Akash Lankala
CS 373
Assignment 4
November 17th, 2019

Using 2 Late days for this homework

2 Kmeans

2.1 Theory

1.

Some limitations of using the k-means algorithm involve the need to choose k manually. Without sufficient information, determining the best value for k can be a challenge for certain datasets. Additionally, k-means will cluster data based on this value of k, so it is crucial for k to be close to the actual number of clusters in the data. Additionally, k-means can have issues clustering data where clusters are of varying sizes and density. Additionally, outliers can influence the centroids, and in some cases, get their own cluster instead of being ignored. There are a variety of solutions to these issues, mostly dependent on the dataset in question.

2.

You cannot always discover structure in data using the k-means algorithm. Based on the value you choose for k, the clusters may provide meaningful clusters or it may output clusters that either generalize too many datapoints into individual clusters or unnecessarily divide up individual clusters into multiple clusters due to a high value for k. Generally, if we know our data is binary, we know to set the value of k to 2. In this case, we can have high confidence that the k-means algorithm will cluster the binary data well into two separate clusters. However, this becomes more tricky as we have non-binary data with many labels, and as a result k-means may return clusters that have high variance and do not necessarily have strong structure. K-means works better with uniform data since we know that the clusters will each have about the same number of datapoints. As noted in Question 1, one limitation of k-means is with clustering data where clusters are of varying sizes and density – so uniform data would tend to output more meaningful clusters with better overall structure.

3.

The theoretical time complexity of k-means:
$O(i * k * n * d)$
Where,
i: number of iterations
k: number of clusters

Akash Lankala
CS 373
Assignment 4
November 17th, 2019

n: number of points
d: number of attributes

We can parallelize the algorithm by dividing the algorithm into sub-tasks that can be executed independent of each other without communication or shared resources.

One such way we can do this is by having multiple processors each running k-means with a different value for k. This way, we do not need to run the algorithm first with a value for k, and then *re-run* it with a different value of k, and then keep doing that until we get an output with low variance.
Another way we can parallelize is by instead of randomly choosing k points initially, we can have certain processors choose k points on certain chunks of the data. For example, if we have $k = 4$, we can have the four initial centroids be located on the four quadrants of a 2-dimensional dataset. If $k = 5$, the dataset could be split into 5 sections. These splits can then be executed on different processors in parallel which could reduce the overall time complexity by coming with an optimal output sooner.

4.
Approach A uses NBC on the p discrete features, which would then yield a 0/1 loss score of say, X.
Approach B uses both NBC *and* k-means on discrete and continuous features respectively. As a result, approach B would provide *more information gain* than approach A as it is able to cluster continuous features *in combination* with predicting discrete features that would yield an overall more confident prediction in the binary class label since we have more information to base the prediction on.
Since the binary class label is discrete (only has two values), the number of instances of "1"s (or whatever score we assign to misclassifications) would be lower in approach B as compared to approach A since we would expect the added k-means classification to provide more accuracy in our class label prediction.

Akash Lankala
CS 373
Assignment 4
November 17th, 2019

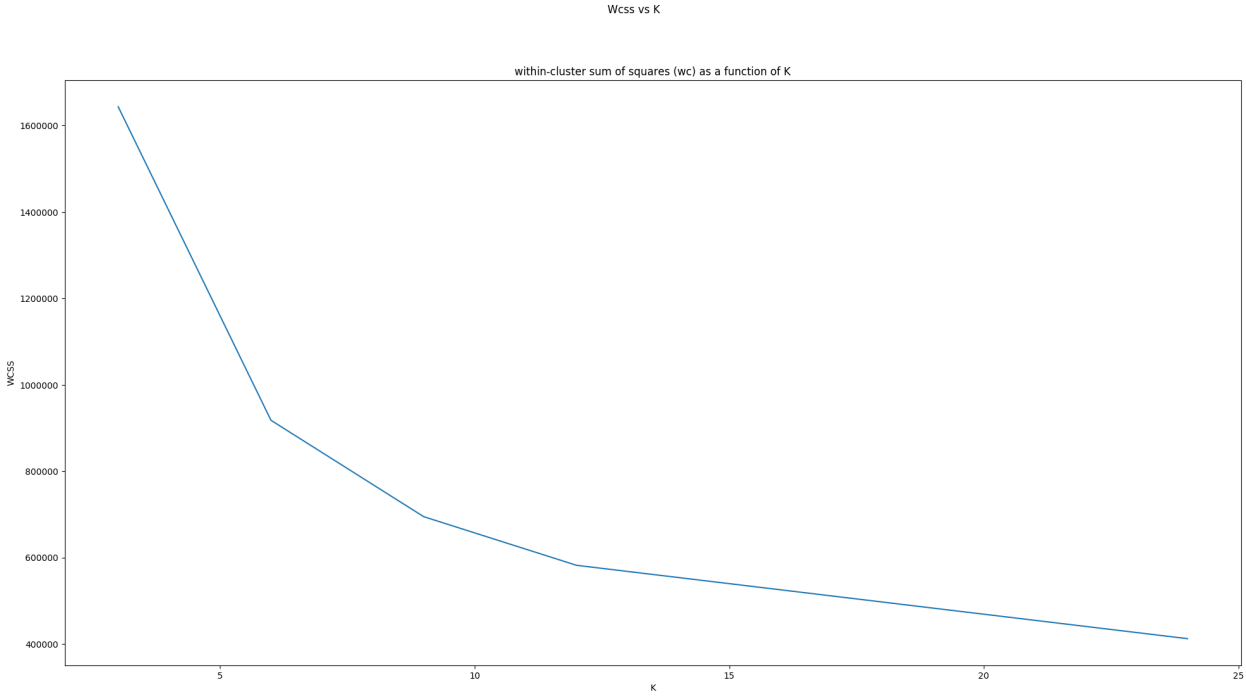
2.2 Implementation

See kmeans.py

3 Analysis

1. Cluster the Yelp data using *k*-means

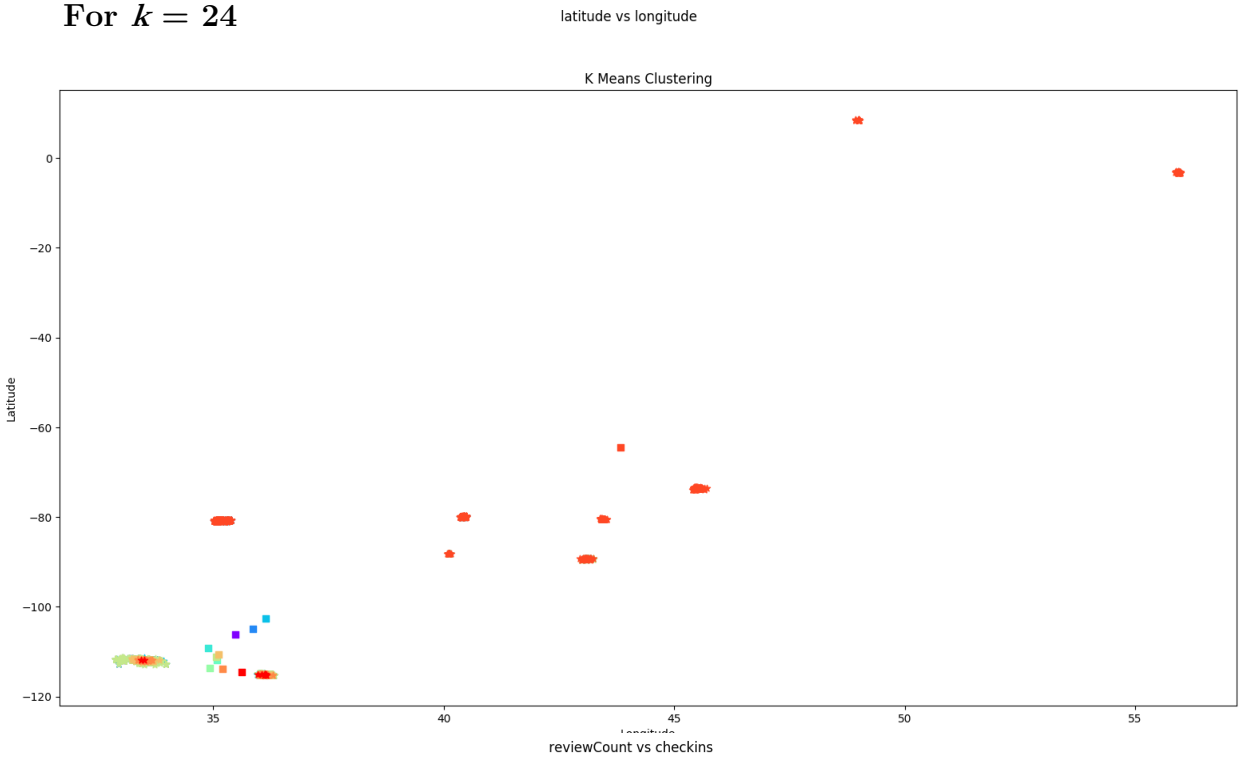
K	WC
3	2444345
6	1516267
9	1138508
12	924437
24	692845

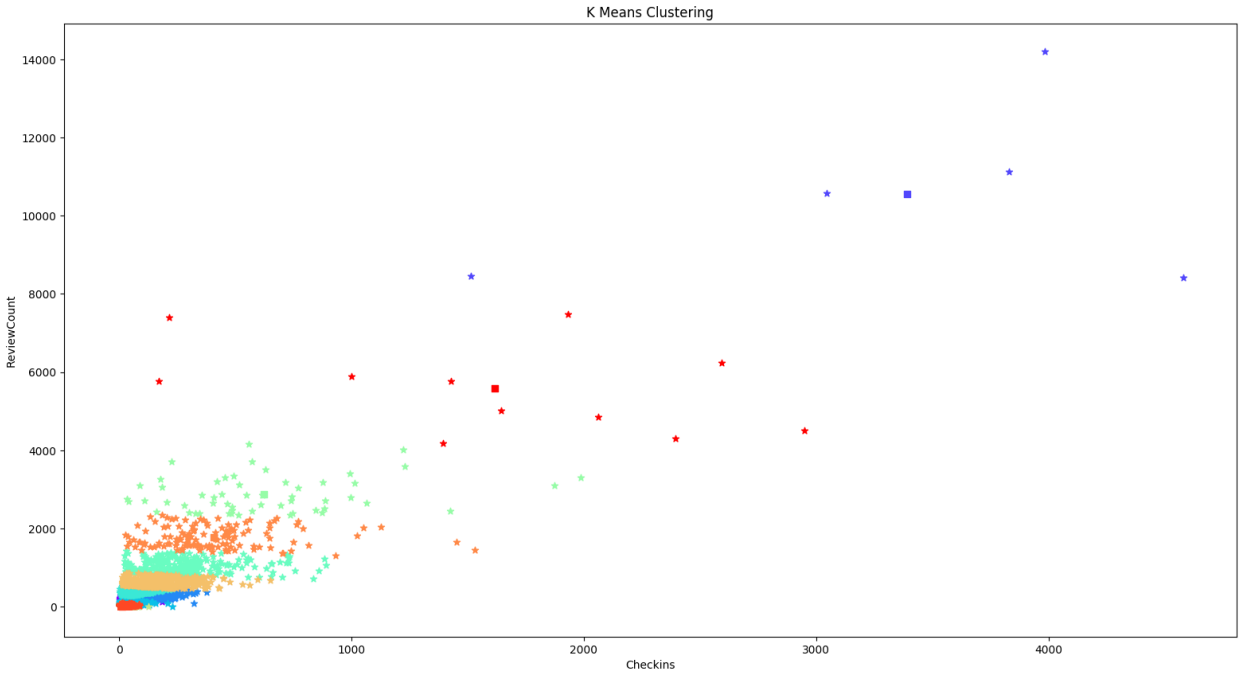


From the graph it can be observed that wc(k) is a deceasing function.
Therefore, $\boldsymbol{k = 24}$ gives the least squared error.

Akash Lankala
CS 373
Assignment 4
November 17th, 2019

For $k = 24$





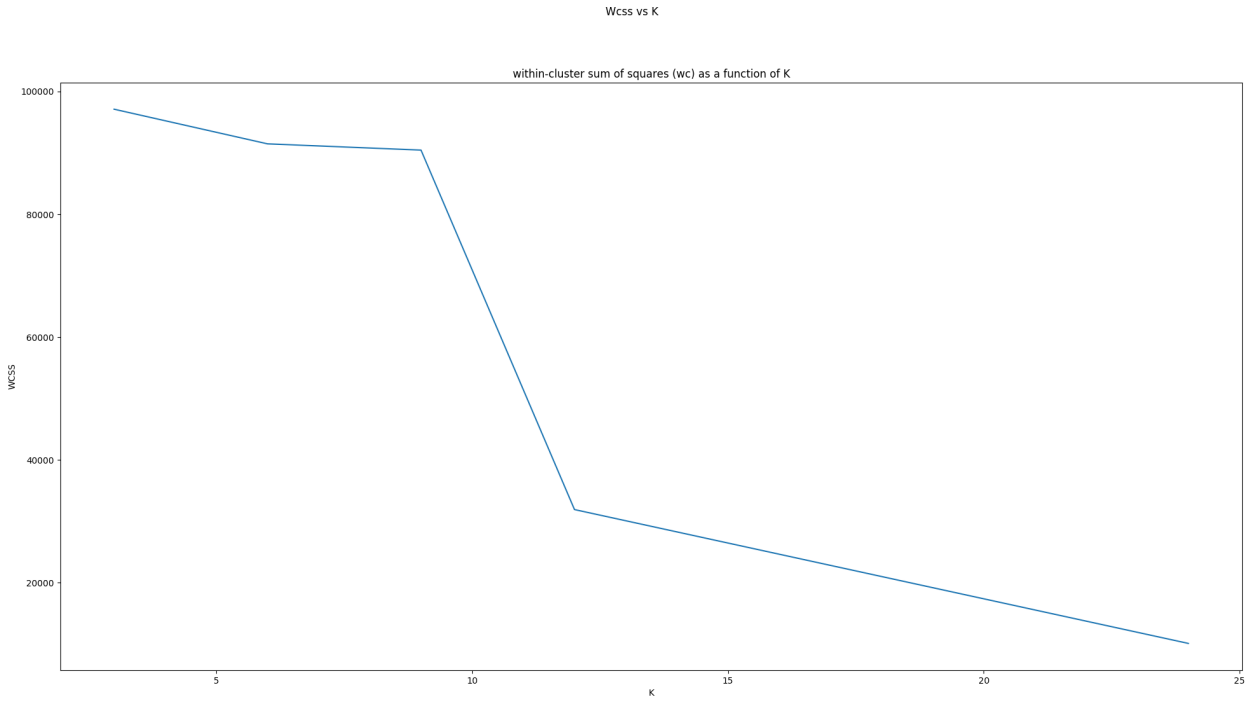
From the above two graphs, it can be inferred that *reviewCount* and *checkins* are the prevailing features because there is a clear boundary between clusters in the second graph.

Akash Lankala
CS 373
Assignment 4
November 17th, 2019

2. Log transformation applied to *reviewCount*, *checkins*

It can easily be observed that *reviewCount* and *checkins* have large variances. For two data points with close longitude and latitude, there may be large differences in the *reviewCount* and *checkins* attributes, and thus they may be placed into two different clusters. The *wc* is also high due to this large variation. By applying the log transformation to the *reviewCount* and *checkins* features in the dataset, the net distance between the two data points along these two axis is reduced drastically. Therefore, the clustering depends more on longitude and latitude and less on *reviewCount* and *checkins*.

k	***wc***
3	165320
6	40795
9	33766
12	23064
24	16729
48	11796

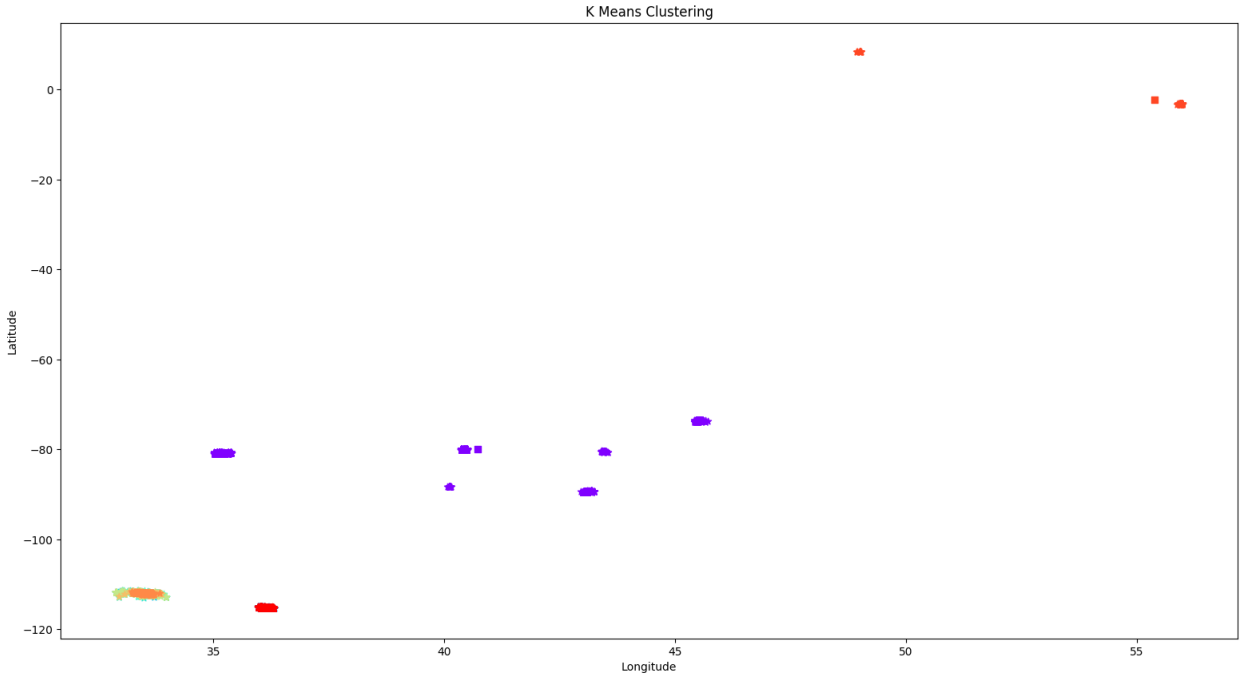


wc values are smaller than the previous case. Again, it is observed that $k = 24$ leads to smallest squared errors.

Akash Lankala
CS 373
Assignment 4
November 17th, 2019

latitude vs longitude

For $k = 24$



reviewCount vs checkins

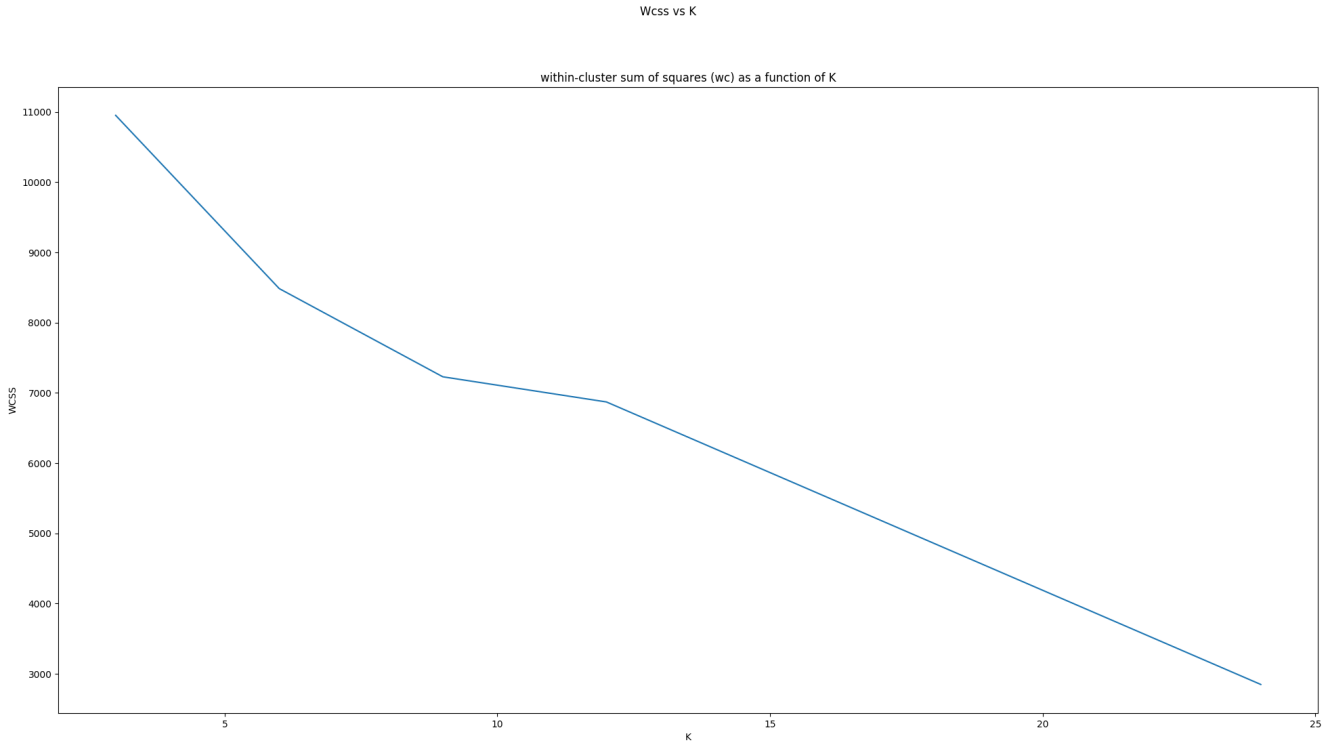
Akash Lankala
CS 373
Assignment 4
November 17th, 2019

From the above graphs it is evident that unlike previously, longitude and latitude are now the dominant features and data points with close longitude and latitude values are clustered together.

3. Standardization of values

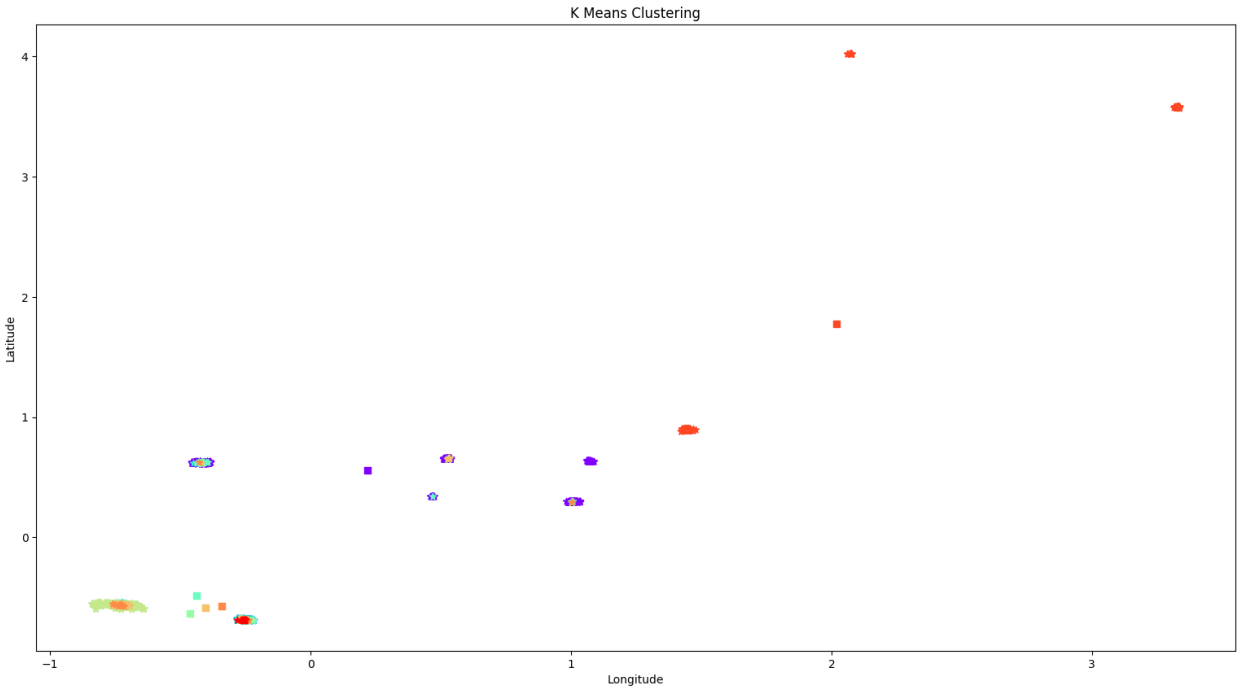
Standardization of values ensures that all features have the same scale. Thus, there is no one dominant feature.

K	wc
3	21161
6	11952
9	8898
12	7957
24	3491

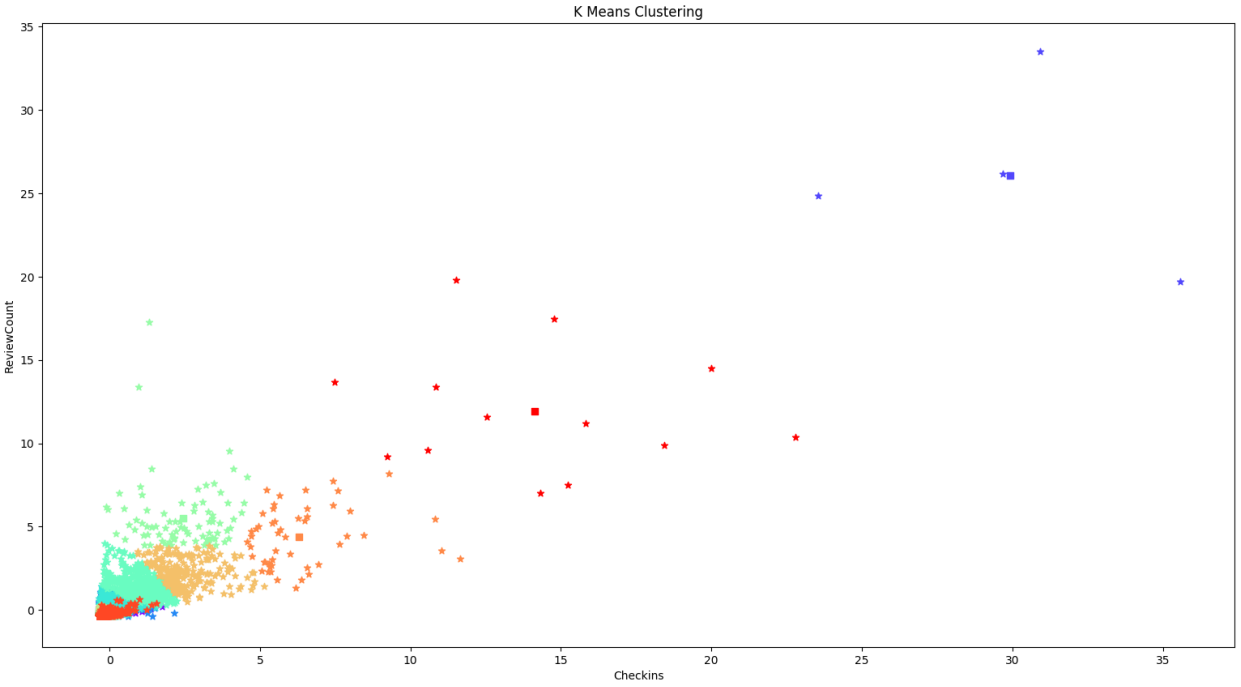


Akash Lankala
CS 373
Assignment 4

latitude vs longitude



reviewCount vs checkins



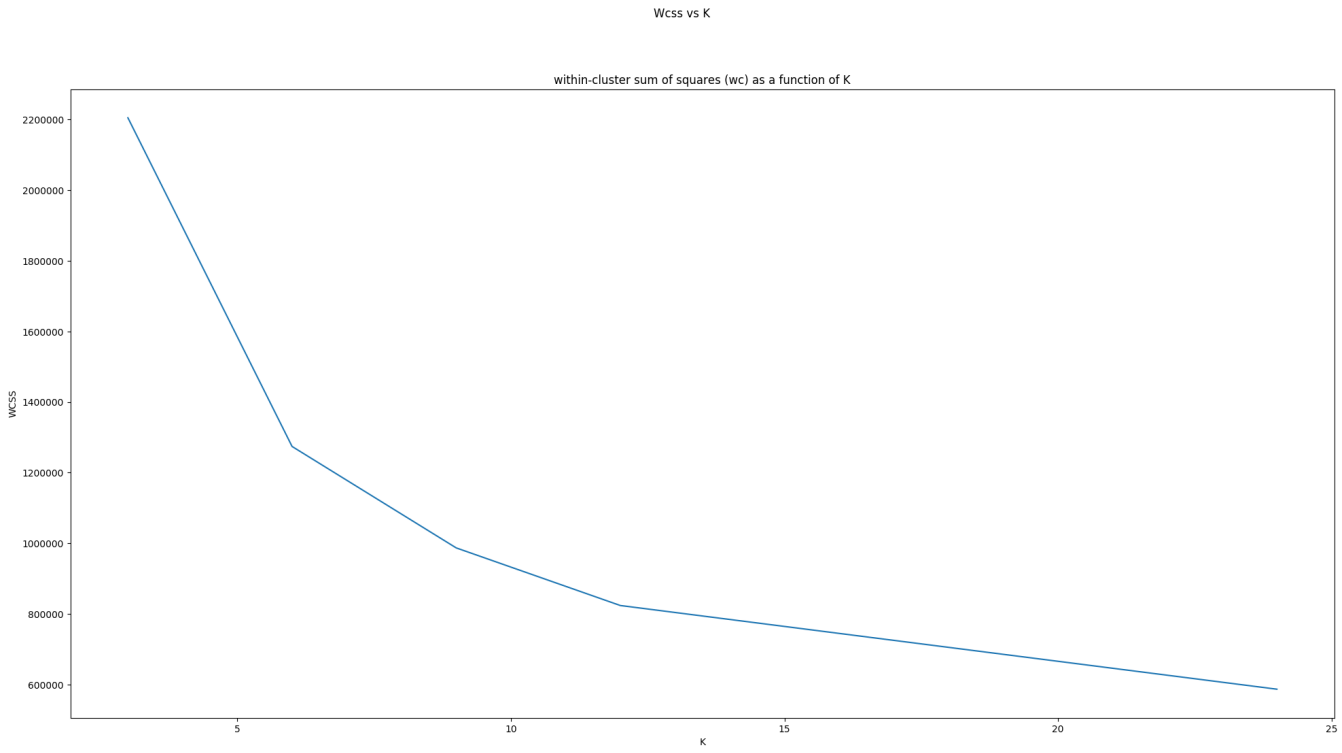
From the above graphs, it can be seen that there is a clear cluster boundary in both of the graphs, which is expected from the transformation change to the clustering results.

4. Manhattan distance

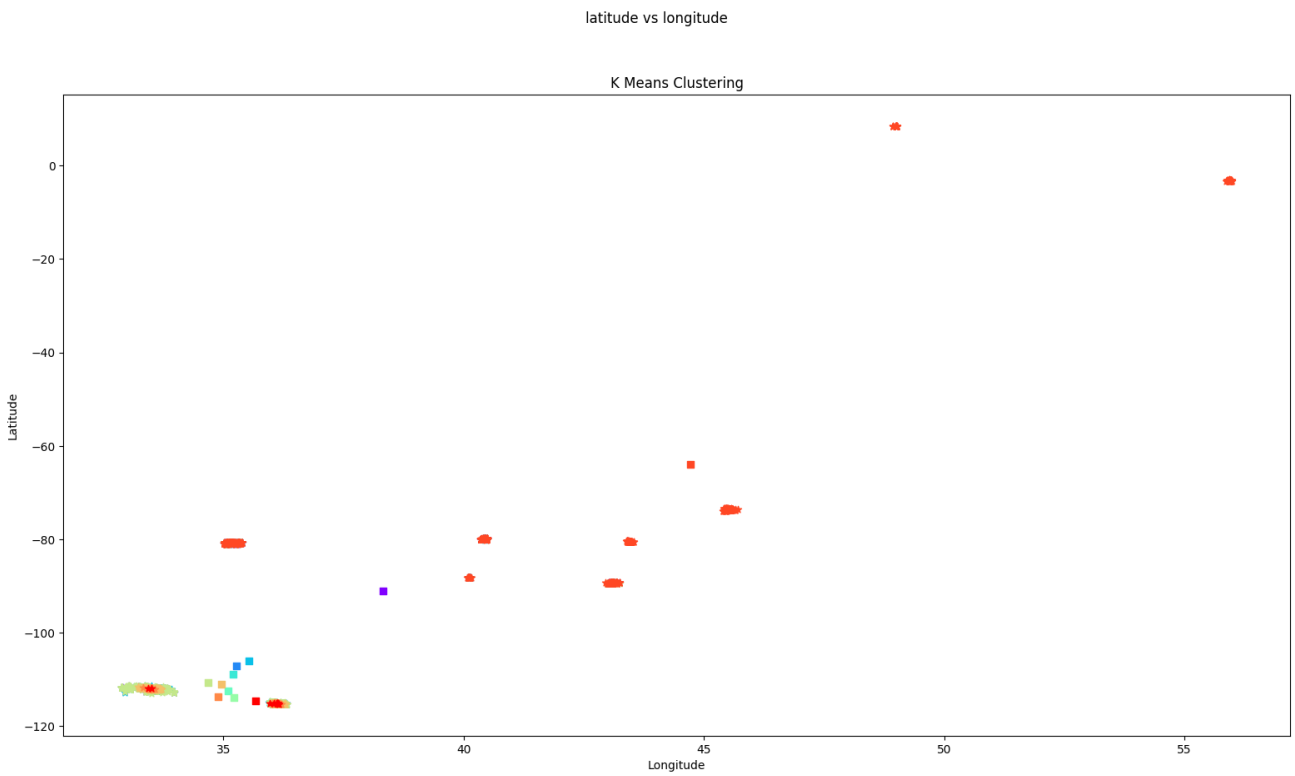
Using Manhattan distance for clustering is not recommended because it comes with some distortion. *Wc* is also higher than that of *wc* for the Euclidean distance case. Also the number of iterations required to arrive at the final centroids is large, but the results are usually similar to the Euclidean distance results.

K	Wc
3	3417126
6	2197246
9	1660217
12	1326803
24	971635

For $k = 24$



Akash Lankala
CS 373
Assignment 4
November 17th, 2019



The above two graphs are very similar to the graphs of case 1. This similarity is that there is a clear boundary between the clusters.

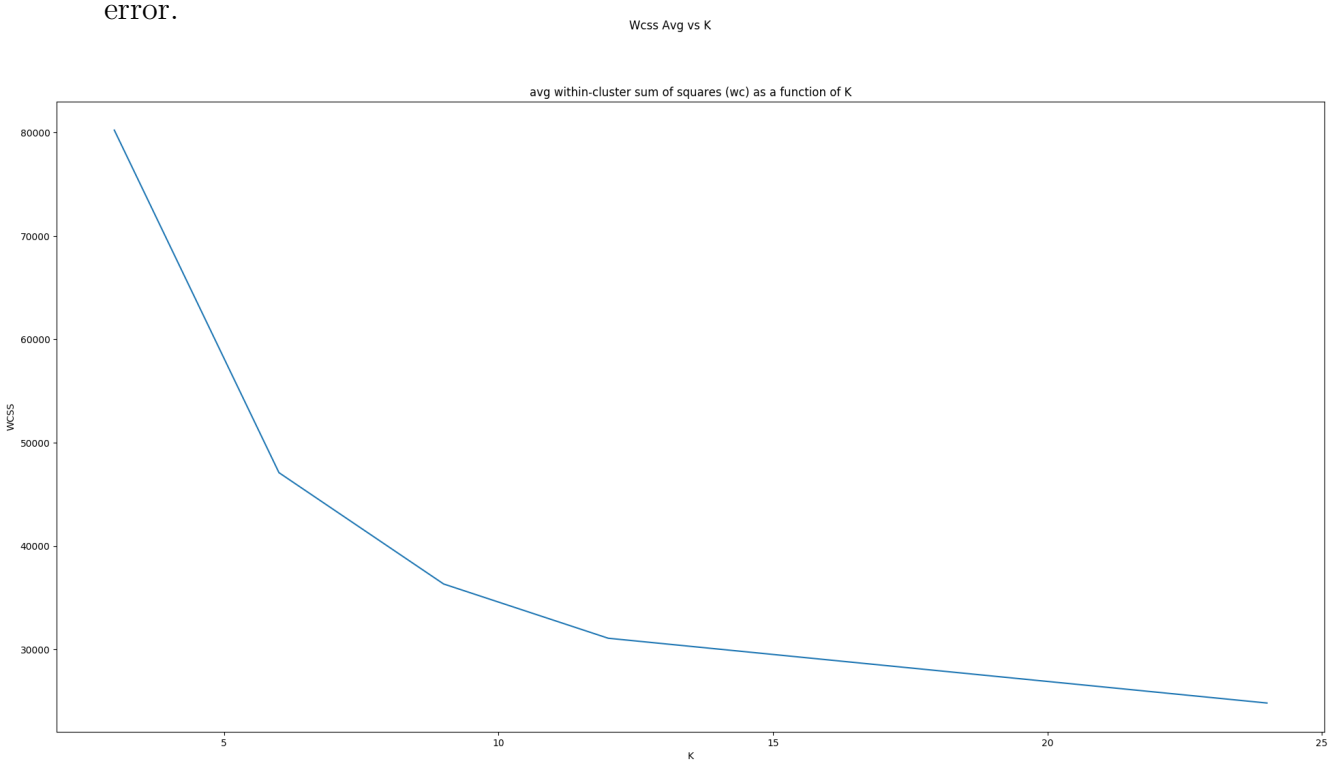
5. Downsampling

Downsampling reduces the execution time of k-means clustering. k-means clustering has the time complexity of $O(n)$, where n is dataset size. Stability of the k-means algorithm also increases due to downsampling. By downsampling, the optimal centroids can be found within the max iterations limits – which may not be the case for large datasets with scattered points.

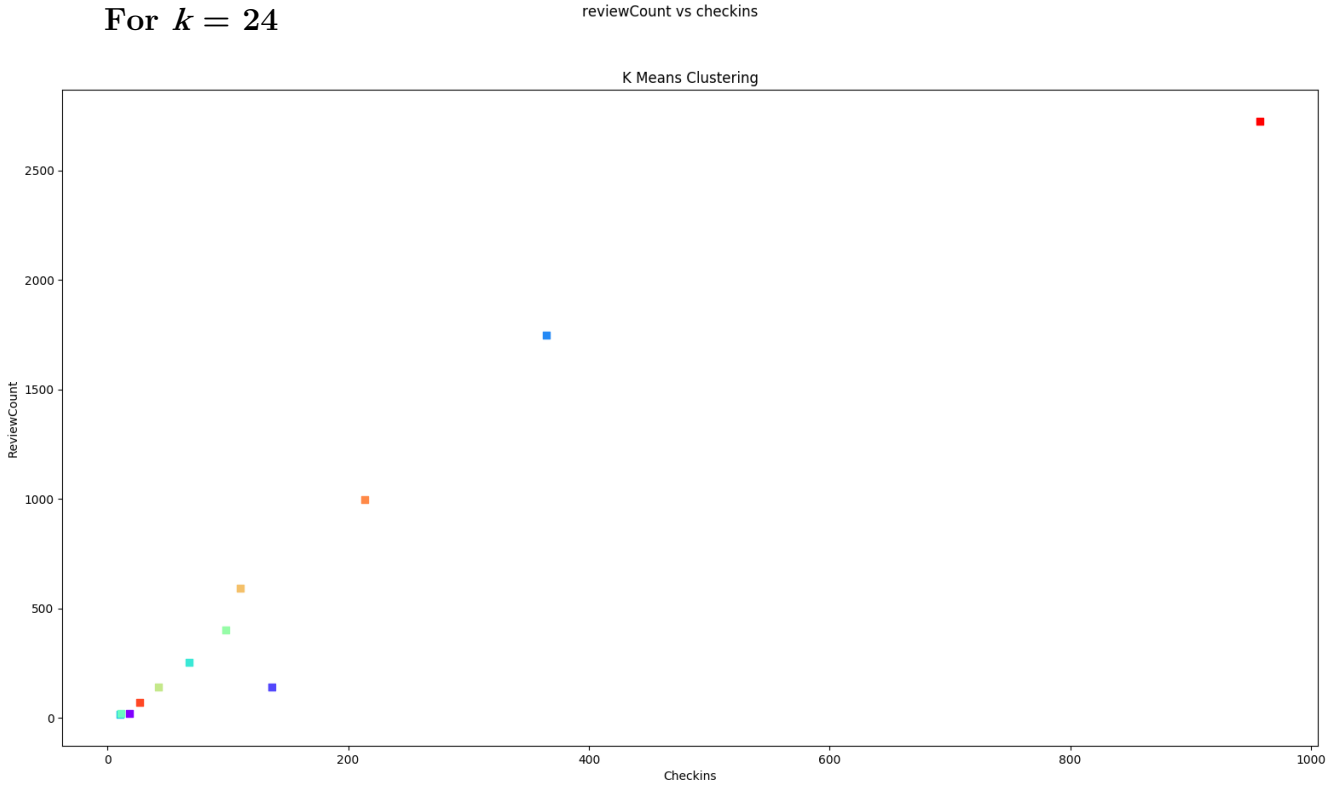
K	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Average
3	77285	77285	77285	77285	77285	77285
6	51019	45382	45382	45382	51019	47636
9	36963	36963	35926	36963	36480	36659
12	30812	30157	30143	30845	30759	30543
24	26514	25441	25399	25533	25634	25704

Akash Lankala
CS 373
Assignment 4
November 17th, 2019

Trial 4 has *wc* values closest to the *wc* average, and K=48 has the lowest squared error.



For $k = 24$



From the above two graphs it can be inferred that *reviewCount* and *checkins* are the dominant attributes because there is a clear boundary between clusters in second graph. Since k value is high and dataset is small, some clusters just have a single element.

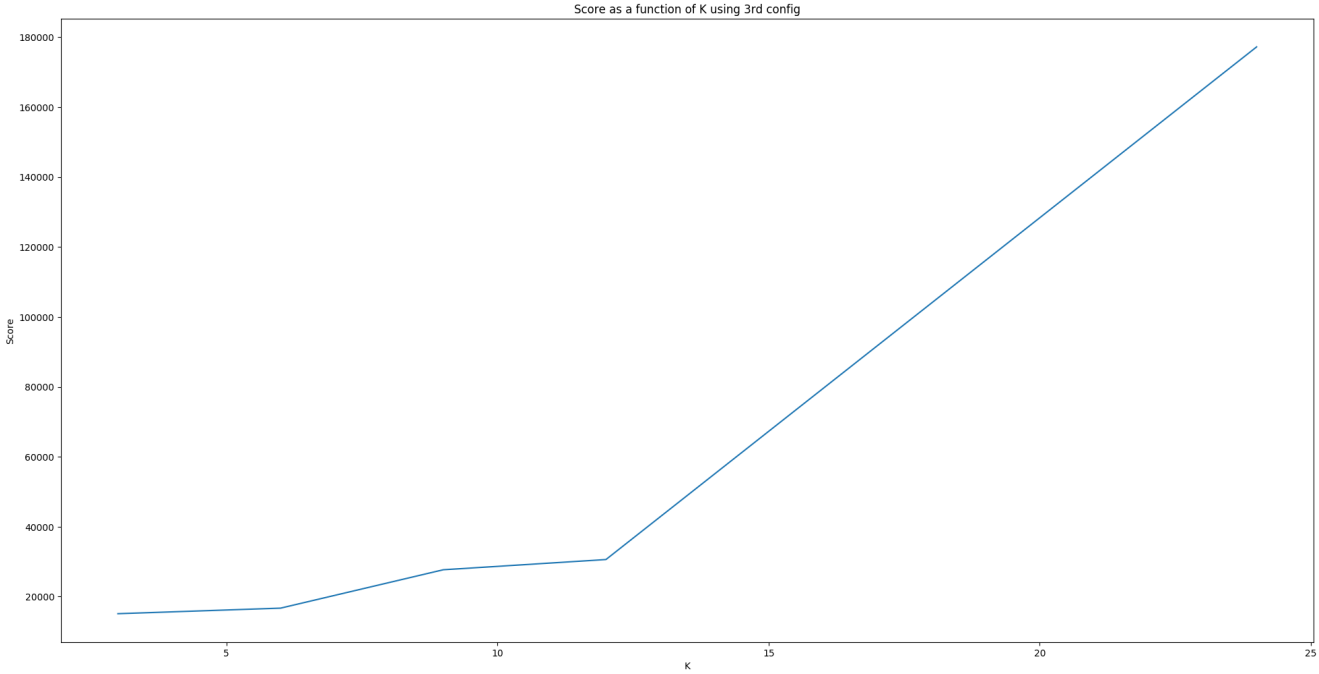
6. Improved score function.

A better performance score function can be the sum of ratio of distance of a data point to its centroid to its distance to the nearest centroid for all data points. This score function is a much better metric than *wc* because it gives the measure of spread (for example, how far clusters are spread apart).

X - Data Set C - Centroids $S(X, C) = \Sigma_x\ d(x, c(x))\ /\ min(d(x, C - c(x)))$

K	S
3	8726
6	122985
9	208016
12	360870
24	511224

Score vs K



Akash Lankala
CS 373
Assignment 4
November 17th, 2019

For $k = 3$, clusters are farthest apart from each other, and compared to Q1, the clusters are more separated. This is expected.