Homework Assignment ##

Due: MMM DD, YYYY

**1. Evaluate the below algebraic expressions. Implement the solution in Python (using NumPy) and submit your scripts using the supplied Jupyter Notebook (homework1.ipynb).**

(a) I used + operator

A = np.array([[1, 2, 3], [4, 5, 6]])

B = np.array([[5, 1, 3], [3, 2, 2]])

sol = A + B

print(sol)

Sol)

[[6 3 6]

[7 7 8]]

(b) I used \*, - operator

A = np.array([[3, 1, 3], [1, 2, 2]])

B = np.array([[8, 2, 2], [4, 4, 6]])

""" TODO: Write your solution here """

sol = 2\*A - 0.5\*B

print(sol)

Sol)

[[2. 1. 5.]

[0. 2. 1.]]

(c) I used \*, - operator

A = np.array([[3, 1, 3], [1, 2, 3]])

B = np.array([[1, 5, 3], [5, 4, 1]])

""" TODO: Write your solution here """

sol = 0.5\*(A + B)

print(sol)

Sol)

[[2. 3. 3.]

[3. 3. 2.]]

(d) I used dot product function.

A = np.array([[3, 1, 2], [3, 2, 4]])

B = np.array([[1, 1, 0], [2, 0, 1]])

""" TODO: Write your solution here """

sol = np.dot(A, np.transpose(B))

print(sol)

Sol)

[[ 4 8]

[ 5 10]]

(e) I used trace function

sol = np.trace([[1,3,2],[6,4,5],[9,8,7]])

print(sol)

Sol)

12

(f) I used inv function

sol = np.linalg.inv([[3, 2],[-7, -5]])

print(sol)

Sol)

[[ 5. 2.]

[-7. -3.]]

(g) I used inv function

sol = np.linalg.inv([[1, -3, 5],[-1, 4, -5],[-1, 3, 6]])

print(sol)

Sol)

[[ 3.54545455 3. -0.45454545]

[ 1. 1. 0. ]

[ 0.09090909 0. 0.09090909]]

(h) I used det function

A= np.array([[1, 3],[4, -5]])

sol = np.linalg.det(A)

print(sol)

Sol)

-17.0

**2. Solve the following systems of linear equations. Implement the solution in Python (using NumPy) and submit your scripts using the supplied Jupyter Notebook (homework1.ipynb).**

(a) first, get inv matrix of left side matrix, and do matrix multiplication between right side matrix and inv matrix.

A= np.linalg.inv([[2,1],[-1,3]])

B= np.array([5,1])

sol = np.dot(A, np.transpose(B))

print(sol)

Sol)

[2. 1.]

(b) first, get inv matrix of left side matrix, and do matrix multiplication between right side matrix and inv matrix.

A= np.linalg.inv([[5,1, -2],[2, 1, 5],[1,-1,2]])

B= np.array([-4,1,1])

sol = np.dot(A, np.transpose(B))

print(sol)

Sol)

[-0.5 -0.5 0.5]

3.

(a) Euclidean distance is

(a, b is d dimension vector)

Input: X - list of input vectors

K - number of centroids

max\_iter - limit of iteration

tol - variable to check wheter some of squared error is smaller than some value

distance\_metric - the method of calculating distance (ex. euclidean distance)

Output: c - c[i] = centroid which is nearest to i-th vector

centroids - k centroid's location.

log\_centroids - history of centroid

log\_c - history of c

log\_sse - history of sum of suquared error

(b)

What do you need to provide for ‘X’ and ‘K’?

- X is array of input vectors.

- K is number of cluster

What do the values ‘max\_iter’ and ‘tol’ do?

-Max\_iter is limit of iteration.

-Tol is a variable to check whether some of squared error is smaller than some value.

What do the output ‘c’ and ‘centroids’ contain?

-C : Cluster to which input vector belongs(C[i] = centroid which is nearest to i-th vector).

-centroids : k centroid's location.

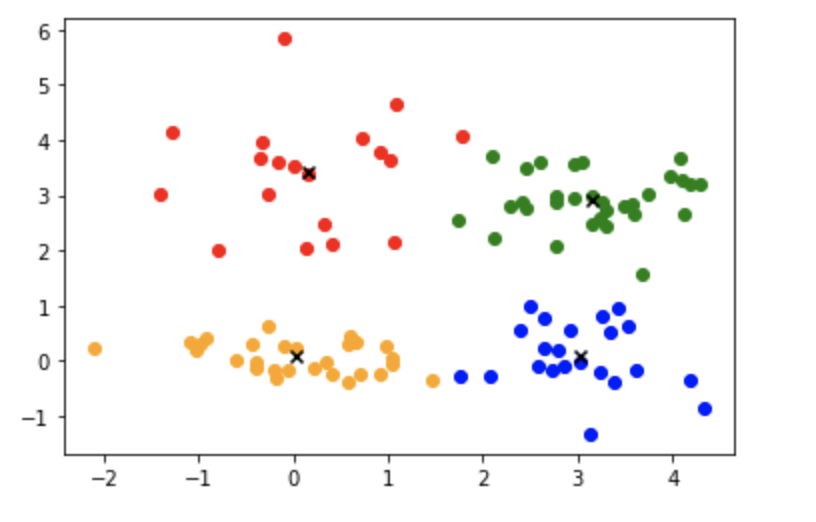
What do the output ‘log\_centroids’, ‘log\_c’, and ‘log\_sse’ contain?

-log\_centroids : history of centroid

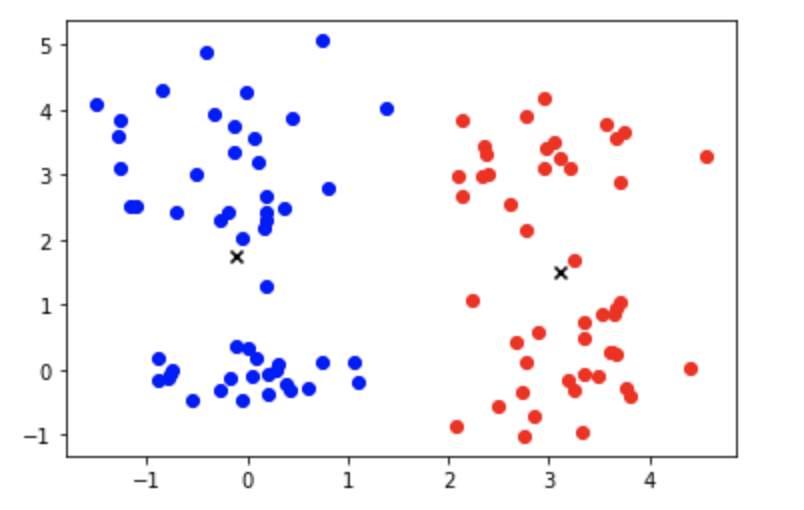
-log\_c : history of c

-log\_sse : history of sum of suquared error

(c) part1 - In your report, include the plot that you have created.



Part2 - In your report, include the plot that you have created.



- Based on the results, between *k* = 4 and *k* = 2, which parameter value do you think better and why? Submit your answer in the report.

* I think the case k=4 is more better, because points are gathered near the cluster in the case of k=4. However, if k=2, there is an empty space near the cluster.

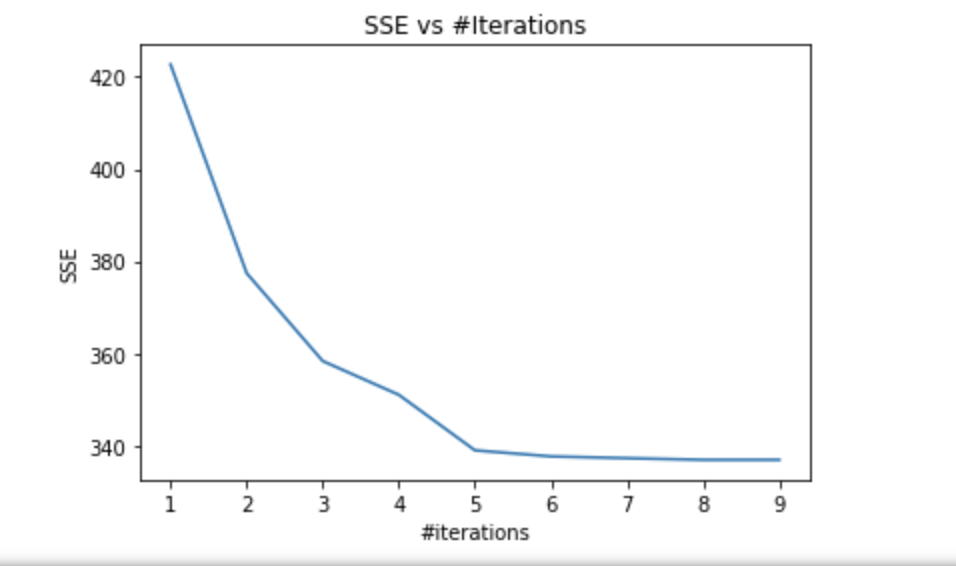
(d) - What does stats.describe(my\_data) do? What kind of information can you find out from the output of stats.describe(my\_data) mean?

- it analyzes and shows data’s characteristic. It informs nobs(num of vectors), minmax, mean, variance, skewness, kurtosis information.

in your report, include a formula that you can use to normalize (standardize) the data

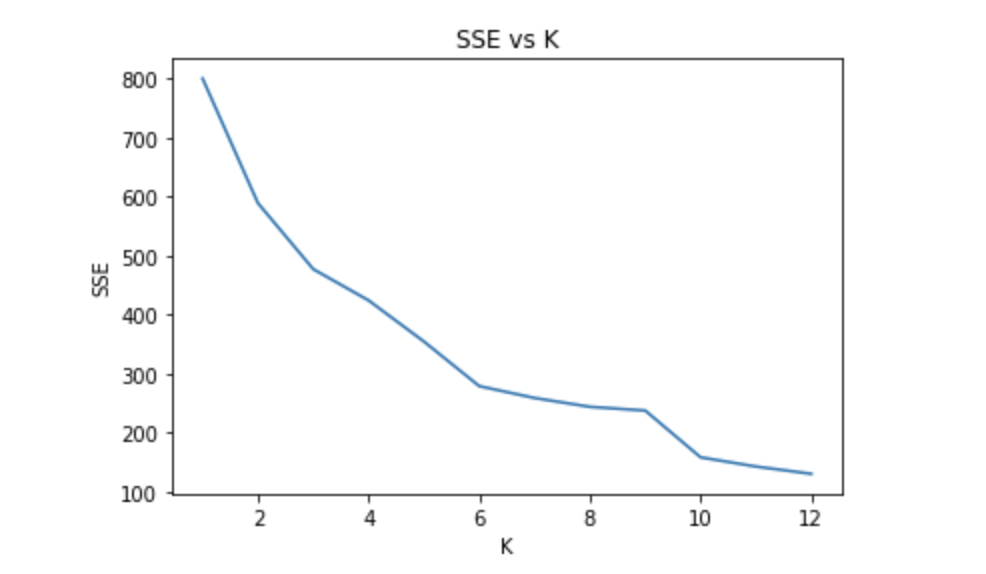
* The formula is Xi,j = X[i][j](data) – average of j-th column of X) / standard deviation of j-th column of X.

(e) In your report, include a plot that shows the trace of SSE (sum of squared errors) throughout the clustering (*i.e.*, a line plot of SSE over the iteration number). Explain how SSE changes over iterations



- The more iteration, the lower the SSE. When iterations go from 1 to 2, SSE change the most. Then, the amount of change of SSE decreases

(f) In your report, draw and submit an *SSE vs K* plot that looks similar to the one provided below.



Which *k* do you think the best *k* for this dataset? Write and justify your answer in your report.

I think best k is 12, because the SSE is the lowest.

Perform a formal analysis of the results. You may want to examine all individual clusters for your choice of *k*, by evaluating the mean, standard deviation, median, minimum, maximum, *etc.* to understand the properties of each cluster. With your best of knowledge, characterize and distinguish the clusters. (Write your answer in the report.)

텍스트이(가) 표시된 사진

자동 생성된 설명

I used 4 for K.

Each column means gender, age, annual income and spending score.

In case Cluster 1, all member’s gender is 1. The age group is relatively diverse.

In case Cluster 2, all member’s gender is 0. the group with the smallest average age.

In case Cluster 3, there are higher proportion of gender 0. the group with the highest average age. The age group is relatively diverse. Their income is high, but they don't spend much.

In case Cluster 4, there are similar proportion of gender 0 and gender1. Annual income and spending score are the biggest.

(g) In your report, explain why this is happening and how the outcome would be like (hint: the disclaimer given in Problem 3(b)).

Because of zero division problem.

In kmeans function,

for n in range(N):

if c[n] == k:

tmp\_sum += X[n]

tmp\_count += 1

centroids[k] = tmp\_sum / tmp\_count 🡨 it occurs the problem

if tmp\_count is zero, it occurs zero division problem.

If tmp\_count is zero, k-th centroid should become previous k-th centroid