Hw5

Cr statement: Linbin Sun
trainingDataFileName:corresponds to a subset of the data that should be used as the training set for your algorithm

K: the value of k to use when clustering

Clustering option: takes one of the following five values,
1 (use the four original attributes for clustering, which corresponds to Q3.1)
2(apply a log transform to reviewCount and checkins, which corresponds to Q3.2)
3(use the standardized four attributes for clustering, which corresponds to Q3.3)
4(use the four original attributes and Manhattan distance for clustering, which corresponds to Q3.4)
5(use 3% random sample of data for clustering, which corresponds to 3.5)

Your code should read in the training sets from the csv file, cluster the training set using the specified value of k, and output the within-cluster sum of squared error and cluster centroids. For the centroid of each cluster root the values for each of the four attributes in the following order

Attitude, longtitude, reviewCount, checking

The expected output is given below. Note that this yelp.csv, k=4, cluster option 1

$ python kmeans.py yelp.csv 4 1
WC-SSE=15.2179
Centroid1=[49.00895,8.39655,12,3]
…
CentroidK=[33.33548605,-11.7714182,9,97]

2 Means
2.1 theory: what are the benefits of the k-means clustering algorithm?

What are the benefits of the k-means clustering algorithm? What are the issues? In which situations should we use k-means? Your answer should compare to other algorithms learned in class (both unsupervised and supervised) in less than 4 sentences.

K means is scalable but cannot use for flexible data. K means is very useful of large data sets and only applicable on only numeric data.For large datasets, you must not use Hierarchical clustering. Hierarchical clustering is generally applicable to a small set of data. The k-mean does not have the label while the supervised learnings have labels

2,2 implementation

3 Analysis

3.1(10 points) Cluster the Yelp data using k-means.

- (a) Use a random set of examples as the initial centroids.
- (b) Use values of K = [3,6,9,12,24].
- (c) Plot the within-cluster sum of squares (wc) as a function of K.
- (d) Choose an appropriate K from the plot and argue why you choose this particular K.
- (e) For the chosen value of K, plot the clusters with their centroids in two ways: first using latitude vs. longitude and second using reviewCount, checkins. Discuss whether any patterns are visible.
- 3.2(10 points) Do a log transform of reviewCount, checkins. Describe how you expect the transformation to change the clustering results. Then repeat the analysis (1). Discuss any differences in the results.

(10 points) Transform the four original attributes so that each attribute has mean = 0 and stdev = 1. You can do this with the numpy functions, numpy.mean() and numpy.std() (i.e., subtract mean, divide by stdev). Describe how you expect the transformation to change the clustering results. Then repeat the analysis (1). Discuss any differences in the results.

(10 points) Use Manhattan distance instead of Euclidean distance in the algorithm. Describe how you expect the change in the clustering results. Then repeat the analysis (1). Discuss any differences in the results.

(10 points) Take a 6% random sample of the data. Describe how you expect the downsampling to change the clustering results. Then run the analysis (i) five times and report the average performance. Specially, you should use a single random 6% sample

of the data. Then run 5 trials where you start k-means from different random choices of the initial centroids. Report the average wc when you plot wc vs. K. For your chosen K, determine which trial had performance closest to the reported average. Plot the centroids from that trial. Discuss any differences in the results and comment on the variability you observe.

I would expect similar outcome since the data are drawn from the same distribution

(10 points) Improve the score function. To evaluate the clustering, it is not sufficient to measure only the within-cluster sum of squares (wc) that you used above. It is also desired to have each cluster separate from others as much as possible. To improve the resulting clustering, define your own score function that takes into account not only the compactness of the clusters but also the separation of the clusters. Write a formal mathematical expression of your score function and explain why you think your score function is better than the within-cluster sum of squares. Also, using the best configuration from Questions 1-5, plot the results of your score function for K = [3, 6, 9, 12, 24], and compare the results to the appropriate algorithm from Question 1-5.

To show the consideration of distance between different groups I would penalize the model if the cluster points are too close.with a penalty of p