Miner2 Username: Sravv
Mason UserID: ssangara
Part 1 – Iris clustering: Best public score: 0.72, Rank: 191
Part 2 – Image clustering: Best public score: 0.53, Rank: 314

HW 3: K-MEANS CLUSTERING (Part 1 and 2)

Problem:
To implement K-Means clustering algorithm which is divided into two parts:
Part 1: *Iris Clustering*
To assign 150 instances of the iris dataset to 3 cluster ids based on 4 features – sepal length, sepal width, petal length and petal width.
Part 2: *Image Clustering*
To assign 10,00 instances of handwritten digits images to 10 clusters, where each row consists of 784 comma-delimited integers.

Data-preprocessing:

Iris Dataset
First, I have imported all the required libraries such as Numpy, Pandas, Seaborn, Matplotlib and Sklearn. Then for loading the dataset, I used read_csv() from Pandas where I selected 'space' as separator in order to get the sepal length, sepal width, petal length and petal width in different columns of the feature matrix.

For checking if there are any null values, I used isnull().sum() and found out that there are no such values present in the given dataset. Next, I have performed feature scaling on the dataset using normalization so that the data is in the range of [0,1].

Image Dataset
In a similar fashion, for the image dataset after importing the required libraries, I have loaded the given dataset using Pandas' read.csv() function. As the data was already flattened out into array consisting of 784 columns per row for each image instance, the next step was to directly check for null values using isna() function. Then I replaced the NaN values with 0 using the replace() function to be able to perform further actions on the data. For feature scaling on this data, I have divided the data points with 255 instead of normalization.

PCA (Principal Component Analysis):

In the next step, for dimensionality reduction I have used Principal Component Analysis. It is a non-dependent procedure where the attribute space is reduced to a smaller number of factors from large number of variables. It searches for linear combination of variables in order to extract maximum variance from the variables. PCA helps speed up the algorithm with high number of

features. Data visualization is another important application of PCA which I have implemented in this assignment. To reduce the size of the dataset to 2 dimensions, Decomposition module from sklearn library is used, from which PCA class is imported. Then I have created an instance of this class which takes the parameter n_component=2 which reduces the dimensionality of the data to 2.

K-Means Implementation:

K-Means clustering algorithm is an unsupervised learning algorithm which groups the data points into 'k' different clusters based on distance between points as a measure of similarity. This clustering algorithm can be described as mentioned below:

1. Select k random points from the given data set and denote them as initial centroids.
2. Calculate the distance from each data point to the k centroids.
3. Form K clusters and assign each point to its closest centroid.
4. Recalculate the new centroids by taking the average of all the data point in the cluster.
5. Repeat the process until the centroids do not change anymore.

Pseudocode:

```
for i in range(k)
{
   Generate random index from dataset;
   centroids[i] = data[random_index]
}

while (Distance between old and new centroids)>0:
   for i in data
{
      distances = [Euclidean distance between data[i] and centroids]
      nearest_centroid = Minimum(distances)
      clusters[i] = nearest_centroid

   old_centroids = centroids
   for i in clusters:
      centroids[i] = average of clusters

}
```

For the iris dataset the centroids converged, and the clustering algorithm terminated after 9 iterations, grouping the data points into 3 clusters for different species of the iris flower. Whereas for the image dataset, the algorithm terminated after 54 iterations and the data points were grouped into 10 clusters representing each digit.

Data Visualization:

After the implementing the K-Means algorithm on the given datasets, I have used the Seaborn library to visualize the clustering results. After reducing the dimensionality of the data using PCA, I have used the scatterplot() function to plot the clusters that were formed after the algorithm terminated. The following are the output results for the two datasets.

Iris Dataset:

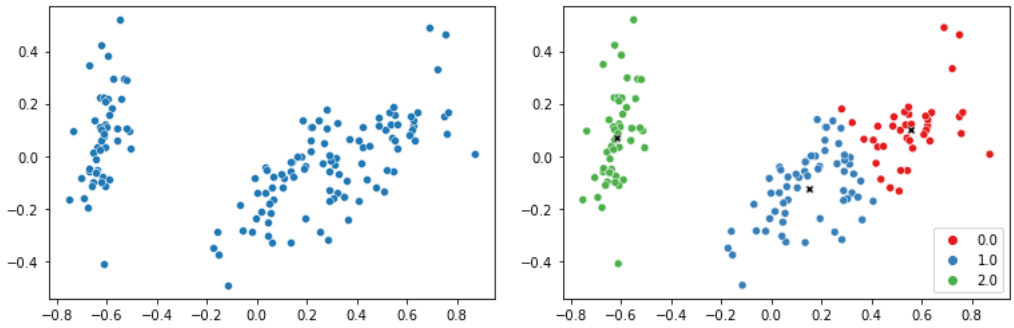
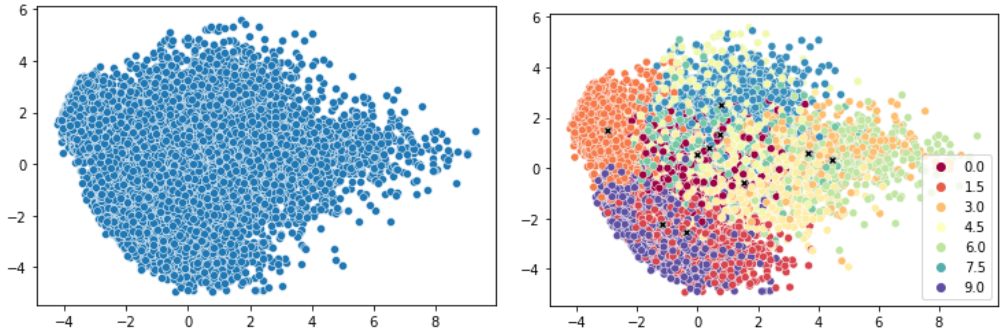


Image dataset:



References:

https://realpython.com/k-means-clustering-python/
https://scikit-learn.org/stable/modules/classes.html
https://towardsdatascience.com/how-does-k-means-clustering-in-machine-learning-work-fdaaaf5acfa0
https://machinelearningmastery.com/principal-components-analysis-for-dimensionality-reduction-in-python/
https://towardsdatascience.com/pca-using-python-scikit-learn-e653f8989e60