David Tang
008282339
CMPE 255
Program 3 Report
Rank: 37
F1-Score: .1571

I started off by loading the data into a CSR matrix. Once that was done, I created 300 initial clusters by using **MiniBatchKMeans** from the **sklearn** library.

The **DBSCAN** algorithm variant I used involved modifying the existing code provided by **Activity-clustering-1**. Although I take in the initial K-means cluster as the input, I'm treating each cluster as a point by using each cluster's centroid.

I'm basically "refining" the initial clusters with **DBSCAN** by combining clusters with centroids that are closer to each other. My initial metric for "closeness" was Euclidean distance. The project specification mentioned that the CSR data provided is based on text records so I modified the DBSCAN algorithm to use cosine similarity. I precomputed the pairwise cosine similarities among the 300 clusters based on their centroids and then changed the line:

```
np.linalg.norm(points[i] - points[p]) <= eps
```

to

```
cos_sims[i][p] >= eps
```

Pseudo code for **DBSCAN** variant:

1. Construct pairwise cosine similarities from input cluster using each cluster's centroid
2. Determine core, border, and noise points using cosine similarity as a metric
3. Determine connected components in the graph of centroids
4. Assign each border point to connected component to which it is best connected
5. Return centroids in each connected component as a cluster

I failed miserably at graphing my computed clusters with varying **minPts** and **Eps** so I cannot show them here. I kept getting the following stacktrace.

```
TypeError                                 Traceback (most recent call last)
/anaconda3/lib/python3.7/site-packages/networkx/classes/graph.py in add_nodes_from(self,
nodes_for_adding, **attr)
    553         try:
--> 554             if n not in self._node:
    555                 self._adj[n] = self.adjlist_inner_dict_factory()

TypeError: unhashable type: 'set'
```

I verified that the line list(networkx.connected_components(graph)) creates a list of sets just like our activity. For whatever reason my clusters just won't plot.

I've tried playing a tad with dimensionality reduction using **SVD**. I used **SVD** immediately before creating the initial K-means clusters. The number of features I tried to reduce to was 500 and 1000. I didn't see any improvements with this change. In fact, my results got worse so I decided not to pursue this path.

When I was initially tweaking **minPts** and **eps**, I ended up with only 1 cluster because my **eps** was too small (using cosine similarity). When I set **eps** too big, **DBSCAN** returned the same initial 300 k-means cluster

When it came down to creating the file with my cluster results, I looped through each of the initial 300 K-means cluster and checked to see which refined cluster (the one generated after **DBSCAN**) it belonged to. If found, it will output that number to the file with a newline. If it is not found, then it was a noise point and got added into the **K+1** cluster where K is the number of clusters generated from **DBSCAN**.