CSCI 4022 Final Project
Kai Hueske-VanCeylon

# Post Minhashing/Clustering on r/WallStreetBets

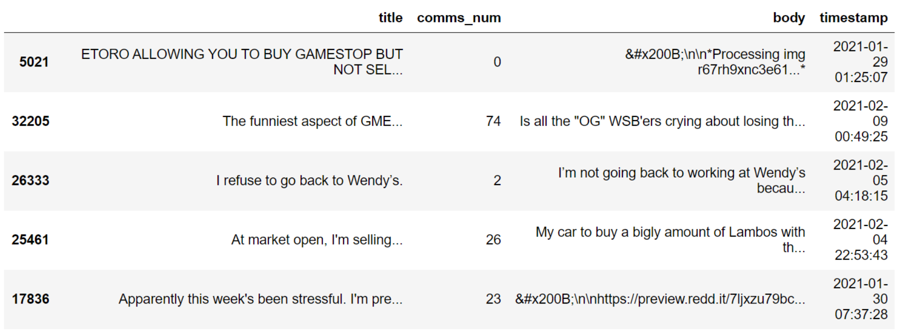
## Motivation

The wider world recently got a chance to see how much of an impact seemingly abstract and unimportant happenings in the Internet can have on tangible facets of everyday life. The January short squeeze on Gamestop, AMC, and Blackberry stocks, driven almost entirely by the r/WallStreetBets subreddit, is an interesting opportunity to study this dynamic further. This project investigated the effectiveness of minhashing and k-means clustering to group posts from r/WallStreetBets into categories. Specifically, I was hoping to determine whether there was a temporal dimension to these clusters, and whether these clusters had any correlation with events in the real world - for example, did the early increase in Gamestop stock (from $4 in June 2020 to $16 in November) correspond to an increase in discussion of Gamestop in r/WSB? Could this apply to other events, other topics of conversation?

Personally, I think the powerful effect an loose, organically organized group of like-minded people can exert on an international stock market is incredibly interesting. Even without considering the ethics or power dynamics of the situation, it’s hilarious that a community based around memes can affect massive corporations so powerfully. But more than simply a project to satisfy my own curiosity, this type of model/approach could be useful in a variety of situations. It could be used as a precursor to or enhancer of sentiment analysis: documents in a particular medium (social media like Reddit or Twitter, or news publications) could be grouped into categories, which could yield particular keywords/topics of conversation for each cluster. Sentiment analysis could then be performed on the documents in each category with regard to these topics. These categories could relate to temporal events, locations, or general topics. Different media would present different challenges, but as a general approach, it could have potential.

## Data

The data for this project come from Gabriel Preda’s excellent collection of data on r/WallStreetBets, hosted on Kaggle (https://www.kaggle.com/gpreda/reddit-wallstreetsbets-posts) and collected using the PRAW Reddit API. Each entry represents a post, and contains the title, number of comments, text (in the ‘body’ column), and timestamp, as well as several columns (the post’s numerical ID, score, URL, and creation UTC timestamp). A sample is shown below.



| | title | comms_num | body | timestamp |
|---|---|---|---|---|
| **5021** | ETORO ALLOWING YOU TO BUY GAMESTOP BUT NOT SEL... | 0 | &#x200B;\n\n*Processing img r67rh9xnc3e61...* | 2021-01-29 01:25:07 |
| **32205** | The funniest aspect of GME... | 74 | Is all the "OG" WSB'ers crying about losing th... | 2021-02-09 00:49:25 |
| **26333** | I refuse to go back to Wendy’s. | 2 | I’m not going back to working at Wendy’s becau... | 2021-02-05 04:18:15 |
| **25461** | At market open, I'm selling... | 26 | My car to buy a bigly amount of Lambos with th... | 2021-02-04 22:53:43 |
| **17836** | Apparently this week's been stressful. I'm pre... | 23 | &#x200B;\n\nhttps://preview.redd.it/7ljxzu79bc... | 2021-01-30 07:37:28 |

A sample of the dataset

I used a random subsample of this dataset of size 7000, for the purposes of quicker training. Initially I chose to use the first 7000 entries, but these results were clustered over a single day; a random 7000 ended up being a little more informative about long-term patterns.

## Context

This problem has been investigated before. Becker, Naaman, and Gravano’s 2009 paper on ensemble clustering in social media addresses a similar problem. It deals particularly with event detection based on social media posts using clustering. Their approach is to use TF-IDF & cosine distance metrics as similarity scores between documents, and ensemble clustering to group documents. However, the data used in their study was labeled with particular events that each document corresponded to, which is different from this project’s unsupervised approach.

The Becker/Naaman/Gravano paper does mention the noisiness of social media, and how it differs from more structured media such as news documents. That agrees with the results of this project; there are certain structures in social media data (like recurrent or copied posts) that can affect the results of clustering which aren’t present in different forms of media. There also tends to be more documents, with less text, than, for example, a dataset of magazine articles; however, the documents often contain more contextual information like timestamps or locations. While this can cause issues (such as the increase in the size of our similarity matrices), it can also provide useful opportunities for diligent data scientists.

## Exploratory Results

The data itself spanned from mid-2020 to April of 2021, and the posts contained mostly text, links, and rocket/diamond hands icons. Originally, I thought that the data was of a manageable size, since after sanitizing the text and dropping the rows with empty post bodies I was left with about 2100 posts - not terrible overall. I was quickly disabused of that notion. As a preliminary step, I performed shingling on the dataset using SKLearn’s CountVectorizer module. The initial characteristic matrix of the data had 600,000 5-shingles and 7,000 documents, for a total of 4 billion entries (to my great dismay). However, as tends to be the case with characteristic matrices, it was very sparse; there were only around 635,000 nonzero entries.

Since text (and its associated ridiculously-high-dimensional vectorization representation) tends to be a little harder to visualize or calculate minima/maxima on than numerical data without actually performing the methods I'd planned, this was about the extent of my exploratory undertakings.

## Methods

My goal was to find the main topics that the posts in the subreddit discussed. A clustering algorithm made the most sense for this; this meant I'd also need a metric of distance between documents (ideally one which wouldn't require supercomputer time to compute). To this end, the methods I chose were shingling, minhashing, and k-means clustering. Since my focus was on the latter two, and particularly on implementing them in clever ways that didn't take obscene amounts of time, I used SKLearn's CountVectorizer to facilitate the shingling.

I chose to shingle on words, and chose shingle lengths of 1, 3, and 5 words. This was so that the similarity measures would be based on content/vocabulary (more appropriate for the higher-level topic/event categories that I hoped to find) rather than just shared letters. 1-shingles would hopefully give the model a general idea of conversation topic, while longer shingles might introduce higher-level structural elements like sentiment or reasoning. Combining each document's shingle representation yielded a characteristic matrix for the dataset which could be used in the minhashing algorithm.

I adapted the minhashing algorithm we'd developed in class & in the homework to this particular problem. It still uses the universal hash function of the form:

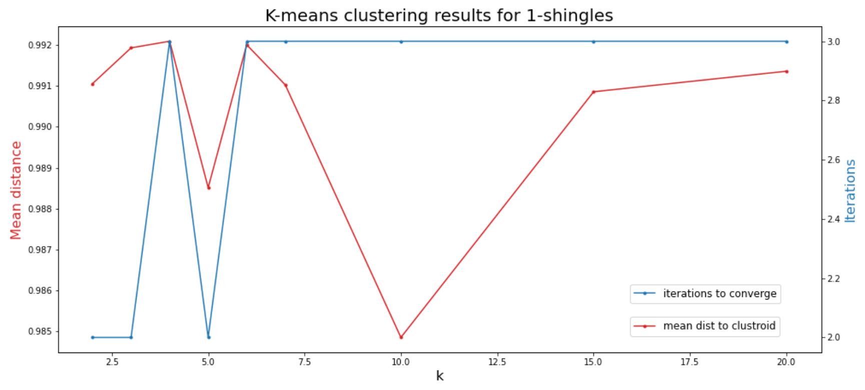
$$h(x) = ax + b \pmod{P} \pmod{S}$$

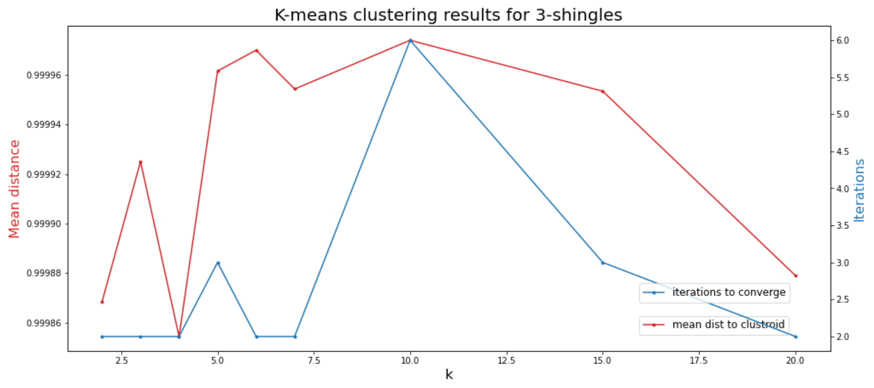
where $P = 1877983$ is a prime larger than our number of characteristic matrix rows, and $S$ is the number of shingles in our dataset. To avoid iterating over every row and every column of our characteristic matrix, I refactored the algorithm to only check & hash on the nonzero entries, using Scipy's compressed sparse row matrix representation and its convenient nonzero() method. I chose to use 150 hashes, as that sped up computation significantly without sacrificing too much accuracy. (Initially, I had tried to choose the same number of hashes as there were documents (21000); after an entire night spent training yielded a memory overflow at 6am, I realized that a lower number of hashes was one of the selling points of the minhash algorithm.) I performed minhashing for each of the 3 different shingle lengths, which yielded 3 corresponding signature matrices. I used these to compute upper-triangular similarity matrices containing the pairwise distances between each document. (This was my motivation to use a smaller subsample of the dataset; this alone would've taken days to calculate each matrix on the whole dataset, and I just didn't have that sort of time.) With this measure of similarity in hand, I moved on to clustering.

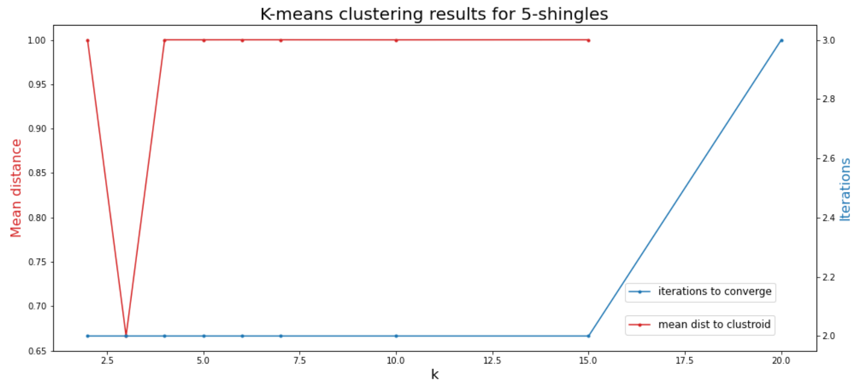
I used a basic k-means clustering algorithm, using clustroids rather than centroids since these metrics didn't allow a calculation of "average position" among the documents in each cluster. It's possible that hierarchical clustering would've been more appropriate, but the time complexity wasn't workable. After converting the upper-triangular similarity matrices into full matrices for convenience, I clustered each shingle's dataset using random clustroid initializations for each $k \in \{2, 3, 4, 5, 6, 7, 10, 15, 20\}$. I saved the number of iterations each clustering took to converge, as well as the clustroids and the mean clustroid distance for each value of *k*.

## Results

The clustering yielded some strange results. Normally, on a plot of *k* vs mean distance to clustroid, you expect to see an "elbow plot", where there's a marked decrease in mean distance between a particular couple *k*-values, and then a more gradual decrease afterwards, as *k* passes the point of particular poignancy and begins to overfit to the data. However, in this case, higher *k*-values often actually *increased* the mean distance to clustroid, as shown in the plots below:

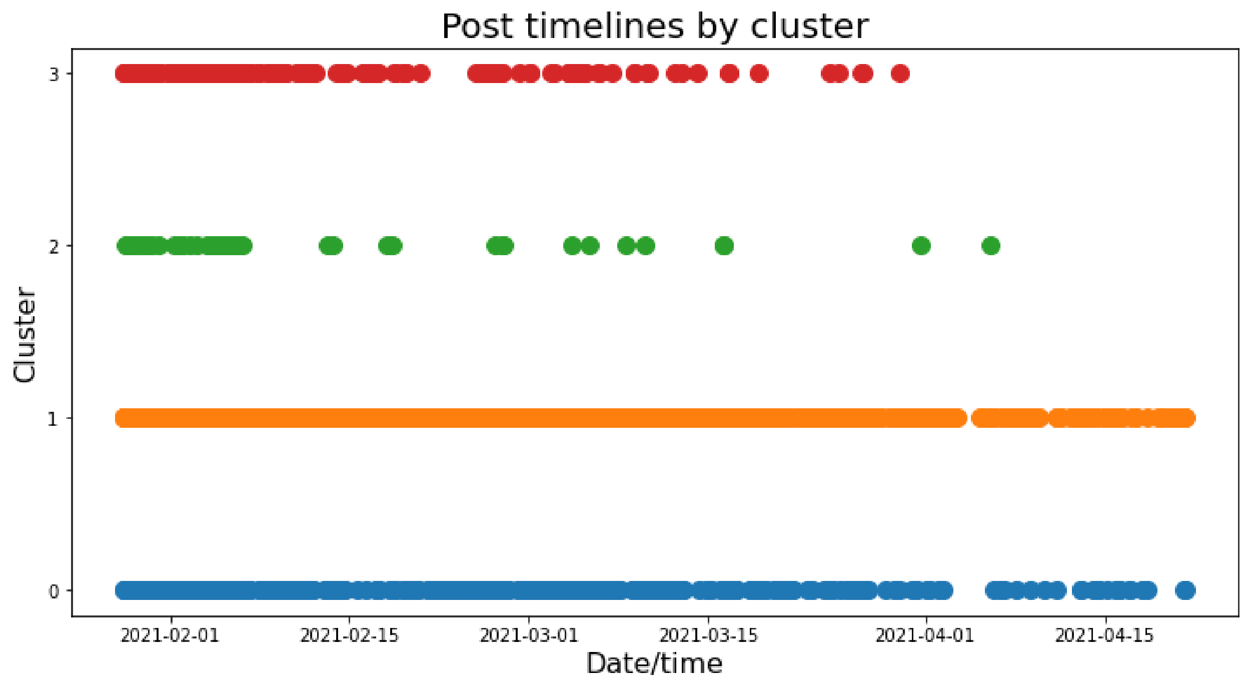






This could be for a number of reasons. First, the inaccuracy inherent in minhashing might have affected the data. Since minhashing is a method for approximating Jaccard similarity, there may be probabilistic noise in the similarities. While this is possible, it seems unlikely to be the main cause; I used a fairly standard number of hash functions, and this noise persisted through multiple trials with different random minhash initializations. The mean distances are very high, which suggests another possibility: the number of clusters might just not have been high enough to group the data well, meaning that an appropriate elbow plot with more *k*-values would show this to be just noise. This seems more likely; when we consider the 7000 documents in our dataset, even 20 clusters might be oversimplifying significantly.

Regardless, we did get some useful information about our dataset. The plot below shows the timelines of each cluster of the 1-shingled data for $k = 4$, the best-performing *k*-value. We can see that cluster 2 is concentrated around the beginning of February 2021, with some occasional outliers popping up afterwards. In contrast, cluster 1 seems to be much more regularly spaced & consistent posts. It turns out that the dataset contains several identical "daily discussion" posts, which all got grouped into cluster 1, accounting for its regularity. Cluster 2 is less immediately explicable; my tentative evaluation is that it seems to contain many more of the terms that characterize WSB discourse (the more repeatable of these being "tendies", "to the moon", "paper hands", etc.), while the other clusters are made up of posts with less catchphrase-y content.



## Conclusion

While there's a lot of fine-tuning & extra work that could be done on this model, I'm pretty pleased with the results. It's evident that there *are* patterns to be detected here, and this relatively simple model does a surprisingly decent job of it, despite the massive dimensionality of the text data, which would tend to make clustering difficult. The 1-shingle model seems to mostly detect differences in vocabulary (it's essentially a bag-of-words model); I chose to showcase it mostly because it was the most interpretable result, grouping posts by identical nature as well as topic.

The biggest changes I'd like to make are to train the model on the entire dataset, and to increase the range of *k*-values the model clusters on. Ideally, this would increase both clustering performance and generalizability. It would also be interesting to implement an upper threshold on similarity to filter out the identical posts. While identical posts are useful indicators for how a community functions, they serve mostly as noise when trying to identify conversation topic and sentiment; future models might simply not consider identical documents. Dimension reduction might also be interesting to explore; I barely even considered methods like PCA or SVD for the text data. It might even be possible to cluster on the reduced data alone, without minhashing, though I'd have to look into this further.

In terms of extensions to this model, other metrics could be considered. For example, posts could be weighted higher in the clustering based on the number of comments they had, or their score. Sentiment analysis by cluster is also a natural next step. Eventually, it might also be informative to cross-reference the results of this analysis with outside datasets - if a particular cluster shows high negative sentiment towards a particular stock, is this accompanied by a drop in that stock's value? What about negative sentiment towards certain hedge funds?

Overall, I consider this project as an excellent starting point in answering my question about temporal patterns in document clusters. Even just within these results, I get the sense that there's a depth of structure that remains to be explored. I'm encouraged by the usefulness of such a relatively simple model, and I'm confident that further experimentation & extension will provide even more insights into an intriguing problem.

## Works Cited

Becker, Hila & Naaman, Mor & Gravano, Luis. (2009). Event Identification in Social Media. http://web4.cs.columbia.edu/~gravano/Papers/2009/webdb09.pdf