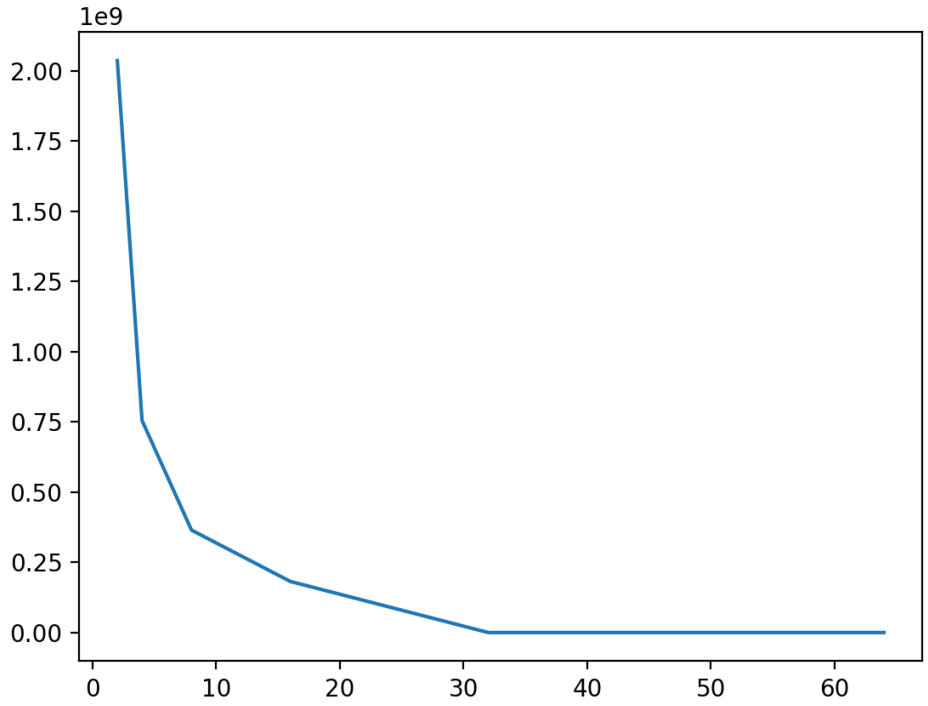
CS 373 Homework 2 (Using 2 late days)
By: Siddharth Shah shah255@purdue.edu

Assignment

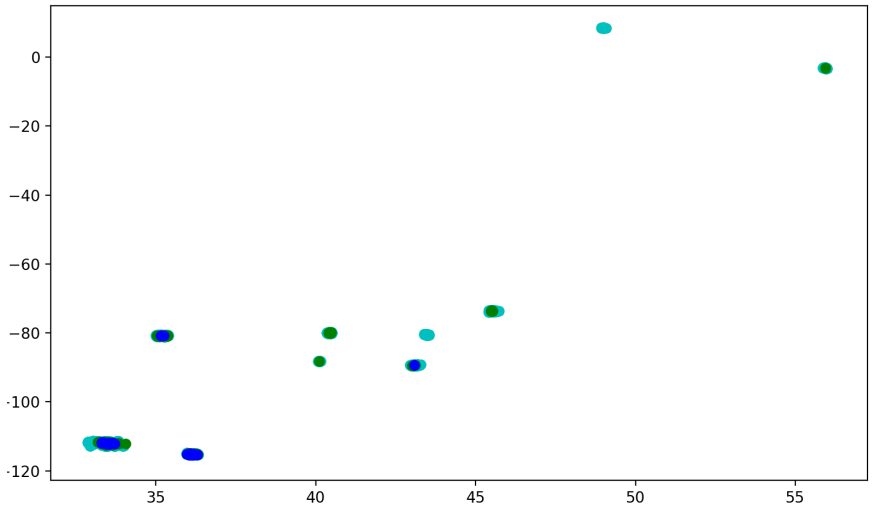
K-Means analysis

a)

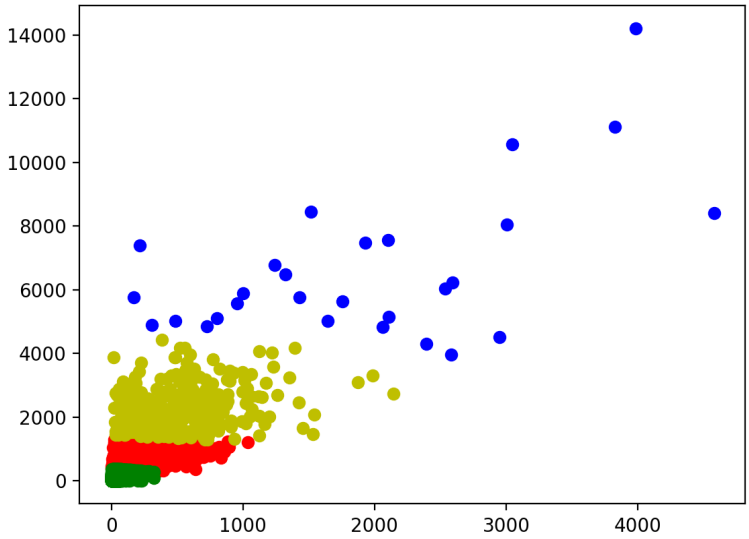


I would choose 8 or 16 as my 'k' value since after these values, depending on the data, I see that adding more clusters is just giving me diminishing returns to the expressivity of the new added cluster, as shown by the elbow plot.

b)



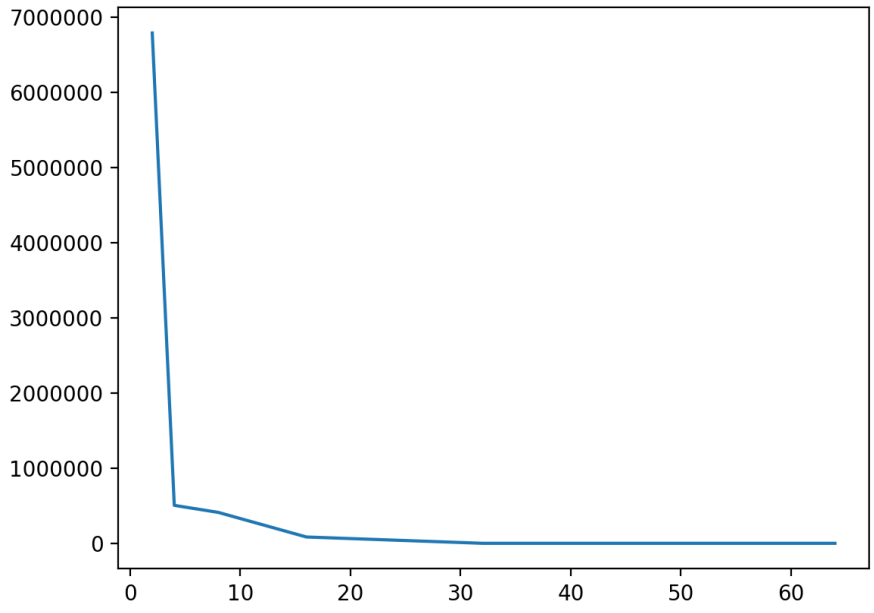
Latitude vs. Longitude for k = 4



ReviewCount vs. Checkins

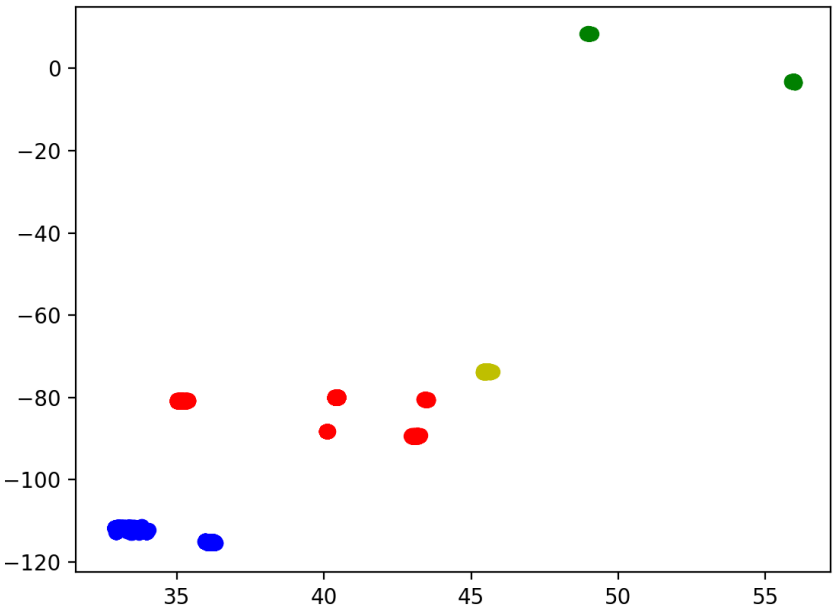
I think that the ReviewCount vs. Checkins features have more weight in deciding the clusters for the raw data. In the Latitude vs. Longitude plot, there aren't well defined clusters as seen in the plot, whereas the ReviewCount vs. Checkins seem to have relatively better distinct clusters.

c)

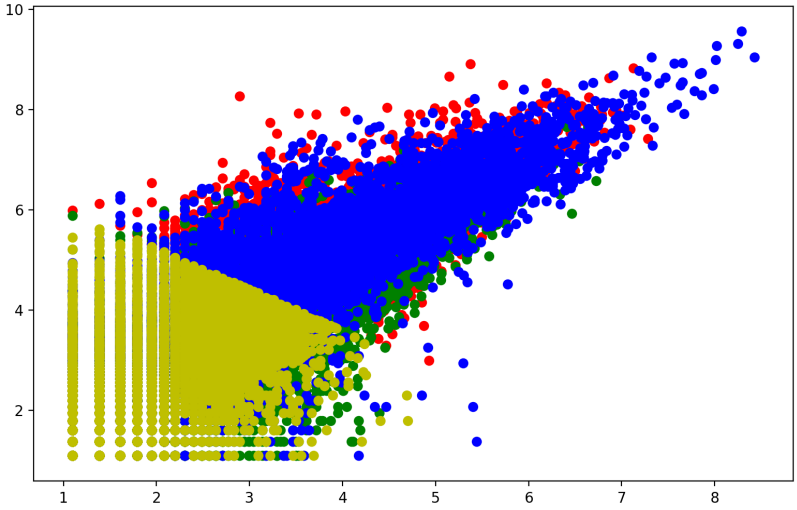


Elbow plot for log transform

I would choose 4 clusters since after 4, adding new clusters are only giving diminishing returns to every new added cluster.



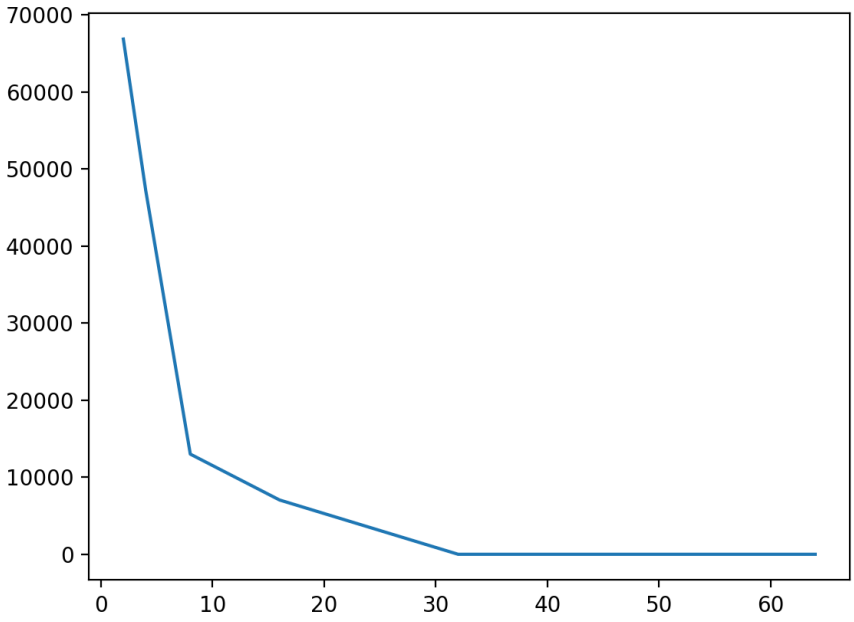
Latitude vs. Longitude



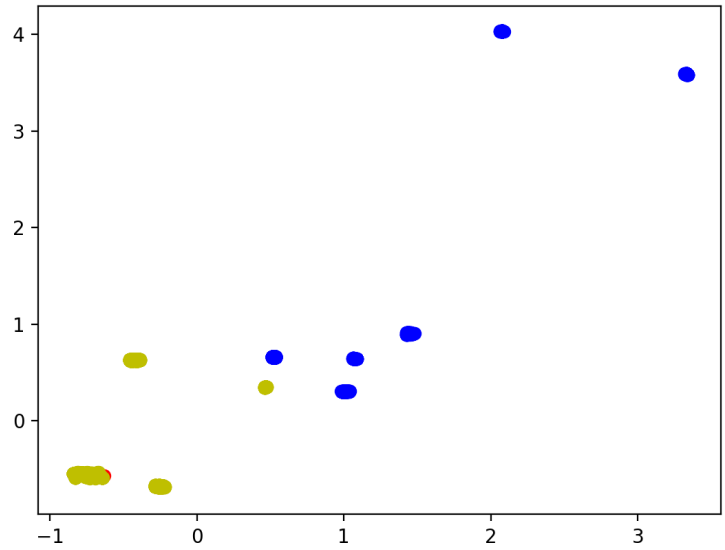
ReviewCount vs. Checkins (log transformed)

I think the log normalization causes the clustering to be weighted by the latitude longitude features as compared to ReviewCounts vs. Checkins since those features are normalized. The affect of normalization caused the values of ReviewCounts and Checkins to be closer to those of latitude and longitudes in just magnitude, which might have affected the degree of affect in clustering. We see merged clusters in the ReviewCount vs. Checkins plot and a more distinct separation in the latitude vs longitude plot.

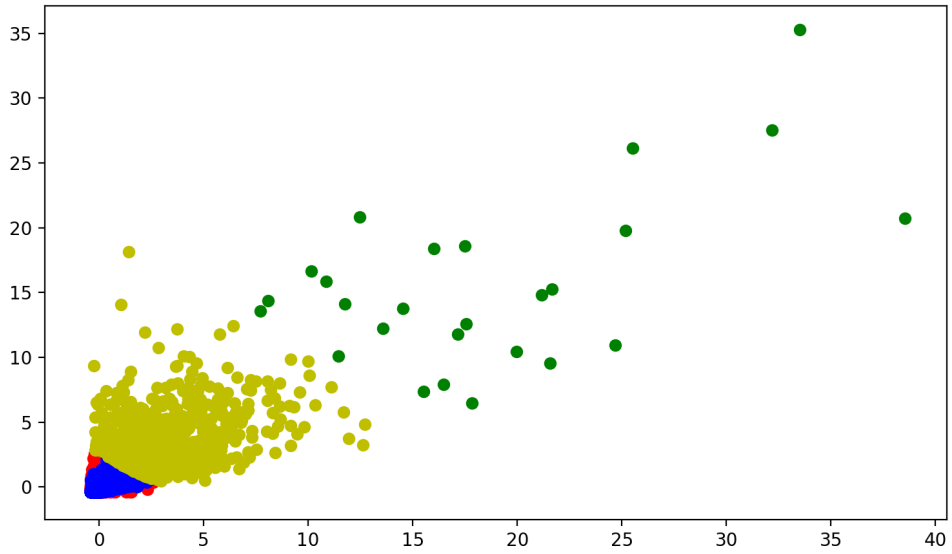
d)



I would choose 16 or even 8 as my number of clusters, since after that point, relatively new clusters provide only diminishing returns to the expressivity of the data via those new clusters.



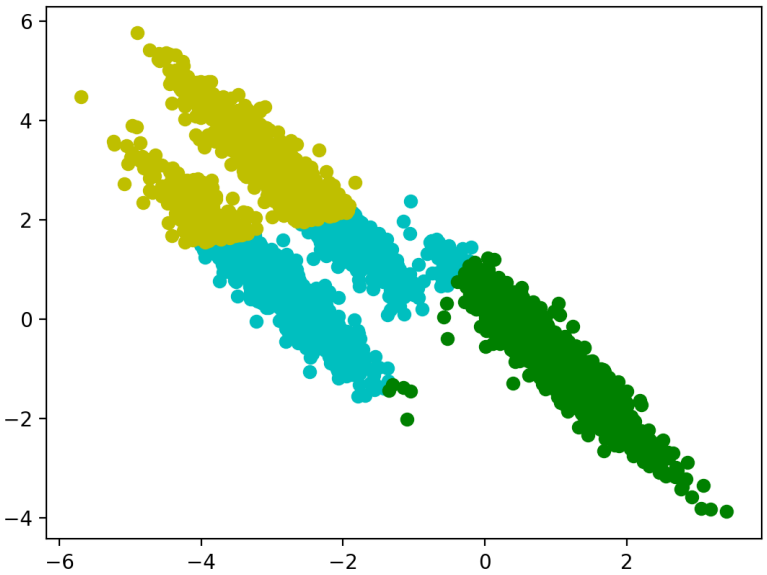
Latitude vs. Longitude



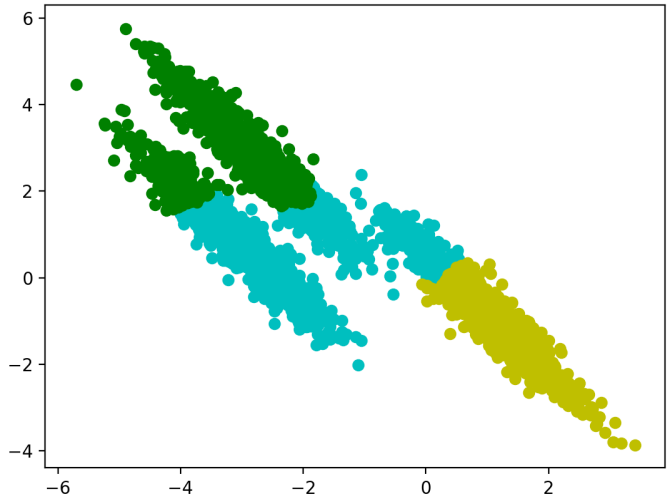
ReviewCount vs. Checkins

I think that for the scaled data, all 4 features are equally contributing to drive the clustering model, since for either plots, there aren't distinctly separable clusters, hence a collective 4D space is where the clusters are equally separable.

3) a)



KMeans for K = 3 on dummy.csv



Agglomerative Clustering for k = 3 and dummy.csv

I think that the Agglomerative Clustering works better on this data set, since the three clusters are pretty well separated. Especially the distinction is clear when we see the points at the bottom of cyan-cluster in graph for Agglomerative Clustering, those points are mis-clustered in by KMeans clustering into the green-cluster in KMeans graph.

b) KMeans does not yield the same answer every time, since the centroid initialization is random. The scores generated by KMeans is not always guaranteed to be globally optimal. This is due to the fact that randomization causes creation of local maxima.
Agglomerative clustering on the other hand, uses average distance between the points, hence there is no randomization. Hence, the assignments and scores are static every time the algorithm runs.

c) If both agglomerative clustering and kmeans are applied to the Yelp dataset, the Agglomerative Clustering will take more time, since it has stark difference in the time complexity. It's complexity is O(MN^2) as compared to the O(ndki) where n - number of points, k - number of clusters, i - number of iterations, d - number of features. Since for small 'K' values (as determined from the elbow plots) we can say that K-means will be faster than Agglomerative Clustering.