Modern Machine Learning
Computer Assignment #6

1. *K-means algorithm:* Implementing the K-means algorithm for clustering

   Background: The K-means algorithm can be summarized in two steps that are performed repeatedly until convergence.

   - **Cluster assigment:** Every example $\mathbf{x}_i$, for $i = 1, \ldots, m$, is assigned a clusted centroid $\boldsymbol{\mu}_{c_i}$, where $c_i \in \{1, \ldots, K\}$ is the cluster label assigned to $\mathbf{x}_i$.

     $$c_i = \arg\min_k \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2, \text{for } k = 1, \ldots, K$$

   - **Cluster position update:** Update the cluster position as

     $$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{j:c_j=k}^{K} \mathbf{x}_i,$$

   where N_k is the number of elements assigned to cluster k. Therefore we update the cluster positions taking the mean of the elements in the cluster.
   The above two steps are repeated until convergence is achieved or a certain number of iterations is reached.

   Python files: The script Kmeans_example.py uses the pre-defined function $Kmeans.get\_centroids(data, K, centroids)$ to perform K-means clustering.
   Submission guidelines:

   - Your submission should be a unique **zip folder**, which is a modified version of Kmeans_example.py and includes your own function implementing the K-means clustering. Your own function should realize the same functionality as $Kmeans.get\_centroids$.

   - $Kmeans.get\_centroids(data, K, centroids)$ function has three inputs: (i) *data* is a matrix with $(m \times 2)$ dimension containing the dataset; (ii) K is the number of clusters; (iii) *centroids* is a matrix with $(K \times 2)$ dimension containing the centroid of each cluster.

- $Kmeans.get\_centroids(data, K, centroids)$ function has two outpus: (i) $new\_centroids$ is a matrix with $(K \times 2)$ dimension containing the centroid of each cluster derived from the above two steps; and (ii) $classes$ is a vector with $m \times 1$ dimension to indicate each data point belongs to which cluster.
- Please rename the modified file Kmeans_example.py replacing the word 'example' in the provided script with your last name, e.g., Kmeans_smith.py. **This should be the main function**.
- A pdf file with a figure showing the dataset and the trajectories of the centroids until convergence.

MATLAB files: The script Kmeans_example.m uses the matlab function $kmeans(\cdot)$ to perform K-means clustering.
Submission guidelines: Your submission should include:

- A unique **zip folder**, which should include a modified version of Kmeans_example.m and the three following functions that implement your own version of K-means: the first function is $\mathbf{c} =$ cluster_assignment$(\mathbf{X}, \boldsymbol{\mu})$, where $\mathbf{X}$ is a matrix containing the m data points (examples), $\mathbf{c} = [c_1, \ldots, c_m]^T$ are the cluster labels for the examples in $\mathbf{X}$, and $\boldsymbol{\mu}$ are the positions of the centroids, the second function is $\boldsymbol{\mu}_{new} =$ cluster_update$(\mathbf{X}, \mathbf{c}, K)$, where $\mathbf{X}$, and $\mathbf{c}$ are as defined above, K is the number of centroids, and $\boldsymbol{\mu}_{new}$ are the updated positions of the centroids. The third function is $[\mathbf{c}_{opt}, \boldsymbol{\mu}_{opt}] =$ myKmeans$(\mathbf{X}, \boldsymbol{\mu}_0, K)$, where $\mathbf{X}$, and K are as defined above, and $\boldsymbol{\mu}_0$ are the initial positions of the centroids. The function function myKmeans$(\cdot)$ should utilize the functions cluster_assigment$(\cdot)$ and cluster_update$(\cdot)$ to implement K-means. You should insert this function where indicated in the script. Make sure the results you obtain are very close to the ones returned by the MATLAB function kmeans$(\cdot)$.
  Please rename the modified file Kmeans_example.m replacing the word 'example' in the provided script with your last name. For example Kmeans_smith.m. **This should be the main function**.
- A pdf file with a figure showing the training data and the trajectories of the centroids until convergence.