Homework 2

CSCI 4907/6444: Big Data and Analytics

Fall 2023

The deadline for homework assignment 2 is **10/10/2023 11:59 pm**. The total number of points for this assignment is 12. Make sure that you upload a *single* zip file for all of the questions in the assignment on blackboard. Create two folders `q1` and `q2` and put all the related content (code, output files, writeups, etc.) in the corresponding folder. Do not include the input data in your submissions. Name you folder `lastname_firstname_hw2` and compress it to create a zip file. Upload your zip file to blackboard.

Academic Integrity

Honor Code: Students may discuss the solutions at a high level with one another. Note that each student must write down their solutions/code independently to show they understand the solution well enough in order to reconstruct it by themselves. Students **should clearly mention the names of all the other students who were part of their discussion group and specify each person's contribution.**

Important: Using code or solutions obtained from the web is considered an honor code violation. You may use printed or online sources for understanding how to use Python functions but you may not search for code or theoretical solutions. We check all the submissions for plagiarism. Please include the following text excerpt in your submission zip file (in the main folder, next to folders `q1` and `q2`):

```
I [insert your name here], affirm that this is my own work,
I attributed where I used the work of others, I did not fa-
cilitate academic dishonesty for myself or others, and I used
only authorized resources for this assignment, per the GW Code
of Academic Integrity.  If I failed to comply with this sta-
tement, I understand consequences will follow my actions.  Con-
sequences may range from receiving a zero on this assignment
to expulsion from the university and may include a transcript
notation.
```

1 Dimensionality Reduction, Clustering and Visualization (5 pts)

In Lecture 1, we saw an image of lego parts. Initially, the lego parts were messy and scattered around. Then, the parts were sorted, arranged, presented visually, and explained with a story. As a result of this process, messy lego parts were transformed into a beautiful house.

In this problem, the goal is for you to replicate that process for the given dataset. You will cleanse and pre-process the data, use techniques such as dimensionality reduction and clustering to analyze the data, and you will use visualizations along with your contextual knowledge of the dataset to offer insights.

Depending on the pre-processing strategy, the answers to the questions below may slightly differ from person to person. Our focus is more on your strategy, implementation, depth of analysis and interpenetration of results rather than checking for exact values. This problem is meant to encourage you to familiarize yourself with exploratory data analysis (EDA) and learn how to read Python documentations and find relevant functions.

Dataset: You are given the *California housing* dataset, which contains housing information in different districts of California, as well as the corresponding median price of houses in each district. Each row in the dataset corresponds to a district, so the features provided for the districts provides a summary of all houses in that district based on the 1990 census data. The data is in its raw format and has not been pre-processed. It is your job to cleanse and standardize the data to extract more meaningful insights.

Task: You need to preform the following tasks:

- Understand and explore the data using Python functionalities
- Pre-process the data
- Analyze the data
- Visualize the data
- Tell the story of the data

You are given a jupyter notebook with a code template for this problem. You can run the code locally on your computer or upload it to Google Colab if you prefer that. Note that you need to add textual description and comments to your jupyter notebook. Imagine the notebook is a report and you want to populate this report with as many insightful information as possible. **At the very least, you should answer the questions outlined below.** When you upload your jupyter notebook, you must include the output for each cell (do not clear the output). There is no need to use Spark for this question.

Data Discovery and Dimensionality Reduction [2.5 pts]

(a)[1 pts] Before performing any dimensionality reduction, are there any features that you think will help the most with clustering? why? (use visualizations

and computations to answer this question. You should answer based on evidence and argue your reasoning.)

Hint: plot histograms for each feature and produce pairwise plots of features.

(b)[0.5 pts] Perform PCA dimensionality reduction on the data. What number of PCA components do you think can help us best visualize our data? why?

(c)[0.5 pts] Produce both 2d and 3d scatterplots of the data. Do you see any outliers? What rule/filter/threshold can you use to remove the outliers?

(d)[0.5 pts] Remove any large outliers and confirm that the outliers have been removed by retraining and re-plotting both 2d and 3d scatterplots for the filtered data. Include your new plot for the filtered data in the notebook.

Clustering [2.5 pts]

We would like to see if the houses can be grouped into distinct clusters based on the given features or not. Use k-means clustering to cluster the houses and use the elbow method to determine the optimal number of clusters. Implement/Answer the following questions in the notebook:

(a)[0.75 pts] Create an elbow plot (within-cluster sum of squares in the y-axis and number of clusters inn the x axis). Based on this plot, what is the optimal number of clusters? why?

(b)[0.5 pts] Silhouette score is another metric that can help with assessing the quality of clustering. Why does this metric help with determining the number of clusters? Explain.

(c)[0.75 pts] Plot a visualization with number of clusters in the x-axis and silhouette score as the y-axis. What is the optimal number of clusters according to the plot? why?

(d)[0.5 pts] Let k be the optimal number of of clusters (from either part a or c). For i=2,3,...k, run k-means algorithm using i as the number of clusters. Plot the data in 3d scatter-plot for each i and use different colors for each cluster to distinguish them in the plot. Inspect the clusters that were formed in each step. Based on the visuals, do you agree that the optimal number of clusters is the k value you previously determined? why or why not?

What to submit

You should upload your jupyter notebook for this file that contains answers to all of the questions above (plus any extra analysis that you have done). No need for a separate write-up.

2 k-means Implementation (7 points)

Note: Students who are enrolled as CSCI 6444 students must implement their solution to this question using Spark. However, you can still get half the credit for each question if you use regular Python programming (3.5 points).

CSCI 4907 students can implement the solution using regular Python programming, and can get up to 3.5 extra credit points if they implement it with Spark.

Note that you will not receive any credit if your answers to the questions are correct, but somehow your implementation does not produce those answers. You may use the code template provided in jupyter notebooks as a starting point. You can upload these notebooks to Google colab environment and run your code from there.

Implementation: You may not use k-means/k-median functions from libraries such as sklearn or Spark MLlib. The goal of this exercise is to make you implement k-means/k-medians functions from scratch. In the Spark implementation, you may store the centroids in memory if you choose to do so.

This problem will help you understand the nitty gritty details of implementing k-means clustering algorithms. In addition, it will also help you understand the impact of using various distance metrics and initialization strategies in practice. Let us say we have a set X of n data points in the d-dimensional space $\mathbb{R}^{\mathrm{d}}$. Given the number of clusters k and the set of k centroids C, we now proceed to define various distance metrics and the corresponding cost functions that they minimize.

Euclidean distance Given two points A and B in d dimensional space such that $A = [\, a_1, a_2, ...a_d]$ and $B = [\, b_1, b_2, ...b_d]$, the Euclidean distance between A and B is defined as:

$$\|a - b\| = \sqrt{\sum_{i=1}^{d} (a_i - b_i)^2} \tag{1}$$

The corresponding cost function ϕ that is minimized when we assign points to clusters using the Euclidean distance metric is given by:

$$\phi = \sum_{x \in X} \min_{c \in C} \|x - c\|^2 \tag{2}$$

Note, that in the cost function the distance value is squared. This is intentional, as it is the squared Euclidean distance the algorithm is guaranteed to minimize.

k-Means: The cost function ϕ is minimized by setting the centroid of every cluster to be the mean of all the points in that cluster. Thus, the algorithm operates as follows: Initially, k centroids are initialized. Thereafter, given the set of centroids, each point is assigned to the cluster associated with the nearest centroid; and the centroids are recomputed based on the assignments of points to clusters. The above process is executed for several iterations, as is detailed in Algorithm 1.

Manhattan distance Instead of minimizing the cost function ϕ, which is defined using the squared euclidean distance, we can minimize a cost function defined using the Manhattan distance: Given two random points A and B in d dimensional space such that $A = [\, a_1, a_2, ...a_d]$ and $B = [\, b_1, b_2, ...b_d]$, the Manhattan distance between A and B is defined as:

$$|a - b| = \sum_{i=1}^{d} |a_i - b_i| \tag{3}$$

The corresponding cost function ψ that is minimized when we assign points to clusters using the Manhattan distance metric is given by:

$$\psi = \sum_{x \in X} \min_{c \in C} |x - c| \tag{4}$$

k-medians: In contrast to ϕ, the cost function ψ is minimized setting the centroid of every cluster to be the *median* in every dimension of the points in that cluster. Thus, we obtain a variation on k-means, which is known as *k-medians*.

Algorithm 1 k-Means Algorithm

```
for iterations := 1 to MAX ITER do
    for  each point p in the dataset do
     Assign point p to the cluster with the closest centroid
    end for
     Calculate the cost for this iteration.
    for each cluster c do
     Recompute the centroid of c as the mean of all the data points assigned to c
    end for
end for
```

Please use the dataset from `q2/data` folder for this problem.

The folder has 3 files:

1. `data.txt` contains the dataset which has 4601 rows and 58 columns. Each row is a document represented as a 58 dimensional vector of features. Each component in the vector represents the importance of a word in the document.
2. `c1.txt` contains k initial cluster centroids. These centroids were chosen by selecting k = 10 random points from the input data.
3. `c2.txt` contains initial cluster centroids which are as far apart as possible, using Euclidean distance as the distance metric. (You can do this by choosing 1st centroid c1 randomly, and then finding the point c2 that is farthest from c1, then selecting c3 which is farthest from c1 and c2, and so on).

Note: Set number of iterations (`MAX_ITER`) to 20 and number of clusters `k` to 10 for all the experiments carried out in this question.

When assigning points to centroids, if there are multiple equidistant centroids, choose the one that comes first in lexicographic order.

(a) Exploring initialization strategies with Euclidean distance [3.5 pts]

1. **[2 pts]** Using the Euclidean distance (refer to Equation 1) as the distance measure, compute the cost function $\phi(i)$ (refer to Equation 2) for every iteration i. This means that, for your first iteration, you'll be computing the cost function using the initial centroids located in one of the two text files. Run the k-means on `data.txt` using `c1.txt` and `c2.txt`. Generate a graph where you plot the cost function $\phi(i)$ as a function of the number of iterations i=1..20 for `c1.txt` and also for `c2.txt`. You may use a single plot or two different plots, whichever you think best answers the theoretical questions we're asking you about.

   Hint 1: Note that you do not need to write a separate Spark job to compute $\phi(i)$. You should be able to calculate costs while partitioning points into clusters.

   Hint 2: The cost after 5 iterations starting from the centroids in c1.txt is approximately 460, 538, 000.

2. **[1.5 pts]** What is the percentage change in cost after 10 iterations of the K-Means algorithm when the cluster centroids are initialized using `c1.txt` vs. `c2.txt` and the distance metric being used is Euclidean distance? Is random initialization of k-means using `c1.txt` better than initialization using `c2.txt` in terms of cost $\phi(i)$? Explain your reasoning.

(b) Exploring initialization strategies with Manhattan distance [3.5 pts]

1. **[2 pts]** Using the Manhattan distance metric (refer to Equation 3) as the distance measure, compute the cost function $\psi(i)$ (refer to Equation 4) for every iteration i. This means that, for your first iteration, you'll be computing the cost function using the initial centroids located in one of the two text files. Run the k-medians on `data.txt` using `c1.txt` and `c2.txt`. Generate a graph where you plot the cost function $\psi(i)$ as a function of the number of iterations i=1..20 for `c1.txt` and also for `c2.txt`. You may use a single plot or two different plots, whichever you think best answers the theoretical questions we're asking you about.

   Hint 1: This problem can be solved in a similar manner to that of part a.

   Hint 2: The cost after 5 iterations starting from the centroids in c1.txt is approximately 412, 012.

2. **[1.5 pts]** What is the percentage change in cost after 10 iterations of the K-medians algorithm when the cluster centroids are initialized using `c1.txt` vs. `c2.txt` and the distance metric being used is Manhattan distance? Is random initialization of k-medians using `c1.txt` better than initialization using `c2.txt` in terms of cost $\psi(i)$? Explain why.

   Hint: to be clear, the percentage refers to (cost[0]-cost[10])/cost[0].

What to submit

You should upload your python file/jupyter notebooks, as well as a writeup that includes your answers and the plots. Make sure you add plenty of comments to your code to demonstrate your understanding. Submit the following:

1. Include the jupyter notebook(s)/python file(s) for 2(a) and 2(b).
2. A plot of cost vs. iteration for two initialization strategies [2(a)]
3. Percentage improvement values and your explanation [2(a)]
4. A plot of cost vs. iteration for two initialization strategies [2(b)]
5. Percentage improvement values and your explanation [2(b)]