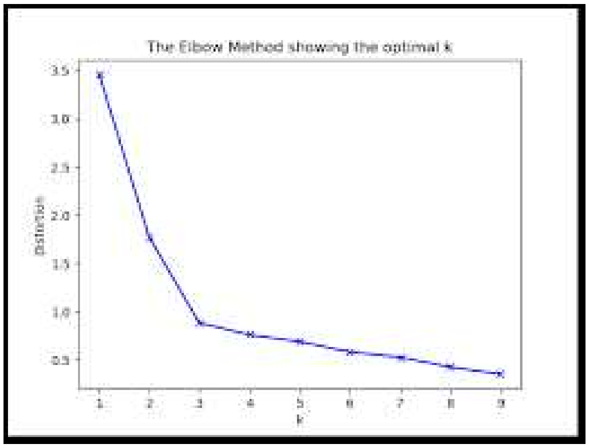


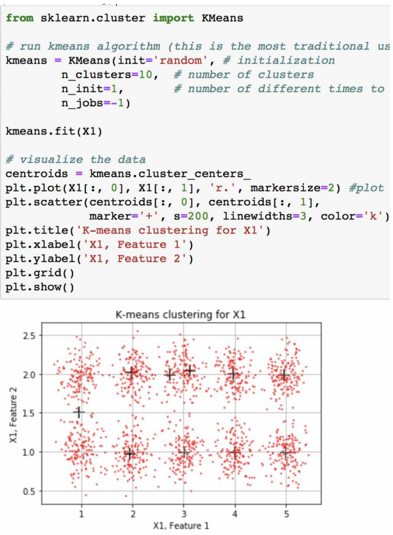
Fig. 3. Elbow Method

The algorithm for finding the clusters is as follows: First, *k*-means randomly assigns *k* points in the vector space to be the center of each cluster. It runs through each document and finds the euclidean distance from each center and clusters it with the center it is closest to. Once all documents are clustered, it finds the new center for each cluster and starts the process all over with the new centers. It keeps running until the centers are unchanged or only changed slightly [8].

k-means is a very efficient algorithm and so it's best used as a baseline to compare more expensive algorithms. Figure 7 shows an example of how *k*-means would cluster the group of points.

4.3 Hierarchical agglomerative clustering

Hierarchical clustering is another unsupervised machine learning tool, and is more computationally intensive than *k*-means. There are two methods of Hierarchical clustering: Agglomerative (bottom-up) and Divisive (top down). We just focused on Agglomerative. Essentailly, Hierarchical Agglomerative clustering starts at the bottom layer and



```
from sklearn.cluster import KMeans

# run kmeans algorithm (this is the most traditional us
kmeans = KMeans(init='random', # initialization
        n_clusters=10,  # number of clusters
        n_init=1,       # number of different times to
        n_jobs=-1)

kmeans.fit(X1)

# visualize the data
centroids = kmeans.cluster_centers_
plt.plot(X1[:, 0], X1[:, 1], 'r.', markersize=2) #plot
plt.scatter(centroids[:, 0], centroids[:, 1],
            marker='+', s=200, linewidths=3, color='k')
plt.title('K-means clustering for X1')
plt.xlabel('X1, Feature 1')
plt.ylabel('X1, Feature 2')
plt.grid()
plt.show()
```

Fig. 4. k-means

looks at each individual cluster and links them to documents that are similar. Once it does that, it starts to move up and link clusters that are similar. It continues this process until it has the desired number of clusters [10]. A visual representation for a basic example of Agglomerative clustering can be found in Figure 8[2]

4.4 Latent Dirichlet Allocation (LDA)

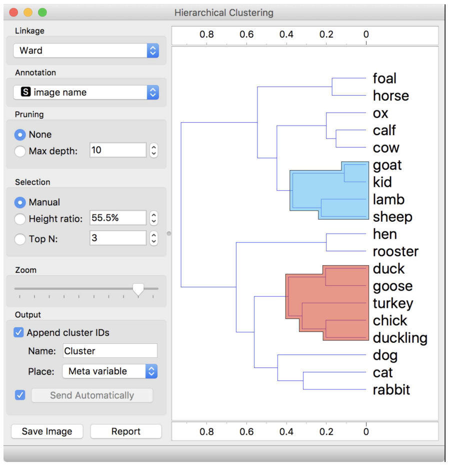
LDA is an example of a topic modeling statistical model. Each document is made up of several topics, and then the documents are grouped together based on the similarity of their topics. One document might contain both ”cat” topic and a ”dog” topic and LDA would determine the words ”cat”, ”kitten”, and ”meow” are a part of a ”cat” topic whereas ”dog”, ”puppy”, and ”bark” are part of a ”dog” topic [5]. ”LDA extracts insights from the documents, themselves using the data-up approach to define common themes these are the topics and report on where, and to what extent, they appear in each document. [2]”

5 Analysis

5.1 Models Before Dimension Reduction

The first round of clustering was done prior to any dimensionality reduction. The first model was *k*-means with a *k* size of 8. From the elbow method, we found that we needed at least 6 or 8 clusters. We chose 8 as it looked to cluster better, visually, in 2-dimensions

[2] Zupan, BlaÅ: Orange Data Mining (2017)



Hierarchical Clustering

Linkage
Ward

Annotation
image name

Pruning
● None
○ Max depth: 10

Selection
● Manual
○ Height ratio: 55.5%
○ Top N: 3

Zoom

Output
☑ Append cluster IDs
Name: Cluster
Place: Meta variable
☑ Send Automatically

Fig. 5. Example of Heirarchical Agglomerative Clustering

and also that the SIC has 10 distinct codes (plus ”not used” and ”Nonclassifiable” classifications) so we wanted to get close to their model.

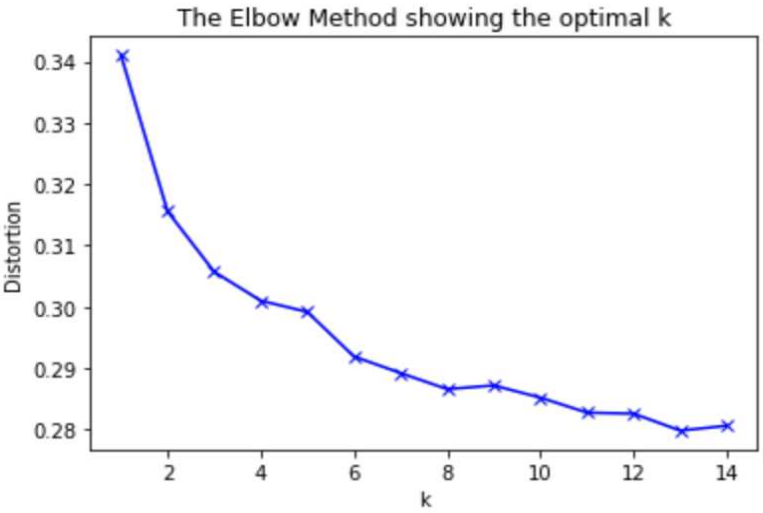


Fig. 6. The Elbow Method showing the optimal k

Figure 6[3] is the visual used to find the optimal number of clusters for the not reduced set of text. This was a total of 8,000 distinct feature words. Some dimensional reduction was needed. *k*-means.

Figure 7 is a 2-dimensional graphical representation of our 12 clusters generated by the non-reduced data using *k*-means. To improve visibility, the data has been transformed through principal component analysis onto a two dimensional space.

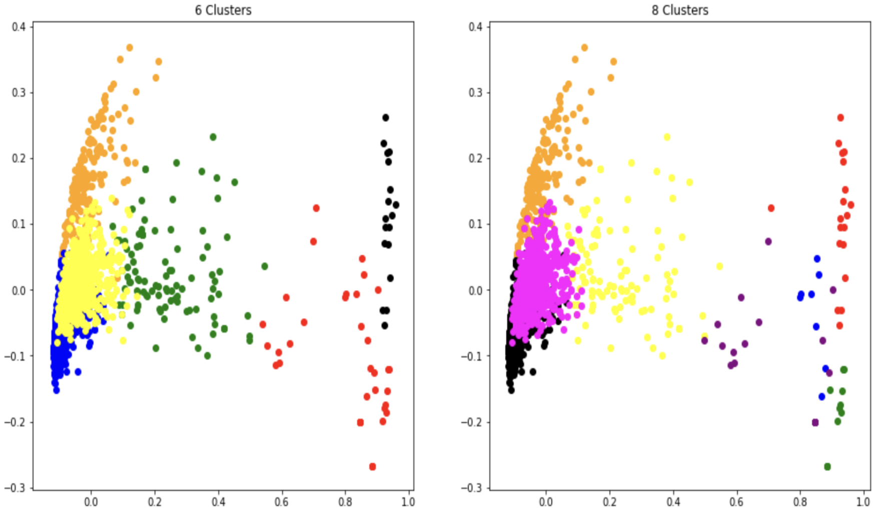


Fig. 7. k-means With 6 and 8 Clusters

We also ran Hierarchical Agglomerative clustering models using 6 and 8 clusters that can be seen in Figure 8. Visually the Hierarchical and *k*-means models were very similar in the large feature space.

These clusters are very inconsistent in size with a single cluster consisting of most of the data-points. Looking at the data, it is clear that much of data is bunched around a single point; so all clustering will result 1-3 very large clusters, with the rest of the data spread across very sparse clusters. To fix this, we reduced the feature space of the data by altering the cleaning process of our data. The first thing we did was dropped all words which were not English. These words were then run through a Porter Stemmer. This reduced our word count to about 1500.

To further decrease the size of our feature space, we changed some of the features of the Term Frequency – Inverse Document Frequency Vectorizer to leave out more words which are common to most documents. We wanted to ensure we were not classifying based on common words. This was done to spread out most of the observations; so that the 10-Q filings we were clustering were somewhat evenly spread out across all

[3] Image found at https://pythonprogramminglanguage.com/wp-content/uploads/2017/07/elbow-method.png. Last accessed August 20, 2019

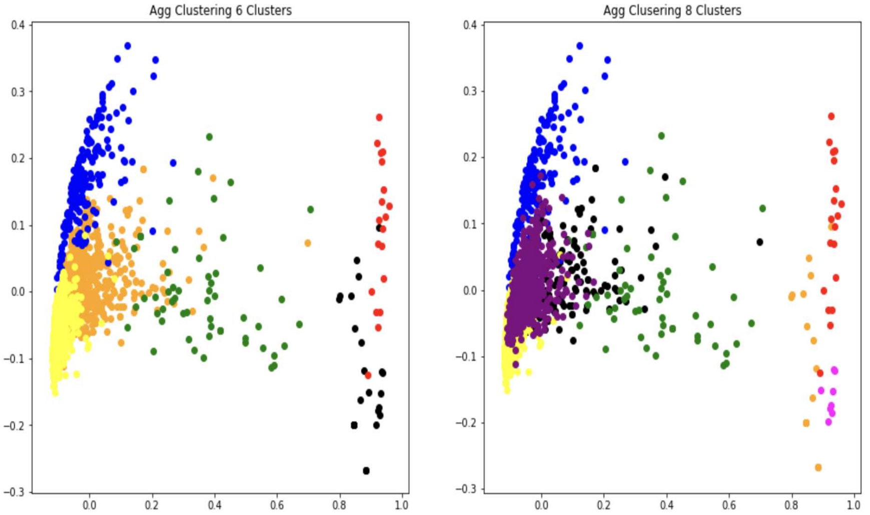


Fig. 8. Hierarchical Clustering With 6 and 8 Clusters

of the clusters so we reduced the threshold document frequency down to 0.29. We also wanted to ensure that we were not classifying based on words that were dominant in just a few filings. This was done to reduce the amount of data-points which are off in distant euclidean space when compared to the other points. This was done by increasing the minimum document frequency required to include words in our model. The effect of increasing the minimum document frequency has on the visualization is that it decreases the amount of points which are in the distance. We used a value of .08 for minimum document frequency.

With all of these changes made, we were left with 320 features. This feature space could have been adequate, but as an additional step, we utilized Principal Component Analysis to reduce the feature space down to 50. We ran both *k*-means and Hierarchical Agglomerative clustering with both 6 and 8 clusters. The results of both of these models are shown in figure 9 for *k*-means, and figure 10 for Hierarchical Agglomerative.

The difference here between *k*-means and Hierarchical Agglomerative is much more pronounced than it was for the larger feature space. *k*-means produces visibly more cohesive clusters than Hierarchical Agglomerative. This is at least true when they are transposed into two dimensional space. We also observed that there is a much more clear pattern to the data, and the observations are much better distributed throughout the feature space. There are no clusters that only represent a small amount of observations in this data. There is still a single cluster which is much larger than all of the others, but the severity of this delta is much less significant, and it now seems feasible that the center cluster represents something inherent in the data.

We used LDA and Topic Modeling to guide us in visualizing the them in each cluster. For instance, we were able to determine cluster 6 in Figure 11 were all energy

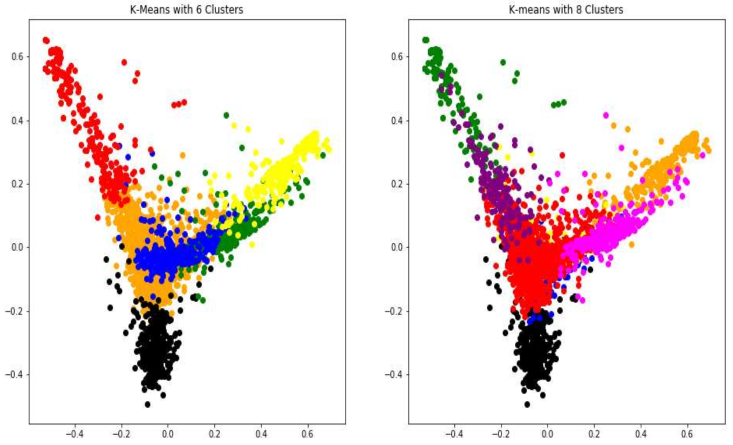


Fig. 9. Optimized Kmeans Clustering With 6 and 8 Clusters

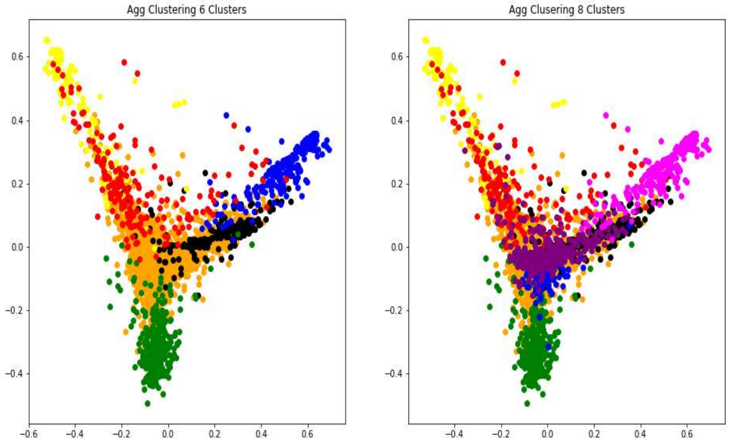


Fig. 10. Optimized Hierarchical Clustering With 6 and 8 Clusters

companies. The words power, energy, and utility were, for the most part, only found in documents in this cluster.

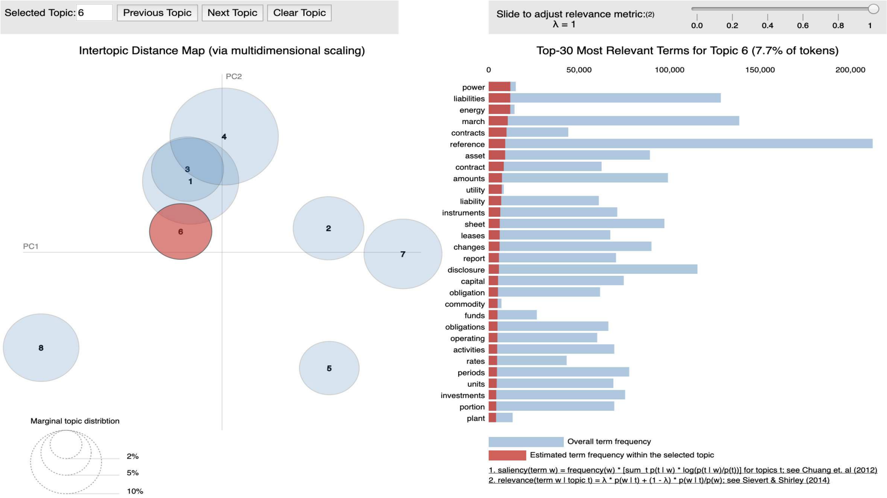

Selected Topic: 6
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 1
0.0
0.2
0.4
0.6
0.8
1
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
4
3
1
6
2
7
8
5
Marginal topic distribtion
2%
5%
10%
Top-30 Most Relevant Terms for Topic 6 (7.7% of tokens)
0
50,000
100,000
150,000
200,000
power
liabilities
energy
march
contracts
reference
asset
contract
amounts
utility
liability
instruments
sheet
leases
changes
report
disclosure
capital
obligation
commodity
funds
obligations
operating
activities
rates
periods
units
investments
portion
plant
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Fig. 11. Topic Modeling

Since this is entirely unsupervised Machine Learning, it's not straight forward to determine how well our model worked. Our method of validation utilized a pair of companies that we expected to be incorrectly classified by SIC code and compare them to other companies within their same SIC classification. We would then compare those companies to companies within the clusters determined by our model. Since our feature space is sparse, the metric we use to compare is cosine similarity. Cosine similarity is used to determine how similar documents are irrespective of their size.

The first of those companies is General Electric (GE). General Electric The SIC code for 4911 is described as Electric Services, or Establishments engaged in the generation, transmission, and/or distribution of electric energy for sale. [4] We compared the cosine similarity of those documents to the documents our models considered to be in the same cluster as GE. Cosine Similarity determines the similarity between two documents by finding the angle in vector space in between the two documents. A cosine similarity of 1 would signal two documents are essentially identical, where a score of 0 would mean they are entirely unrelated.

The cosine similarities were as followed:

- GE vs. companies that had the same SIC code: 0.9068
- GE vs. companies within the same *k*-means cluster: 0.9743

[4] More information can be found at https://www.osha.gov/pls/imis/sic_manual.display?id=945&tab=description. Last accessed August 18, 2019

- GE vs. companies within the same Hierarchical cluster: 0.9767

While Hierarchical performed marginally better in this test, that could have been purely by chance. But both models were much better than the SIC code classification. We also found the Cosine Similarity between GE and 50 randomly selected customers to be over 0.9, which is very high. The high correlation between uncorrelated documents suggests there's a lot of redundancies in the documents. This lead us to believe we need to start working on reducing the feature size of our model.

Another company we wanted to check was Amazon. Amazon has a SIC code of 5961. The SIC code for 5961 is described as Establishments primarily engaged in the retail sale of products by television, catalog, and mail-order. [5] For Amazon, we decided to compare our cluster to companies within the sic code, and companies which Amazon competes with directly in the tech sector.

The cosine similarities were as followed:

- Amazon vs. companies that had the same SIC code: 0.9542
- Amazon vs. companies which it competes directly with in tech: 0.8253
- Amazon vs. companies within the same *k*-means cluster: 0.9519
- Amazon vs. companies within the same Hierarchical cluster: 0.8786

6 Ethics

In data analytics, ethics and privacy are extremely important and should adhere to the FORTS framework (fairness, openness, reliability, trust, social benefit) [7]. In conjunction with Data for Democracy's grassroots ethical data initiatives, the Global Data Ethics Project (GDEP) created the FORTS framework and a set of principles to be used by data practitioners and advocates. This creation is a result from the announcement at the September 2017, Bloomberg D4GX conference, where Lilian Huang introduced the ethics work to the Data for Democracy's online community. The framework and principles dive into concepts of privacy, transparency, consent, bias, diversity and ethical imagination. The Global Data Ethics Principles aim to [7]:

- Consider informed and purposeful consent of data subjects for all projects, and discard resulting data when that consent expires.
- Make best effort to guarantee the security of data, subjects, and algorithms to prevent unauthorized access, policy violations, tampering, or other harm or actions outside the data subjects' consent.
- Make best effort to protect anonymous data subjects, and any associated data, against any attempts to reverse-engineer, de-anonymize, or otherwise expose confidential information.
- Practice responsible transparency as the default where possible, throughout the entire data lifecycle.
- Foster diversity by making efforts to ensure inclusion of participants, representation of viewpoints and communities, and openness. The data community should be open to, welcoming of, and inclusive of people from diverse backgrounds.

[5] More information can be found at www.sec.gov. Last accessed June 10th 2019

- Acknowledge and mitigate unfair bias throughout all aspects of data work.
- Hold up datasets with clearly established provenance as the expected norm, rather than the exception.
- Respect relevant tensions of all stakeholders as it relates to privacy and data ownership.
- Take great care to communicate responsibility and accessibly.
- Ensure that all data practitioners take responsibility for exercising ethical imagination in their work, including considering the implications of what came before and what may come after, and actively working to increase benefit and prevent harm to others.

The ethical considerations brought to this project and paper, includes data retrieved and scraped from publicly available SEC filings. The U.S Securities and Exchange Commission (SEC) provides fair access to the content found in the SECs EDGAR (Electronic Data Gathering, Analysis and Retrieval system). SECs efforts to make data accessible and easy to use has also been fully supported by the Office of Structured Disclosure (OSD), within the Division of Economic and Risk Analysis. Furthermore, the SEC adheres to the Privacy Act of 1974 and the requirements with respect to all information about individuals that it collects, maintains, uses or disseminates in a System of Records.

When searching EDGAR for filings, results are presented in chronological order and identified by the filings form type; annual report (10-K), quarterly report(10-Q), or by current report (8-K). The disclosed financial information are in an Inline XBRL format, a format that allows third-party programs to read and parse the information [3].

Our objective was to quantitatively estimate the similarity between the SEC filings in a format that could help investors research and discover companies on their own. Utilizing structured data and analytics to enhance disclosed information to investors can only help in determining whether a security is a good investment. Needless to say, sometimes doing the right thing doesn’t yield favoring results but as data practitioners, remaining ethical, harnesses the positive impacts of data innovation.

7 Conclusions

This project only gives a glimpse of the power of Natural Language Processing and how to extract information from the text of SEC filings. We focused only on industry related clusterings to identify companies that operate in the same industry. But, the SEC filings contain significantly more information than simply the operating industries of the company.

This was not as straight forward as we initially anticipated and the Extract Transform and Load (ETL) pipeline gave us the most headache. Once we were able to jump that hurdle, we were able to find interesting results. The 80/20 Data Science Rule (80% of the work was spent on finding, cleaning and organizing data, whereas only 20% was on analysis) was followed for this project.

8 Future Work

This project did not touch on the financial side of the stock market even though that's what the stock market is best known for. In future work, we'd like to see how are clusters fair in the ebbs and flows of daily trading.

A next step would be finding the correlation of daily stock price changes between companies with in the same clusters and compare them with the correlation of daily stock price changes between companies in different clusters. If we can find that companies in different clusters are more or less independent of each other, then we can use our model to diversity our portfolios. Until then, this was just a comparison between the SIC classifications and our own.

References

1. Barberis, J.: From fintech to techfin: Data is the new oil. The Asian Banker (2016), http://www.theasianbanker.com/updates-and-articles/from-fintech-to-techfin:-data-is-the-new-oil
2. Bauguess, S.: The role of big data, machine learning, and ai in assessing risks: a regulatory perspective. Champagne Keynote Address: OpRisk North America (2017), https://www.sec.gov/news/speech/bauguess-big-data-ai
3. Bauguess, S.: The role of machine readability in an ai world. Financial Information Management (FIMA) Conference (2018), https://www.sec.gov/news/speech/speech-bauguess-050318
4. Bird, S., Klein, E., Loper, E.: Natural Language Processing with Python. O'Reilly Media Inc (2009)
5. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research. **3** (2003)
6. Cohen, P.: We all have a stake in the stock market, right? guess again. NY Times (2018), https://www.nytimes.com/2018/02/08/business/economy/stocks-economy.html
7. Huang, L.: Global data ethics project. Data For Democracy (2018), https://www.datafordemocracy.org/documents/GDEP-Ethics-Framework-Principles-one-sheet.pdf
8. MacKay, D.: Information Theory, Inference and Learning Algorithms, chap. 20, pp. 284–292. Cambridge University Press (1993)
9. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv **3** (2013)
10. Rokach, L., Maimon, O.: Data Mining and Knowledge Discovery Handbook, chap. Clustering Methods, pp. 321–352. Springer, Boston, MA (2005)
11. Trainer, D.: Why robo-analysts, not robo-advisors, will transform. Forbes (2017), https://www.forbes.com/sites/greatspeculations/2017/07/19/why-robo-analysts-not-robo-advisors-will-transform-investing