Homework Assignment 4

Maximum earnable: 40 pt. + 6 extra pt.
Due: 11:59PM November 20, 2023

- Read the assignment carefully. *For some problems, you will need to* ***write and execute several Python scripts****; and* ***submit your code work together with their results***.
- For this assignment, you are required to **submit a report** and ***Python*** **code** *using provided templates*.
  - Download "Assignment template.docx" from *LMS* and write a **report** with your solutions. Ensure you **demonstrate the steps you took to arrive at your answers**.
  - For selected problems (indicated in the problems), you will need to complete code that implements your solutions in the report. Download and use the provided *Jupyter* notebook (an *.ipynb* file) as template.
  - Make sure your code produces the same results as the solutions included in your report; **otherwise, you will get penalties.**
- This assignment is meant to be an individual work; Please be clear with the HGU CSEE Standards:
  - Submitting assignments or program codes written by others or acquired from the internet without explicit approval of the professor is regarded as cheating.
  - Showing or lending one's own homework to other student is also considered cheating that disturbs fair evaluation and hinders the academic achievement of the other student.
  - It is regarded as cheating if two or more students conduct their homework together and submit it individually when the homework is not a group assignment.
- You are **permitted to re-use any code snippets from the lecture slides** while working on your solution code for the problems.
- **Use of ChatGPT or similar AI tools**: Students are prohibited from using ChatGPT or similar AI platforms to directly obtain solutions for this assignment. The intent of the assignment is to exercise your understanding and application of the course material. Leveraging AI tools to bypass this learning process is considered a breach of academic integrity. Any evidence of such behavior will result in penalties.
- Once completed, please submit your work via the *LMS.*

1. Clustering with the *k*-Means algorithm.

(a) (4 pt.) In your report, write down the definition of the Euclidean distance between two points *a* and *b*. Go to the *Jupyter* notebook and write function `euclidian_distance(a, b)` that takes two vectors and returns the Euclidean distance between them.

(b) (3 pt.) Find the code cell starting with **"`# Problem 1 (b)`"** in the provided *Jupyter* notebook. It is a completely functioning implementation of the *k*-means algorithm (you do not need to modify the code in the cell for addressing problems 1(b)-1(f)). Spend some time with the code and try to understand its logical flow. Recall our discussions in class.

```
# Problem 3 (b)
def kmeans(X, K, max_iter=100, tol=0.00005, distance_metric=euclidian_distance):
   ... (omitted: see the provided .ipynb file ) ...
```

* Disclaimer: The provided code is written for ITP40010 and should be used for educational purpose only. The user may experience the division by zero errors when the algorithm yields empty clusters (clusters with 0 instance) during its process.

```
/bin/miniconda3/envs/py37/lib/python3.7/site-packages/ipykernel_launcher.py:46:
RuntimeWarning: invalid value encountered in true_divide
```

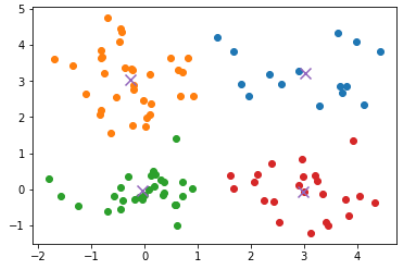
Regarding the implementation, answer the following questions (include your answer in the report):

- What do you need to provide for `'X'` and `'K'`?
- What do the values `'max_iter'` and `'tol'` do?
- What do the output `'c'` and `'centroids'` contain?
- What do the output `'log_centroids'`, `'log_c'`, and `'log_sse'` contain?

(c) (6 pt.) Find the code cell starting with **"`# Problem 1 (c)`"** in the provided *Jupyter* notebook. Function **"`generate_random_data(N)`"** is a utility that generates a 2-dimensional synthetic dataset (an artificial dataset with 2 features). The last two lines of the code cell invoke this function and visualize the generated dataset by drawing a scatter plot. Take a moment with the code to understand its logical flow. Try to execute and investigate the output of the function (you do not need to modify the code in the cell).

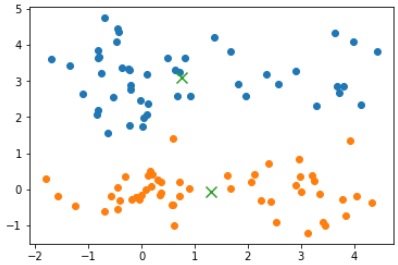
- Once you have reviewed the code, find the cell starting with **"`# Problem 1 (c) - part 1`"** and write a code snippet that executes the above *k*-means clustering implementation on the generated dataset, with $\boldsymbol{k = 4}$. Visualize the datapoints with colors, representing clusters. Also, indicate the position of the centroids that you obtained after running the implementation. In your report, include the plot that you have created. In the *Jupyter* notebook, submit your code snippet.

E.g., Clustering results when $k = 4$ (your answer should be different from the example)



- Now let us move onto the cell starting with **"`# Problem 1 (c) - part 2`"** and write a code snippet that clusters the data with $\boldsymbol{k = 2}$. Visualize the datapoints with colors, representing clusters. Also indicate the position of the centroids that you obtained after running the implementation. In your report, include the plot that you have created. In the *Jupyter* notebook, include your script.

E.g., Clustering results when $k = 2$ (your answer should be different from the example)



- 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. In the *Jupyter* notebook, include your script.

(d) (2 pt.) Turn to the code cell starting with **"`# Problem 1 (d)`"**. Execute the code cell. Make sure to place the data file (**"`Mall_Customers.csv`"**) in the right location (in the same directory as the *.ipynb* file is placed) so that the code cell loads the data without any problem.

You are given a new dataset containing customer information at a mall[1]. The dataset consists of four columns; gender (male/female), age (18-70), annual income (unit: 1000 USD), and spending score (1-100).

In your report, write down your answers to the following questions.

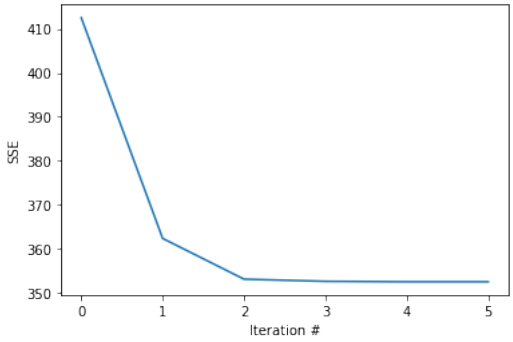
- What does `stats.describe(my_data)` do? What kind of information can you find out from the output of `stats.describe(my_data)` mean?

As you may notice from the output of `stats.describe(my_data)`, the dataset consists of the features that have varying ranges from each other. In your report, include a formula that you can use to normalize (standardize) the data. Then, apply the provided function **"`normalize(X)`"** to `my_data` and examine the resulting dataset with `stats.describe(normalize(my_data))`. How does the dataset change after applying `normalize()`? Describe the differences in your report.

(e) (4 pt.) Write code that normalizes `my_data` and then conduct clustering on it. Use $k = 5$. In your *Jupyter* notebook, include a code snippet that performs the task and prints out the clustering result (the cluster membership of all 200 data instances).

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.

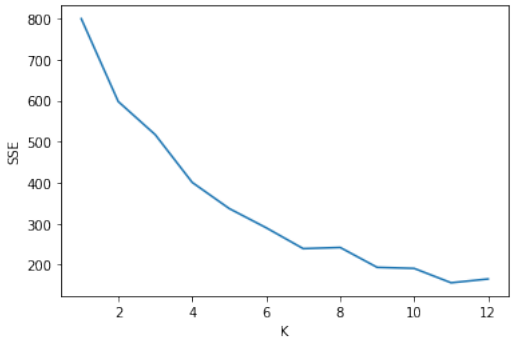
E.g.,



(f) (Max 10 pt.) Write code that tries out clustering with multiple values of k, ranging between 1 and 12. You may see a runtime warning message **"`RuntimeWarning: invalid value encountered in true_divide`"** while executing. Repeat the code run, until you do not see any warning message.

[1] Dataset source: https://www.kaggle.com/akram24/mall-customers

- (3 out of 10 pt.) In your report, draw and submit an *SSE vs K* plot that looks similar to the one provided below. In your *Jupyter* notebook, include your code snippet together with the execution result.



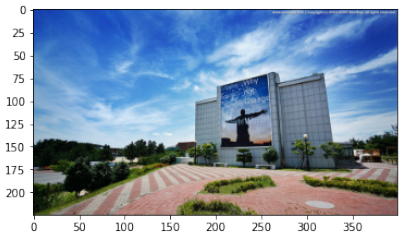
- (Max 2 out of 10 pt.) Which *k* do you think the best *k* for this dataset? Write and justify your answer in your report.

- (Max 5 out of 10 pt.) 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.)

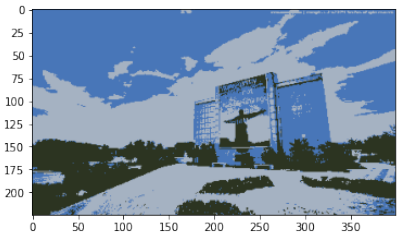
(g) (Max 6 extra pt.) While you are tackling the above problems, you may have seen the runtime warning: `"invalid value encountered in true_divide"` multiple times. This has occurred due to a particular reason. In your report, explain why this is happening and how the outcome would be like (hint: the disclaimer given in Problem 1(b)). In the *Jupyter* notebook, modify function `kmeans()` such that you do not see the warning anymore and obtain correct results.

2. Image segmentation using k-means.

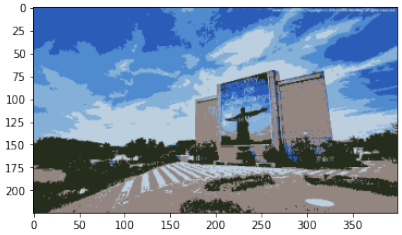
(11 pt.) Using the k-means implementation from Problem 1, write a code that segments provided image file `hyoam.jpg` with k=3, 6, and 9, respectively. Explain the differences among the results with different *k* in your report. Submit your complete code in the *Jupyter* notebook. The resulting image would look like:



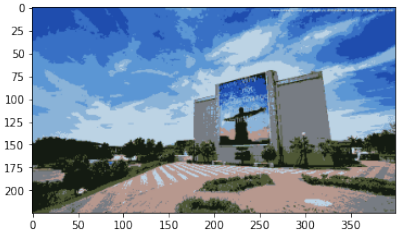
Original



$k = 3$



$k = 6$



$k = 9$

In the notebook, you are provided with a code snippet that reads in `hyoam.jpg`, coverts the pixel representation to the RGB color, and reshapes the pixels into a matrix form (`X_rgb`; where each row represents a pixel, and each column represents R, G, and B, respectively).