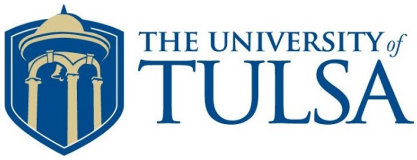



Machine Learning
CS-7333

# Unsupervised Learning Project

*Instructor:*
Prof. Sandip Sen

*Author:*
Selim Karaoglu

April 13, 2022

In this assignment we implement k-Means algorithm and Expectation maximization methods. We conduct the experiment on three part; first we evaluate the clustering results for Gaussian distribution datasets, following, we utilize k-Means algorithm for image compression, further we employ the algorithms on Mall Customers dataset.

# 1 Introduction

In this project, we focus on implementing the k-Means clustering algorithm and Expectation Maximization (EM). First we implement the algorithms from scratch with providing information about their mathematical background. Then we proceed to experiment with the algorithms on different datasets. In this assignment, we employ 3 different dataset; first dataset is the Gaussian distribution data provided with the project assignment, second dataset is 4 different images to apply compression by using k-Means clustering and finally we conduct the experiment with the Mall Customer dataset provided by UCI Machine Learning Repository and compare the scores calculated with 4 different metrics.

# 2 Methods

This section provides in-depth information about the algorithm and data adopted in this assignment. First we present information about the k-Means clustering algorithm and EM. Following the algorithms, we present the datasets we utilized in this project. We explain datasets in depth to uncover possible outcomes of the experiment and to prevent any anomaly in datasets.

## 2.1 k-Means Clustering and Expectation Maximization

k-means Clustering algorithm is an unsupervised machine learning technique that aims to divide $n$ data points into $k$ clusters. k-Means algorithm iteratively partitions the dataset into $k$ distinct and non-overlapping groups. With this algorithm, a single datapoint can only be a member of a single cluster. The main objective of the k-Means algorithm is to partition the data into $k$ clusters where the value of $k$ is unknown.

This algorithm is built with the following procedure;

- Define $k$.
- Randomly select $k$ centroids.
- Iterate until there is no change or stop iterating when the maximum number of iterations is reached.
- For each point, calculate euclidian distance from each centroid and assign points to the closest centroid. This step is called E-step in EM and k-Means is a specific case of an Expectation algoritm.
- Compute the mean of each cluster and reassign the centroids. This step is called M-step.
- Repeat until system reached to maximum number of iterations or there is no change on centroids.

k-Means clustering algorithm uses distance metrics to calculate distances of the points from centroids. To achieve this task, k-Means algorithm adopts the loss function. The loss function in the k-Means algorithm can be defined as;

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^j - c_j \right\|^2$$

where $J$ is the objective function, $k$ is the number of clusters, $n$ is the number of cases, $x$ is the respective case and $c$ is the centroid of a cluster. With the introduction of distances to the loss formula, we get;

$$J = \sum_{i=1}^{m} \sum_{k=1}^{K} w_{ik} \left\| x^i - \mu_k \right\|^2$$

At this point, we need to minimize the objective function with minimizing one of the variables while keeping the other variable fixed. This represents the E and the M steps of the algorithm. First we differentiate the $J$ w.r.t. $w_{ik}$ while keeping $\mu$ fixed and update the cluster assignments, this is interpreted as E step. Following the E step, we differentiate the $J$ again, but this time w.r.t. $\mu$ while the $w_{ik}$ is treated as fixed. We update the centroids after this calculation and this is called the M step.

We can present the formula for E step as:

$$\frac{\partial J}{\partial w_{ik}} = \sum_{j=1}^{k}\sum_{i=1}^{n} \left\| x_i^j - c_j \right\|^2 = 0$$

$$w_{ik} = \begin{cases} 1 \; if \; k = \arg\min_j \left\| x_i^j - c_j \right\|^2 \\ 0 \; otherwise \end{cases}$$

The M step can be formulated as:

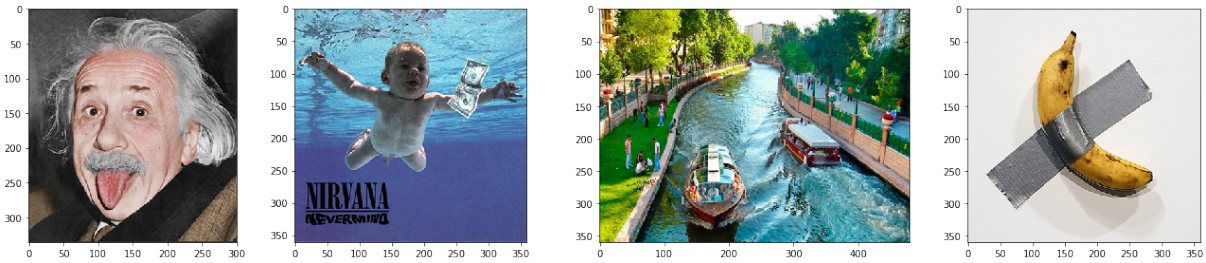
$$\frac{\partial J}{\partial \mu_k} = 2\sum_{i=1}^{m} w_{ik}(x^i - \mu_k)$$

$$\mu_k = \frac{\sum_{i=1}^{m} w_{ik} x^i}{\sum_{i=1}^{m} w_{ik}}$$

## 2.2 Datasets

This project is based on three different tasks, and for each task of this assignment, we present the datasets employed in this project. For the first part of this assignment, we study the Gaussian Distributed datasets created with different parameters. We train and test the k-Means clustering algorithm on the datasets and present the resulting accuracy. These datasets are provided with a code file included in the assignment.

We shift our focus to image compression in the second part of the experiment. Figure 1 shows the original images we selected for the experiment. We picked the images with considerations of color balance, contrast, complication etc. factors that affect the compression process.




(a) 1st Image (b) 2nd Image (c) 3rd Image (d) 4th Image

Figure 1: Original Images

For the last part of our experiment, we examine a real life dataset provided in Kaggle; Mall Customer Data. This dataset is built with collecting basic data about the customers with membership cards. This basic data contains information about the gender, age and the annual income of a customer, in addition to that the spending score feature is a score assigned by the mall authorities to represent the customers spending behavior. Figure 2 provides information about the distribution of the attributes. Age and spending score distributions shows similarity to summed distribution of 3 Gaussians, annual income distribution shows similarity to summed distribution of 2 Gaussians.

The violin plots presented in Figure 3 shows the gender distributions for age, annual income and spending score features. These distribution plots provide some insight about the dataset, helps to detect any incompatible entries in the dataset and see the conditional probabilistic relations between features. Figure 3 shows that the gender distributions against other features, as the plots suggest the data distributions are similar for gender. There majority of the attendants are middle aged, annual incomes average around 60k for both genders with males having slightly higher income and spending score averages around 50 for both genders.

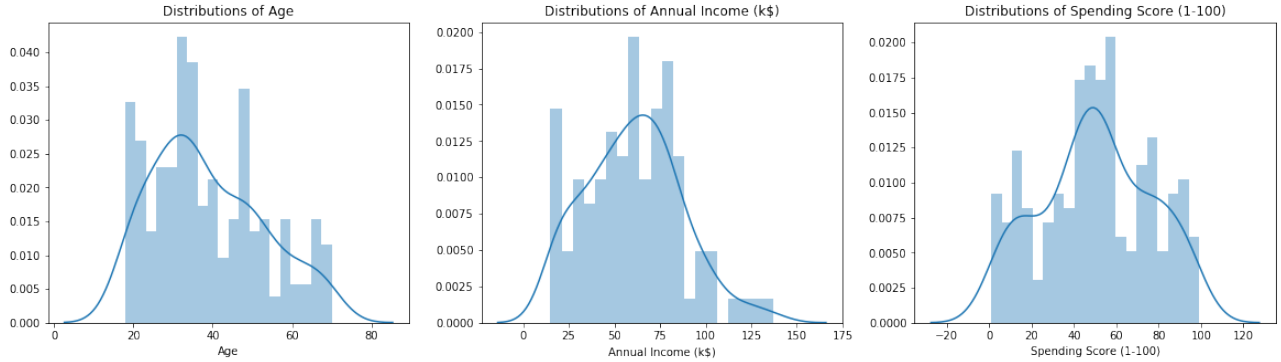



Figure 2: Distribution plots of age, annual income and spending score features

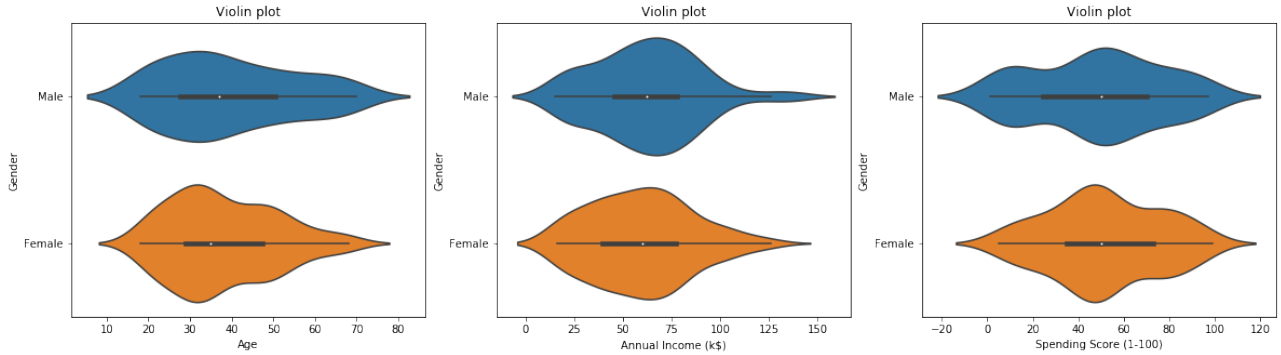
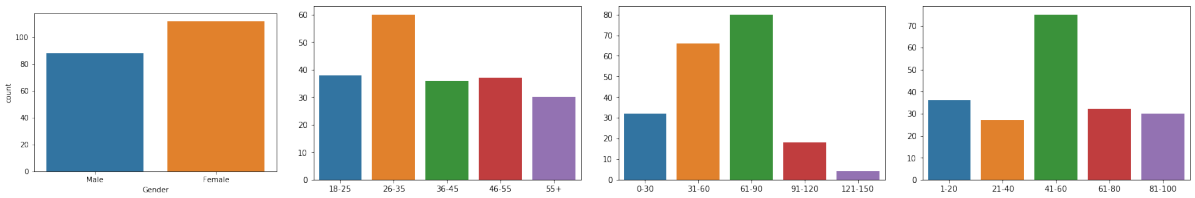



Figure 3: Violin plot of gender distributions over age, annual income and spending score

The distribution and violin plots provide information about each entry in the dataset, however in this project we focus on clustering and therefore categorizing the features might provide us some insight about how the clustering algorithms might recover the underlying data distribution. Figure 4.a shows that the gender distribution is biased in favor of females, although the difference does not appear to be significant enough to cause issues on the training. As Figure 4.b suggests, the age group of 26-35 dominates the dataset. Other ages seem to share similar percentages in the dataset. This imbalance in the age distribution might cause the clustering algorithms to group similar aged people in the same cluster, even though they belong to different clusters. Annual income distribution does not seem as an equal distribution among different income classes. Nevertheless, the distribution of annual income does seem like a Gaussian distribution. As Figure 4.d suggests, spending score distribution is stacked around the mean but overall distribution excluding the middle group seems similar.




(a) Gender Distribution (Pre-Categorized) (b) Age Distribution (Categorized) (c) Annual Income Distribution (Categorized) (d) Spending Score Distribution (Categorized)

Figure 4: Feature Distributions of Mall Customer dataset

The pre-processing applied for this project showed us that this dataset doesn't have any missing values or duplicates. Dataset is built with 200 entries with features; gender, age, annual income and spending score. The mean age in the dataset is 38,85, mean annual income is 60,56k and the mean spending score is 50.2

# 3 Experiment

In this section, we utilize the k-Means and EM to perform on different datasets we presented previously. We begin the experiment with the gaussian datasets. Following, we examine the compression results for each image separately. Further, we implement the unsupervised learning we developed over real life data and present our findings for each dataset. To present the dataset creation process, we provide the variables for each experiment. In our variable set, N is the number of samples, k is the number of clusters, S is the number of gaussian distributions, st_dev is the standard deviation and Spacing is the distance between the means of each gaussian.

## 3.1 Gaussian Distribution Datasets

This experiment is based on the gaussian dataset creation process provided with this assignment. We created the dataset and named the files with the order they utilized in this project. The detailed output for each clustering process is provided with the outputs of the code file, here we present a summary of experiments and some interesting findings about the experiment results.

### 3.1.1 Experiment 1

- First part of this experiment is built with gaussian data where; number of samples (N) is 1000, Spacing is varied as 0.5, 1, 1.5 and 2 also k is varied as 2, 3, 6 and 8.

  The first experiment resulted with k-Means achieving slightly higher likelihood values comparing to EM. On the first experiment where variables defined as S = 3, N = 1000, Space = 0.5 and k = 2; the likelihood performances are the same for both models. On the run where the $k$ is 3 (equal to spacing), the centroid values for EM are more accurate than k-Means. On the run with S = 3, N = 1000, Space = 0.5 and k = 3; EM algorithm predicted the centroids on [0.35, 0.95, 1.55] and k-Means predicted the centroids on [-0.35, 0.94, 2.26] where the nominal means are [0.5, 1, 1.5]; as can be seen from the outputs, the centroid predictions of EM is more accurate than k-Means w.r.t. spacing values. Even on this specific run, k-Means performed slightly better than EM with likelihoods being 0.39 and 0.35 respectively. Other runs yielded similar results to the run we examined.

- Second part of this experiment is built with gaussian data where; N = 1000, k = S = (5, 10), standard deviation is 1 and spacing varied as 0.5, 1, 1.5 and 2.

  This experiment's results showed that considering the likelihoods, EM outperformed k-Means on every run of this experiment. On the run where spacing is equal to 1, the nominal means are [1, 2, 3, 4, 5]; where EM predicted centroids are [1.32, 2.33, 3.18, 3.85, 4.43] and k-Means predicted centroids are [0.31, 1.77, 3.14, 4.47, 6.01]. Similar to the previous part of the experiment, the centroid predictions are closer to nominal means with the EM method. For this specific run, EM achieved 0.65 likelihood while k-Means achieved 0.44. Comparing to the previous part, this part adopts a more complicated distribution. Remaining runs support the findings we presented for the run we examined, while EM outperformed k-Means on likelihood, EM centroid predictions are also closer to the nominal means of the dataset when $k$ is equal to $S$.

### 3.1.2 Experiment 2

- On this experiment, datasets are created with variables N = 1000, k = S = 5, spacing is 3 and standard deviation is varied as 1, 2 and 3.

  Experiment results showed that EM has higher likelihood values than k-Means when the st_dev is larger than 1. With the st_dev being 1, k-Means achieved higher likelihood than EM. When st_dev is 1; EM and k-Means achieved likelihoods 0,65 and 0,79 respectively. As the value of the st_dev increases, the likelihood of the k-Means decreased. These results suggest that k-Means performed better when the st_dev has a low value considering our variables.

### 3.1.3 Experiment 3

- This experiment is conducted on the gaussian data created with variables; N = 1000, k = S = 5, spacing is 1.25 and standard deviation is sampled from uniform distribution over [0.75, 2]

In this run, both algorithms achieved low likelihood values comparing to the previous runs. k-Means beat the EM on likelihood with 0.05 difference where they achieved 0,37 and 0,32 respectively. The centroid predictions are similar to our previous experiment results. EM has more accurate centroids, in addition to that, on this run EM perfectly defined the last cluster of the dataset.

#### 3.1.4 Experiment 4

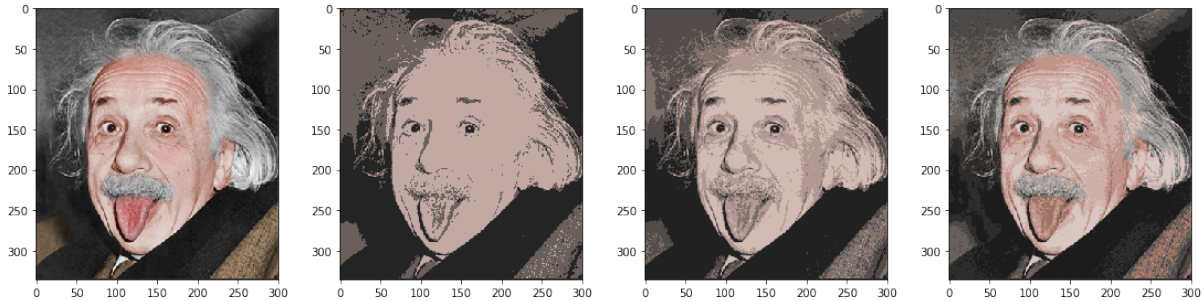
- Experiment 4 is designed where N is varied as 100, 1000 and 5000, k = S = 5, spacing is 1.25 and standard deviation is 0.75.

On the last part of our experiment, we observed that k-Means achieved higher likelihoods on each of the 3 gaussian data comparing to the EM, similarly to first part of the experiment. In contrast to the first run, k-Means centroid predictions are closer to the nominal means on this run. On the second run (where S = 5, N = 5000, Space = 1.25, k = 5) we observed that EM centroid predictions are [1.25, 2.72, 3.51, 4.14, 5.89] and k-Means centroid predictions are [0.71, 2.15, 3.54, 4.93, 6.33] where the nominal means are [1. 2.25 3.5 4.75 6.]. On this run k-Means achieved better centroid predictions than EM.

### 3.2 Image Compression with k-Means

In this section, we utilize the k-Means algorithm to compress images. k-Means algorithm clusters the pixels that has similar value, this allows the image compression. We applied k-Means compression on four different images.

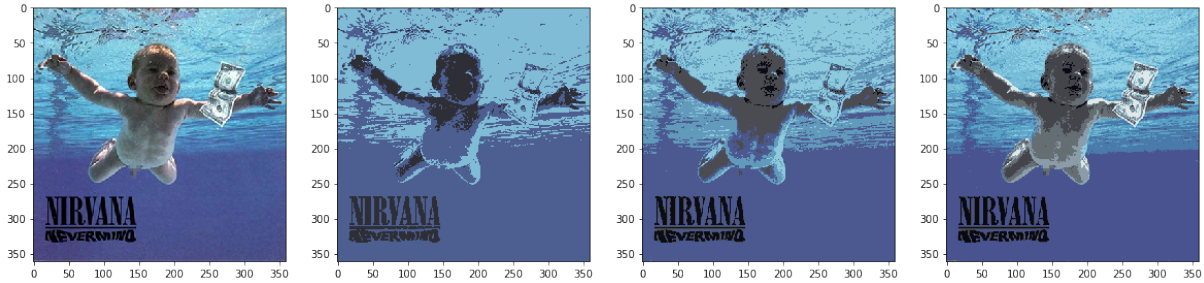
Figure 5 shows the original image in Figure 5.a and k-Means compressed images with number of clusters 3, 5 and 10 represented in Figure 5.b, Figure 5.c and Figure 5.d respectively. This first image is the colored version of the iconic picture of Albert Einstein. As can be seen from Figure 5.a, the original image has a high contrast between foreground and background. Figure 5.b shows that the background - foreground distinction is sharply emphasized in the resulting image, it is clear that it is a portrait and even the identity of the person can be recognized. This result is hard to achieve with only 3 clusters, however the contrast of the image helps in this situation. In Figure 5.b the resulting image can recover the important shadow structures in the image, therefore could be a good choice for posterization of images. In Figure 5.d, where the number of clusters are 10, the compressed image is hardly seperable from the original with only some color loss visible.



(a) Original Image (b) k=3 (c) k=5 (d) k=10

Figure 5: k-Means image compression 1st image

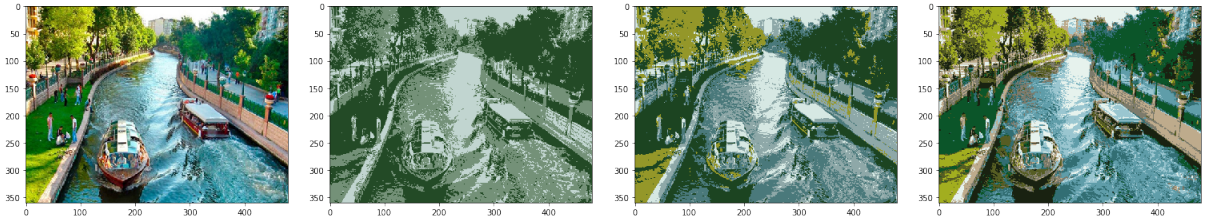
In our second image experiment, we worked on the Nirvana's famous album cover Nevermind as seen in Figure 6.a. This image has very low contrast, and it poses a challenge against small number of clusters. Figure 6.b shows that our first experiment with 3 clusters proved this claim, the features of the image is almost completely lost and the major figures from the original image are hardly distinguishable. With the shadows of the face, we can hardly see that there is a baby in the picture. Figure 6.c shows that even with 5 clusters, the image does not appear very clear, although in this result the baby and the dollar bill are slightly more apparent. As shown in Figure 6.d, image only has clarity when the number of clusters increased to 10. The main features of the image can be observed directly in this result.




(a) Original Image (b) k=3 (c) k=5 (d) k=10

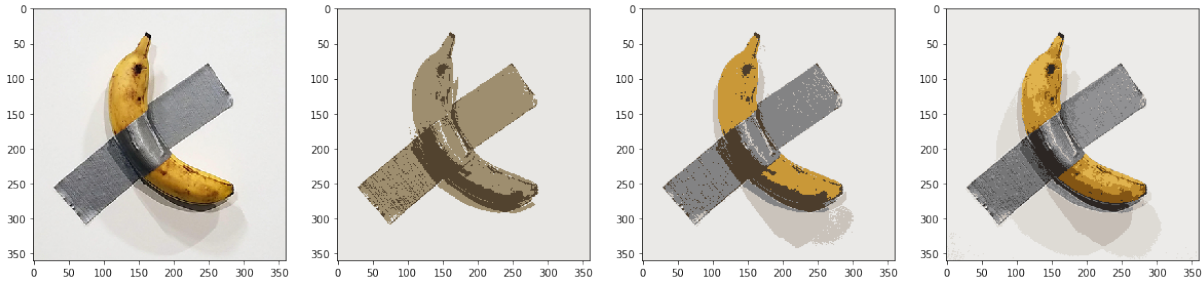
Figure 6: k-Means image compression 2nd image

This part of the experiment focuses on the picture of a beautiful city; Eskişehir. This picture has very distinctive contrast and shadows and enlightened areas are entwined. However the color distribution is biased on the green and blue colors as Figure 7.a suggests. We would not expect an image this complicated can achieve good results with low numbers of clusters. Figure 7.b shows the compressed image where the number of clusters are 3. The image is completely covered in 3 different tones of green. Combined with the previous image, they suggest that if there is a dominant color in the picture, even other cluster centers would be affected by it. When the number of clusters are assigned 5, the resulting image presented in Figure 7.c shows the introduces the second most dominant color of the original image back to the compressed image. It is still not detailed enough for us to observe the people standing on the left side of the image. In contrast, Figure 7.d suggests that when the number clusters increased to 10, most of the important features in the image becomes clearer.



(a) Original Image (b) k=3 (c) k=5 (d) k=10

Figure 7: k-Means image compression 3rd image



(a) Original Image (b) k=3 (c) k=5 (d) k=10

Figure 8: k-Means image compression 4th image

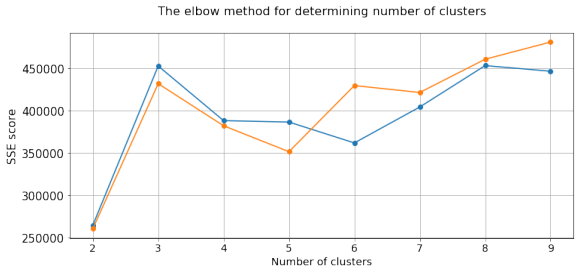
In our last experiment of image compression, we examine the picture of the record breaking art installment. This image has a good contrast and distinguishable color distribution provided by the simple plain white background. There are not too many different colors, therefore the compression on this image should be clear even with the smaller numbers of clusters. Original image is shown in Figure 8.a. Figure 8.b shows the compressed image with 3 clusters. There are only three colors and both of them are resulted from the color of the banana as expected. With the help of the contrast based on shadows, the image is understandable even with 3 cluster compression. Figure 8.c shows the compression with 5 clusters and resulting image is very bright,

vivid and distinctive in colors; this can be a good poster. The 10 cluster compression result shown in Figure 8.d is very similar to the original image, since this image is a simpler image (comparing to other 3 images presented previously) resulting compressed images can represent the original image better.

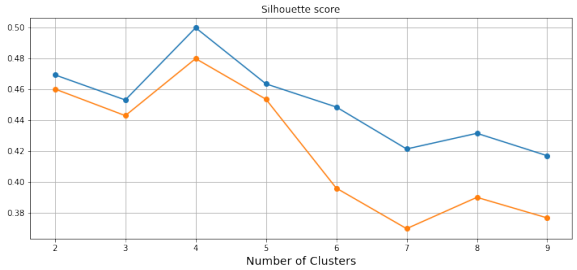
k-Means clustering algorithm can be employed on image compression tasks. To achieve good results with the image compression, the number of clusters for the k-Means algorithm should be picked carefully. As our experiment suggests, different aspects in the images might require modifications on the algorithm. The algorithms success depends on the structure of the image, simpler images can be compressed even further without leveraging clarity.

### 3.3 Mall Customer Dataset

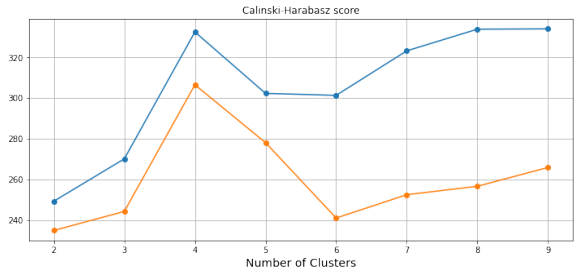
This part of the experiment focuses on the application of k-Means clustering and EM on real life data examples. To define the success of the clustering we employed 4 different metrics. Sum of Squared Error (SSE), Silhouette Score, Calinski-Harabasz score and Davies-Bouldin score. We modified the dataset and conducted two seperate experiments on the same dataset. First we examined the dataset with only age and spending score features. After clustering the dataset with partial features, we evaluate the clustering models on the dataset with all features inserted. For each trial, we clustered the dataset with the number of clusters $k$ altered between 2 and 9. For each $k$ value, we evaluated the EM and k-Means with 4 different metrics.



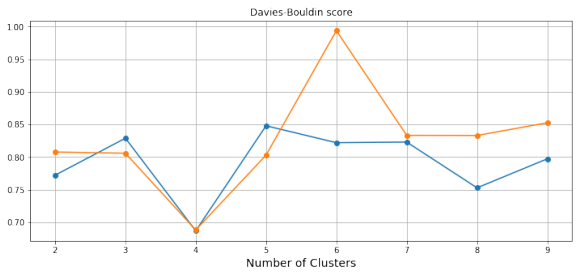

(a) SSE score




(b) Silhouette score




(c) Calinski-Harabasz score




(d) Davies-Bouldin score

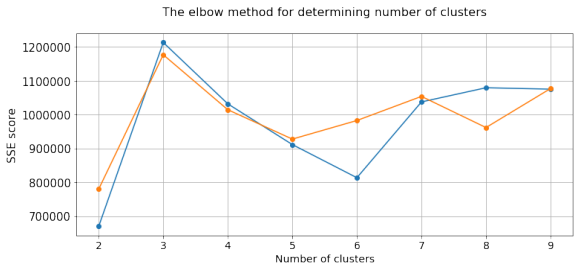

(e) EM clustering results


(f) k-Means clustering results

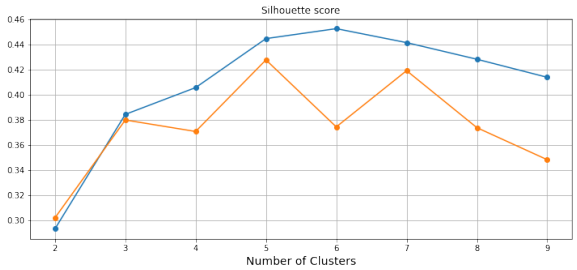
Figure 9: Mall Customer data with age and spending score

First experiment we conducted on real data focused on two features with representation concerns. By training the models on 2 features, we were able to plot the clustering achieved by the models in euclidian space. For each $k$, k-Means achieved higher SSE score, higher Silhouette score and higher Davies-Bouldin score than EM, however; EM performed better than k-Means according to Calinski-Harabasz score. Figure 9 shows the resulting scores for both clustering methods over different numbers of $k$. Figure 9.e shows the predicted clusters by EM and Figure 9.f shows the predicted clusters by k-Means.

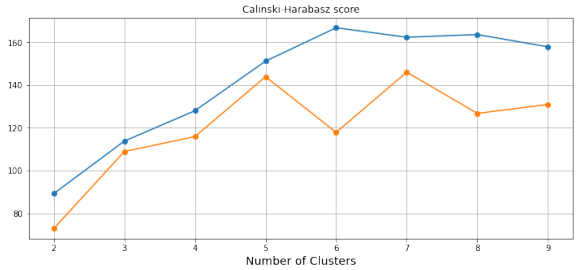
Second part of our experiment conducted on the complete dataset. We apply the same steps with the only difference being we conduct the experiment by utilizing every feature in the dataset. With the introduction of the 3rd feature to the dataset, the scores changed and EM achieved better scores than k-Means on especially SSE and Davies-Bouldin scores. Figure 10 shows the scores for both models with different numbers of $k$. Figure 10.a shows that SSE scores were close for the models and EM performed slightly better overall. As Figure 10.b and Figure 10.c suggests Calinski-Harabasz and Silhouette scores show similar structures with k-Means having the higher scores. Figure 10.d shows Davies-Bouldin metric resulted with higher EM scores, although the scores are close to each other for both models. Figure 10.e represents the EM clusters on 3 dimensional plot, similarly Figure 10.f represents the clusters predicted by the k-Means algorithm.



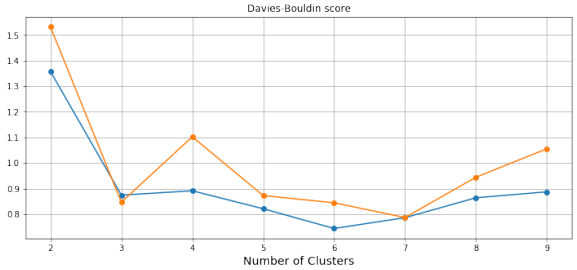

(a) SSE score



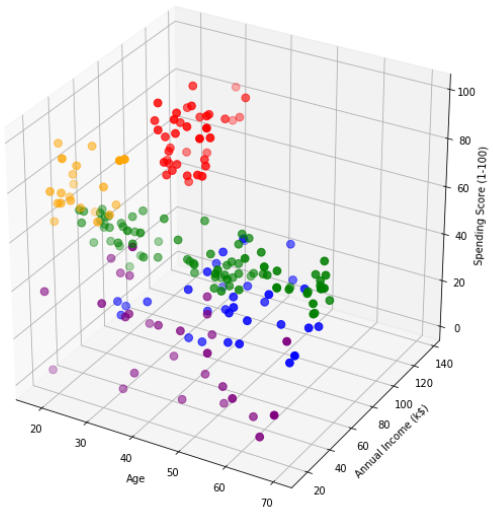

(b) Silhouette score



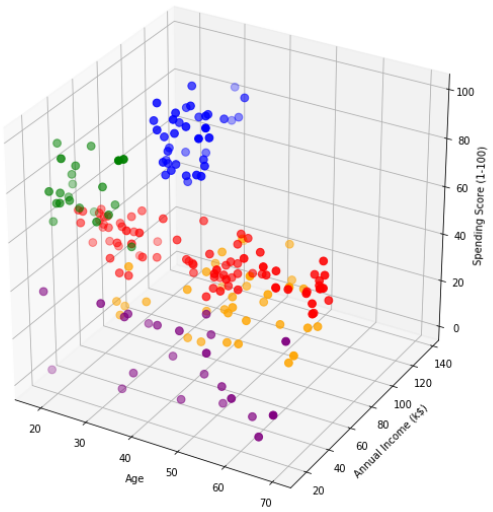

(c) Calinski-Harabasz score




(d) Davies-Bouldin score




(e) EM clustering results




(f) k-Means clustering results

Figure 10: Mall Customer data

# 4 Conclusion

In this assignment, we experimented with unsupervised learning methods k-Means clustering algorithm and Expectation Maximization method. We implemented both models from scratch and compared the results of both models on different datasets. First we focus on gaussian distribution data. Further, we experiment on image compression with k-Means. Following that, we conduct experiments with k-Means and EM on Mall Customer dataset. Different experiments showed that, k-Means and EM have their own advantages and disadvantages; therefore model picking should be done with considerations on different aspects of data.