**CSC685 Advanced Machine Learning**

**Spring 2020**

**Project 6**

**Team 2**

**Jiajie Zhu & Getong Liu & Abby**

**Date: 04/15/2020**

**Part 1. Requirements**

In this project, the following items were performed.

1) Select 4 datasets

2) For each dataset, cluster the images using K-Means and plot the elbow graph to show the best K.

3) Train a classifier to determine which object is represented in each image and evaluate it on the validation set.

4) Use K-Means as a dimensionality reduction tool and train a classifier on the reduced set. Search for the number of clusters that provides the best performance.

5) Train a Gaussian mixture model on each dataset. To speed up the algorithm, use PCA dimensionality reduction (99% variation).

6) Graph results of your experiments for the different data sets and provide a qualitative assessment and detailed discussion of the achieved results.

**Part 2. Details**

**Introduction**

K-means is a method of cluster analysis which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean. It is similar to the expectation-maximization algorithm for mixtures of Gaussians in that they both attempt to find the centers of natural clusters in the data.

In addition, K-means also can work as a dimension reduction tool for machine learning. MNIST was used as the training dataset. Four of its subsets (5,0,1,4) are chosen to perform K-means clustering. For each subset, all the subset has a bunch of images tagged with the same handwritten digits.

Then, we utilize k-means clustering as a classifier and compare with another classifier Logistic Regression.

Later on, k-means clustering was performed as a dimension reduction technique in another classifier - Random Forest.

Gaussian Mixture Model was performed on the whole dataset MNIST as well as four selected subsets (5,0,1,4). All of these GMMs show convergence in the results.

**1) Select 4 datasets**

In this project we input the mnist dataset from keras as the big dataset. Then choose the **sub dataset of image 5, 0, 1, 4** from it. Each dataset has over 5000 images for training.

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

2) **For each dataset, cluster the images using K-Means and plot the elbow graph to show the best K.**

From the Silhouette Coefficient charts below we observed that actually, K = 2,3,4,5 are all good choices for clustering of sub dataset of image 5, 0, 1, 4, because all the clusters are higher than the dash line presented in each chart, but because of the chart of Silhouette score that K=2 has the highest value. Therefore, combined inertia and silhouette scores, **the best k is 2**.

A close up of a map

Description automatically generated A close up of a map

Description automatically generated

elbow graph for Sub Dataset:5 elbow graph for Sub Dataset:0

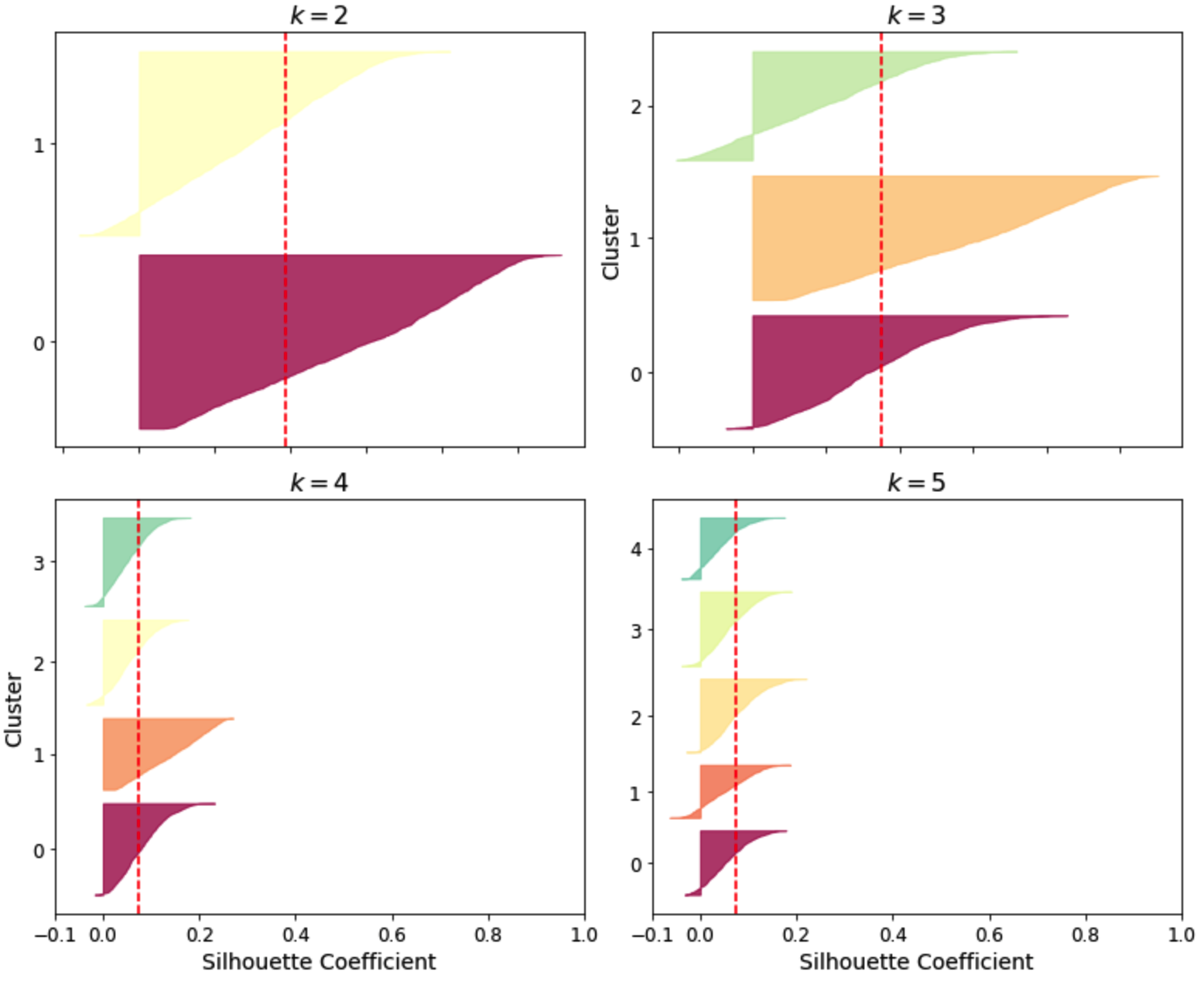
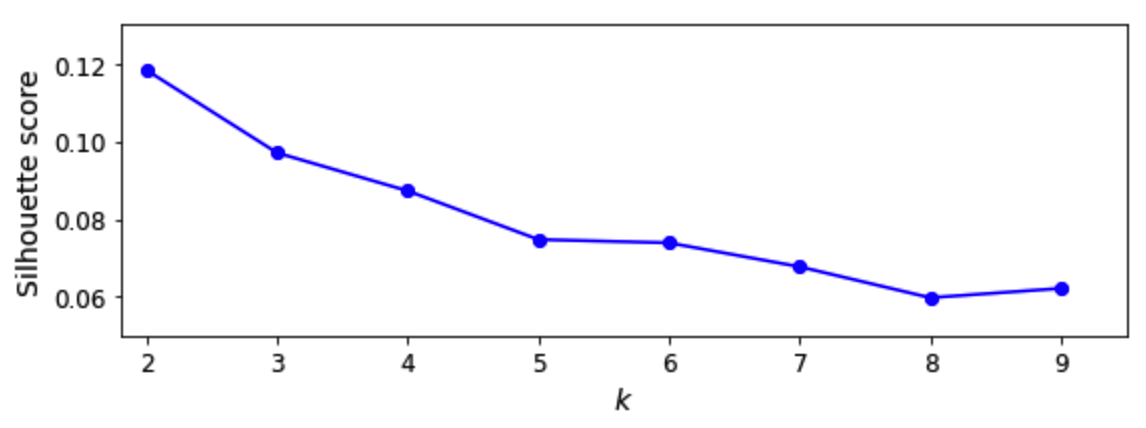
A close up of an object

Description automatically generated A picture containing map

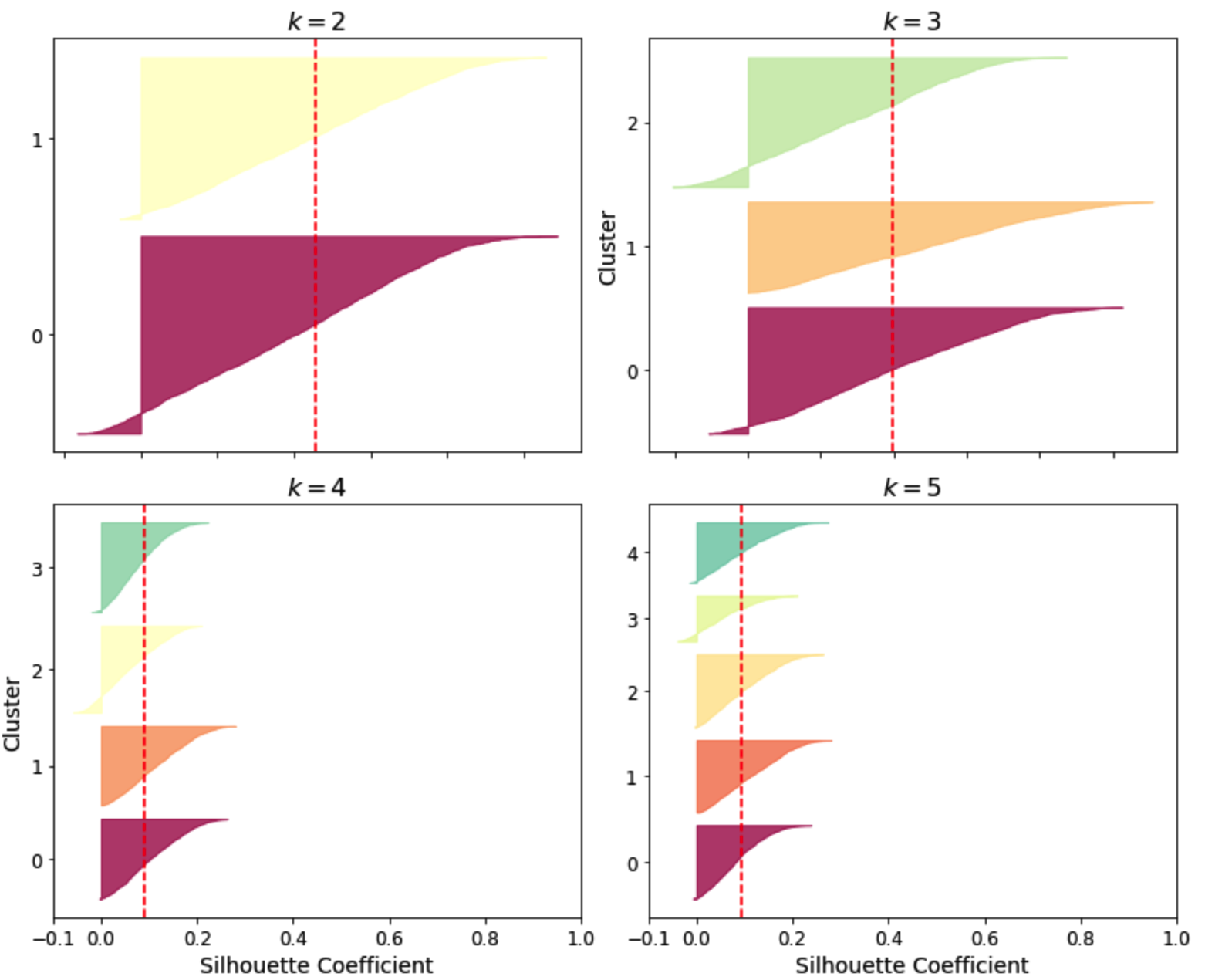
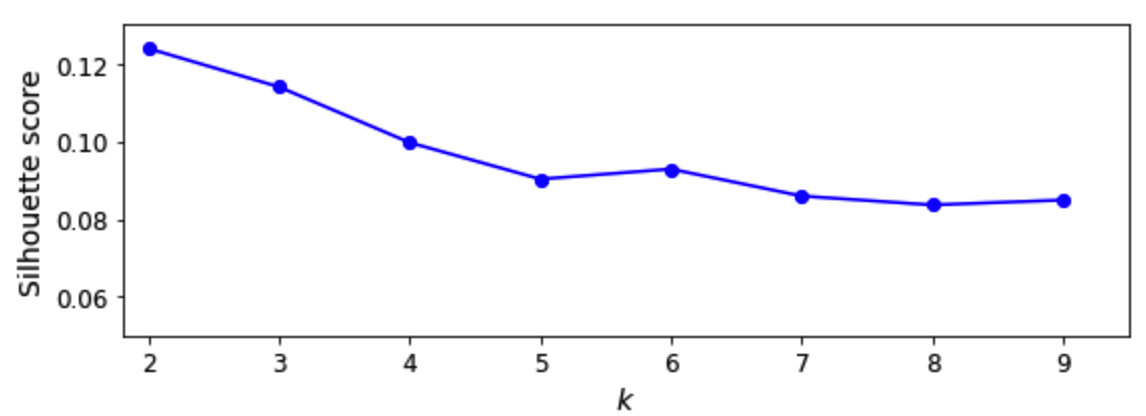
Description automatically generated

elbow graph for Sub Dataset:1 elbow graph for Sub Dataset:4

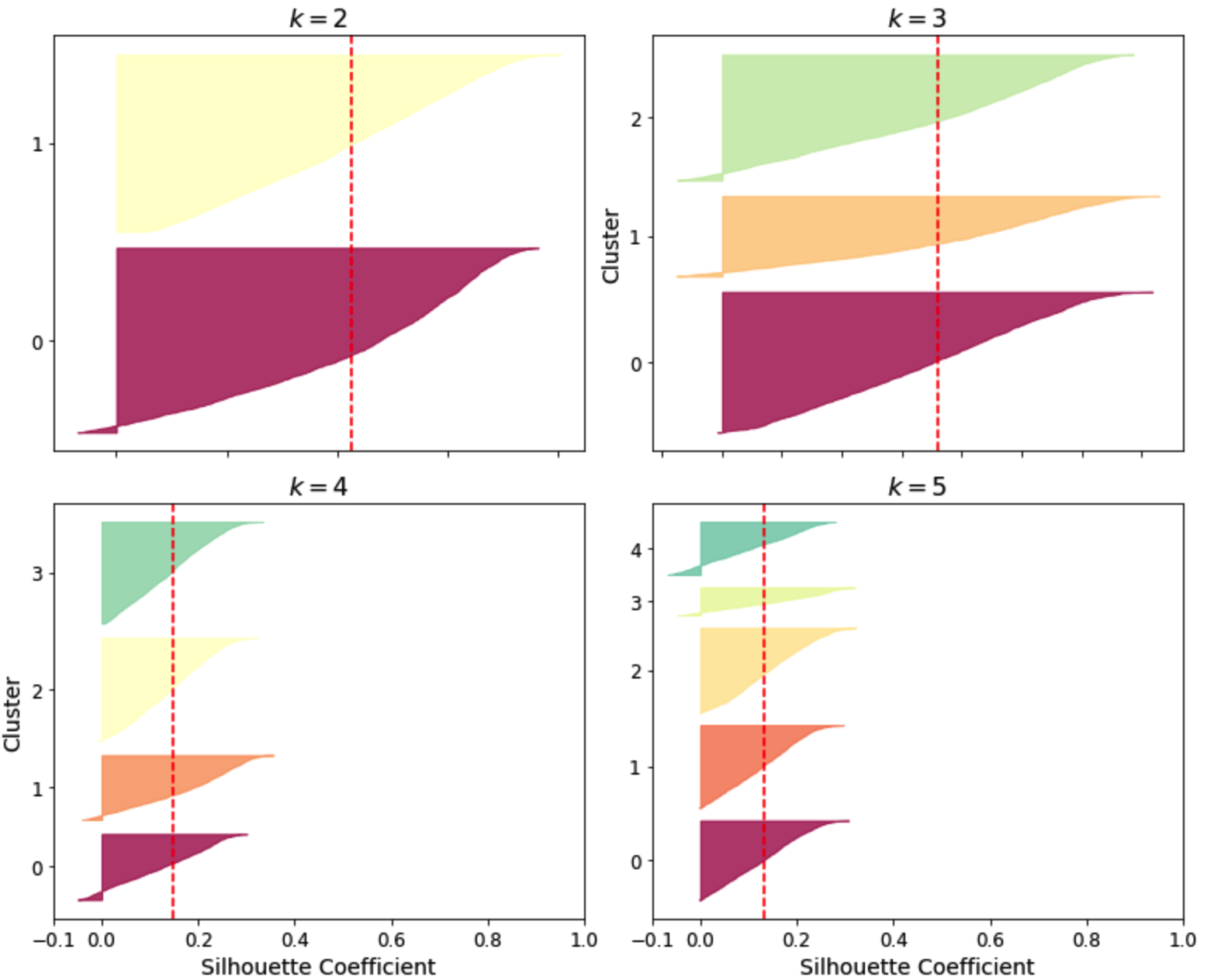
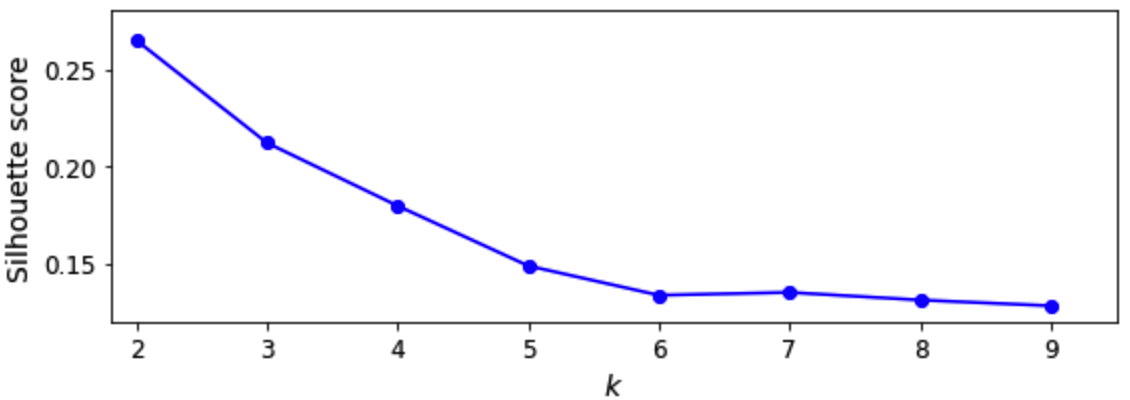
Silhouette Coefficient and silhouette scores for Sub Dataset:5:



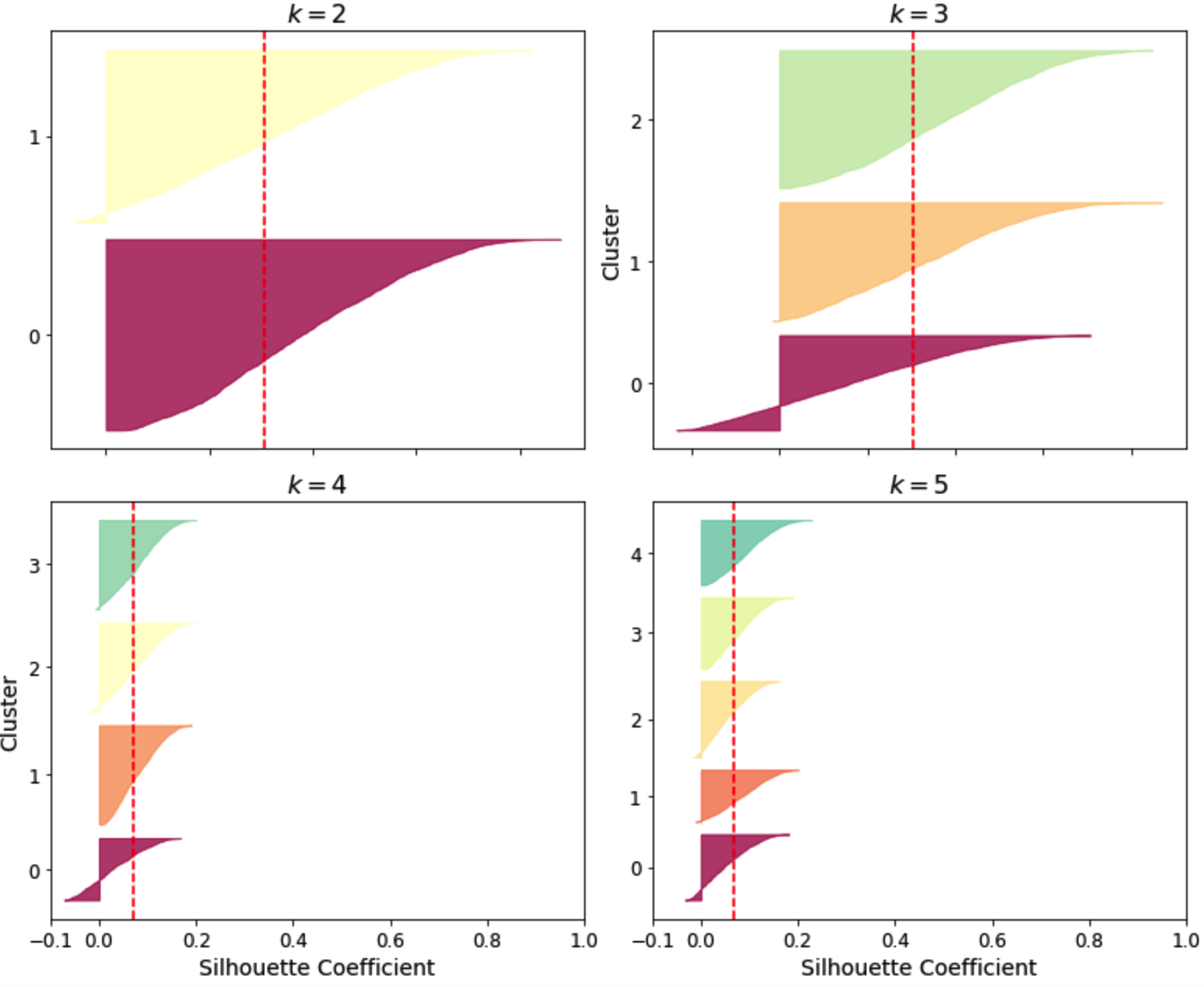
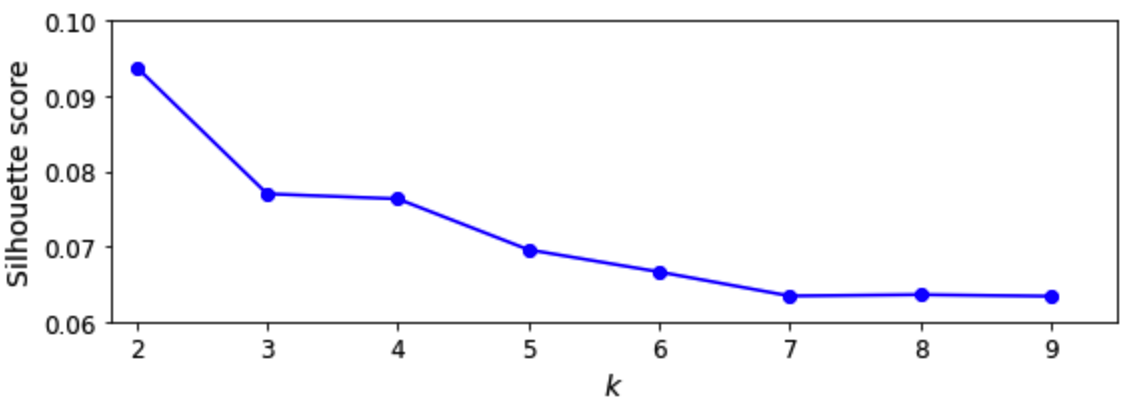
Silhouette Coefficient and silhouette scores for Sub Dataset:0:



Silhouette Coefficient and silhouette scores for Sub Dataset:1:



Silhouette Coefficient and silhouette scores for Sub Dataset:4:



**3) Train a classifier to determine which object is represented in each image and evaluate it on the validation set.**

We are training a classification model on the whole dataset instead of subsets and test its prediction rate. We use logistic regression as an example and the prediction accuracy is 91.98%. Then we used the K-means algorithm which clusters = 50, and the accuracy is 91.25%. Doesn’t show improvement. Therefore, we chose k within the interval of 10 between 10 to 100 with 3 times of cross validation to find the best k, when the accuracy score peaks. After 27 runs, from 10-90, the best value is 90, the accuracy increased from **91.98% to 93.37%.**

**4) Use K-Means as a dimensionality reduction tool and train a classifier on the reduced set. Search for the number of clusters that provides the best performance.**

First, we train a random forest classifier, and get the accuracy is 97.06%. Next, we test using K-mean (k=50) as a dimension reduction tool. By lowering the accuracy rate 3.68%, which is 93.38%.it did work on the task of dimension reduction.Then,we tried this again with the **best K=90** and checked the result.Well, when K get bigger like 90, it takes more time for transformation and boost the accuracy a little **better from 93.38% to 94.05%**, but still lower than the original dataset.

**5) Train a Gaussian mixture model on each dataset. To speed up the algorithm, use PCA dimensionality reduction (99% variation).**

First, we perform PCA dimension reduction with 99% variance remained. And the number of dimensions of the dataset is reduced from 784 to 331. Then we train Gaussian Mixture Model. And using the model to generate some new digits (inverse transform () method is used because, PCA is performed)).

Secondly, we try to modify some images (e.g., rotate, flip, darken) and see if the model can detect the anomalies (i.e., compare the output of the score samples() method for normal images and for anomalies).

A screenshot of a cell phone

Description automatically generated

Some of the bad digits are considered unlikely by the Gaussian Mixture model. Compare this to the scores of some training instances:

A screenshot of a cell phone

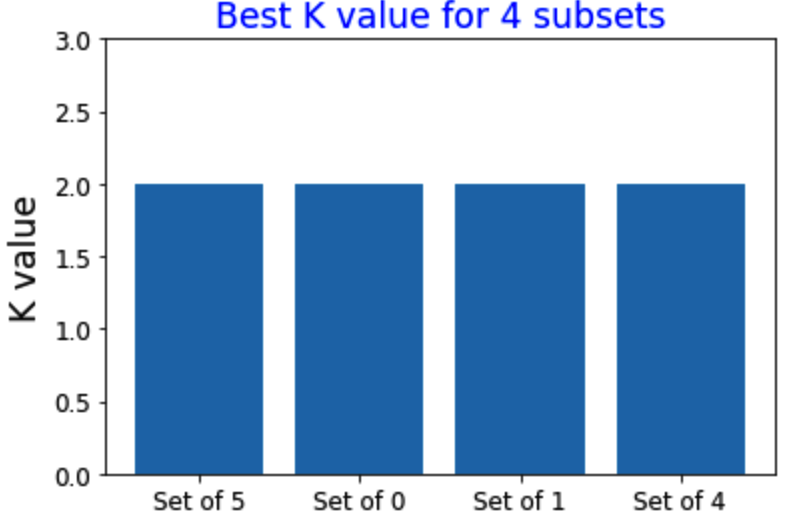
Description automatically generated

**GMMs for the subsets of 5、0、1、4:**

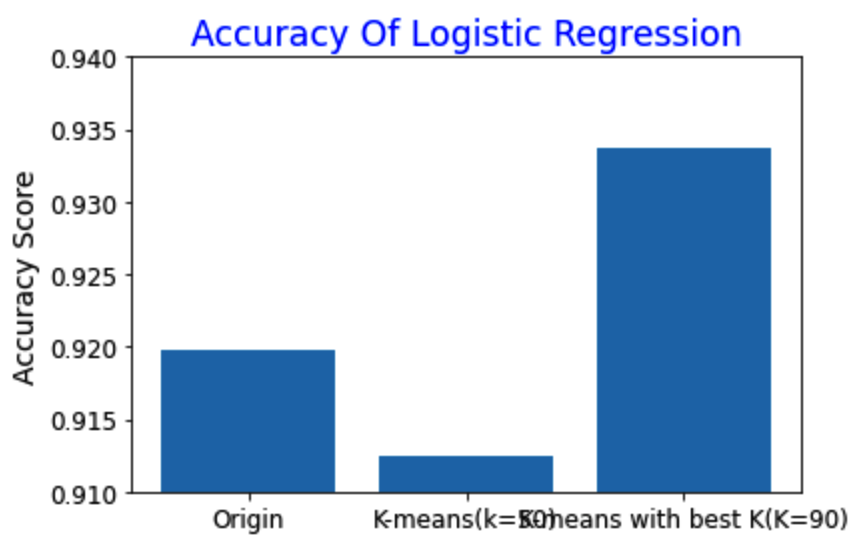
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | pca.n\_components\_ | gm.weights\_ | gm.n\_iter\_ | gm.predict(X\_train\_0) | gm.predict\_proba(X\_train\_0) | gm.score\_samples(X\_train\_0) |
| subset 0 | 270 | array([0.59210159, 0.40789841]) | 18 | array([0, 0, 0, ..., 0, 0, 0]) |  |  |
| subset 1 | 176 | array([0.64685982, 0.35314018]) | 29 | array([1, 0, 0, ..., 1, 1, 1]) |  |  |
| subset 4 | 287 | array([0.69513812, 0.30486188]) | 33 | array([1, 0, 1, ..., 1, 1, 0]) |  |  |
| subset 5 | 282 | array([0.35270245, 0.64729755]) | 20 | array([0, 1, 1, ..., 1, 1, 1]) |  |  |

**6) Graph results of your experiments for the different data sets and provide a qualitative assessment and detailed discussion of the achieved results.**

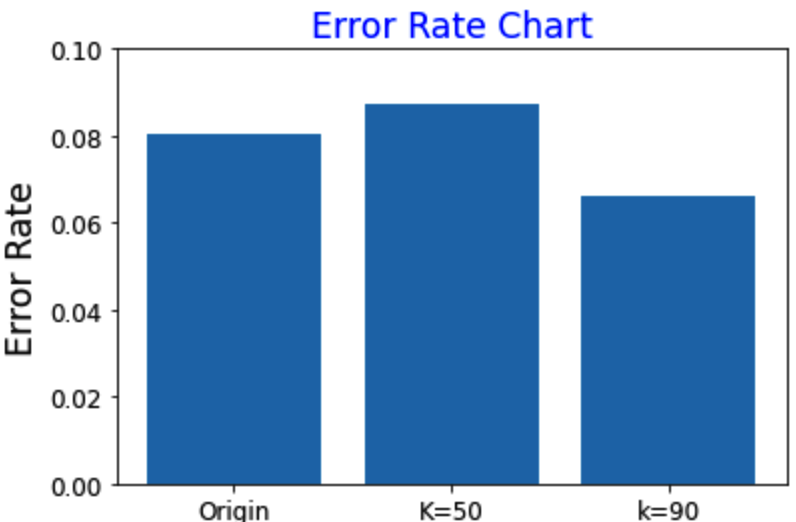
From the Silhouette Coefficient charts we observed K = 2,3,4,5 are all good choices for clustering. Next, we check the chart of the Silhouette score which shows that K=2 has the highest value for each subsets.In conclusion, all of the 4 subsets having the best K value equal to 2**.**

****

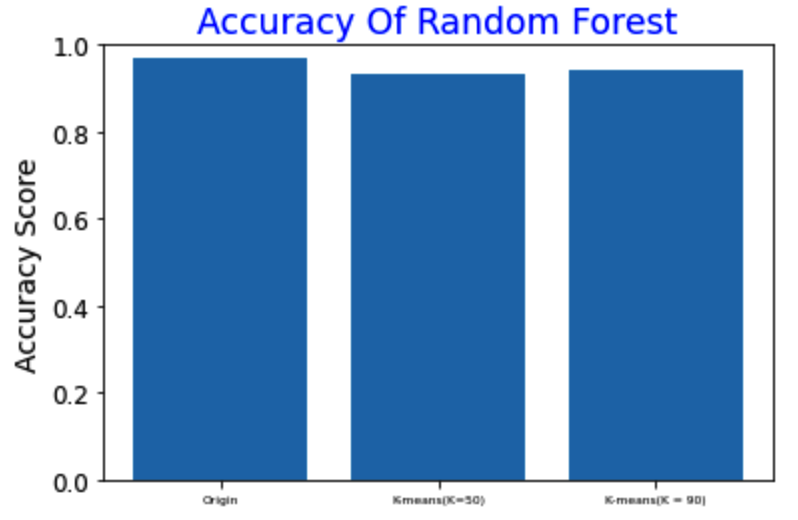
In this project, we first trained MNIST dataset with Logistic Regression model on the whole dataset instead of subsets and test its prediction rate. The prediction accuracy for original data is 91.98%.And then we implement the K-means algorithm as dimensional reduction tool. With cluster = 50, the accuracy doesn't show improvement, but get slice lower to 91.25%. Since the cluster number is arbitrary, we then chose k with in interval of 10 between 10 to 100 with 3 times of cross validation, and found the best K in 10-90. It took us more than 3 hours to get this best K. The accuracy increased from 91.98% to 93.37% with this best K.

****

By using K-means clustering, the error rate dropped 17.33%.

****

We've also trained the whole MNIST dataset with the Random Forest model. The prediction accuracy for original data is 97.06%, which is pretty high to start with.Then again, we implement the K-means algorithm as a dimensional reduction tool. With cluster = 50, the accuracy doesn't show improvement, but gets lower to 93.38%.And last we reduced dimension again with the K-means algorithm with cluster = 90, the best K. The accuracy increased to 94.05% from when K = 50. However, accuracy is still lower than the original data.



From the experiment, we observed that using K-means clustering as a dimension reduction tool, the training time even increased after the transformation. Therefore, also K-means is a feasible dimension reduction technique, it is not a suit for this dataset since it prolonged the training time and lowered the accuracy rate. Another aspect that needs to be taken into consideration is that the process of training a k-means transformation takes time as well. In this practice, take k=90, the training time for k-means transformation took 570.63s (9.5 minutes).

