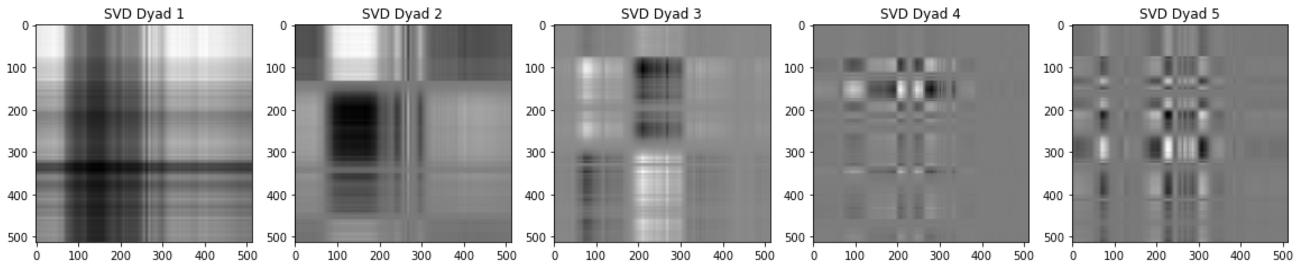
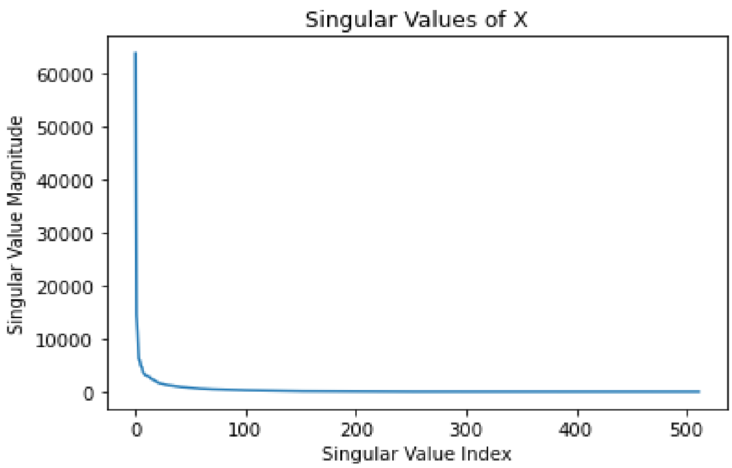
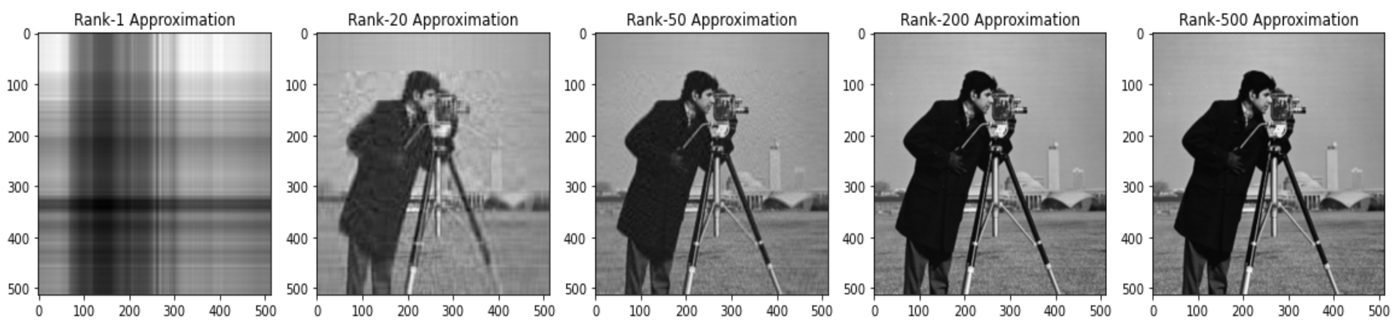
HomeWork 2_1

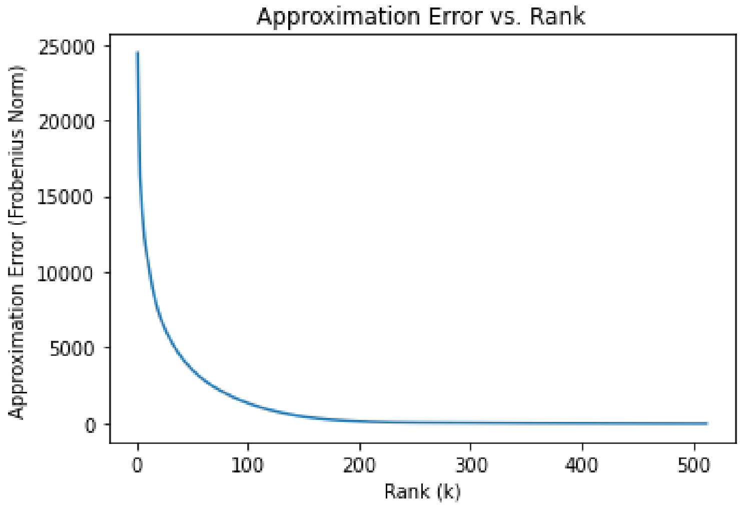
The true rank of image is: 512.
Image dimensions: 512x512
Dimensions of U: (512, 512)
Dimensions of Sigma: (512,)
Dimensions of V^T: (512, 512)
Difference (Frobenius norm): 2.426340581710773e-10



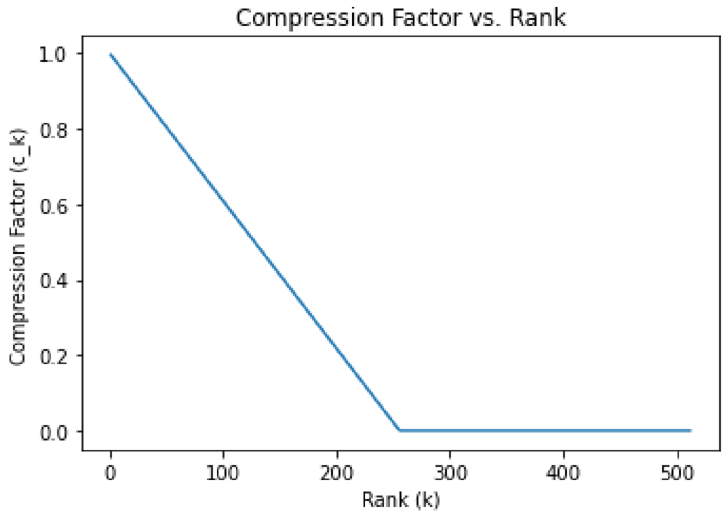
First singular value: 63868.99960191935 (Captures the most significant data pattern)
Second singular value: 14491.089714437125 (Represents the second most significant data pattern)
One hundred and first singular value: 267.13779646304965 (Much smaller, indicating less significant data patterns)
One hundred and first singular value: 0.2414182617783167 (Much smaller, indicating less significant data patterns)







(0, 0.5, 'Approximation Error (Frobenius Norm)')



Text(0, 0.5, 'Compression Factor(c_k)')

```
Rank k where c_k becomes zero: 256
Approximation Error (Frobenius Norm) when c_k = 0: 57.220171799640546
Approximation Error (Norm 2) when c_k = 0: 7.115457223551676
```

In practice, choosing a value of `k` that yields `c_k = 0` for SVD compression is not advantageous because:

1. We are not reducing the required storage space.
2. We might incur an approximation error despite theory suggesting there should be none.

The goal of compression is to reduce storage space while maintaining an acceptable level of approximation error. Typically, one selects a value of `k` that is

```
significantly less than the original dimensions of the image, thus balancing the
quality of the approximation with the benefit of compression.
`k = 100` seems to be a good choice to have a great trade-off.
```

Comparison of Rank-k Approximations for Two Different Images:

Observations Between Images and k Values: When comparing the rank-k approximations of two different images, it is observed that some images can be approximated more accurately than others with a lower rank-k, depending on the distribution of singular values. If an image has a few dominant singular values, it means that most of the information can be captured with a lower rank-k.

Relationship Between Dyad and Associated Singular Value:
Each singular value represents the "weight" of the information contained in the corresponding singular vector. The higher the singular value, the more significant the contribution of the associated singular vector to the original image. A high-rank dyad might thus be more "significant" and contain more details of the image.

Graph of Approximation Error vs. Singular Values: Plotting the approximation error (e.g., using the 2-norm) as a function of k and comparing it with the singular values, one should see that the error decreases as the singular values are included in the approximation. It might also be observed that the error drops rapidly at the beginning and then decreases more slowly, indicating that the initial singular values capture most of the information.

Behavior of Compression Factor c_k and Approximation Error: Behavior of c_k for Increasing Values of k:

The compression factor c_k will decrease as k increases, indicating that the image is less compressed. At high k, it approaches zero, corresponding to no compression (the approximated image has the same amount of information as the original).

Relationship Between Visual Quality and c_k:

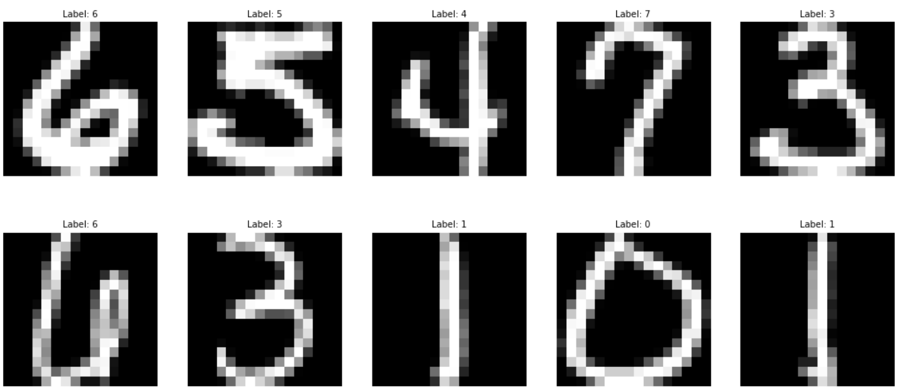
With increasing values of k, the visual quality of the approximated image X_k improves because more details are captured. However, there is a point beyond which additional improvements in k only lead to marginal increases in visual quality.

Approximation Error at c_k=0:

When c_k=0, there is no approximation error because all the original information is preserved (there is no reduction of information). However, due to the limited precision of numerical computation, there might still be a small error due to rounding.

HomeWork 2_2

```
(256, 319)
(256, 252)
(256, 202)
(256, 131)
(256, 122)
(256, 88)
(256, 151)
(256, 166)
(256, 144)
(256, 132)
[[0.     0.     0.     ... 0.     0.     0.    ]
 [0.     0.     0.     ... 0.     0.     0.    ]
 [0.     0.     0.     ... 0.     0.     0.    ]
 ...
 [0.     0.1645 0.     ... 0.     0.     0.    ]
 [0.     0.086  0.     ... 0.     0.     0.    ]
 [0.     0.     0.     ... 0.     0.     0.    ]]
[[6 5 4 ... 7 9 8]]
X dimensions: (256, 1707)
I dimensions: (1, 1707)
```




```
Number of columns for digits 3 and 4: 253
Dimensions of X after extraction: (256, 253)
Dimensions of I after extraction: (1, 253)
X_train shape: (256, 202)
```

```
I_train shape: (1, 202)
X_test shape: (256, 51)
I_test shape: (1, 51)
Dimensions of X1 (digit 3): (256, 105)
Dimensions of X2 (digit 4): (256, 97)
Dimensions of U1: (256, 105)
Dimensions of S1: (105,)
Dimensions of V1T: (105, 105)
Dimensions of U2: (256, 97)
Dimensions of S2: (97,)
Dimensions of V2T: (97, 97)
```

Implications of SVD on the MNIST Dataset

Subspace of R ^256: The column space of X_1 is a subspace of R^256. Each column of X_1 represents a vector in R^256(since each image is flattened into a 256-element vector), and the column space is formed by all possible linear combinations of these columns.

- Dimensionality of Column Space: The dimension of this column space is 106, indicated by the number of linearly independent columns in X_1. This implies that there are 106 independent directions or dimensions in the column space of X_1.
- Orthogonal Basis in U1: The matrix U1, obtained from the SVD of X_1, contains the first 106 columns that form an orthogonal basis for this column space. Each column of U1 is a vector in R^256, and together they represent all the necessary directions to construct any vector in the column space of X_1.
- Practical Implications: This means that although the original images are vectors in a high-dimensional space R^256, their actual variety and the information they carry can be captured and represented in a lower-dimensional subspace (106 dimensions in this case). This allows us to have an efficient representation and dimensionality reduction of data while preserving most of the relevant information.

Randomly selected index: 35

The selected image rappresent a : 3

Shape of the orthogonal projection onto space of digit 3 (y_ort_1): (256, 1)

Shape of the orthogonal projection onto space of digit 4 (y_ort_2): (256, 1)

```
Distance from y to its projection onto space of digit 3: 1.2985815617333103
Distance from y to its projection onto space of digit 4: 3.848340655659126
Y is a 3!
(1, 51)
Number of experiment done: 51
Number of misclassifications: 0
Misclassifications for digits 0 and 2: 35
```

```
Error rate for digits 0 and 2: 33.33333333333333 %
Top 15 pairs with highest misclassifications rates:
Couple 0-8 : Error Rate = 35.48 %
Couple 0-9 : Error Rate = 35.16 %
Couple 0-2 : Error Rate = 33.33 %
Couple 0-3 : Error Rate = 33.33 %
Couple 0-4 : Error Rate = 28.09 %
Couple 0-7 : Error Rate = 27.84 %
Couple 0-6 : Error Rate = 26.60 %
Couple 1-9 : Error Rate = 25.97 %
Couple 0-5 : Error Rate = 24.39 %
Couple 2-5 : Error Rate = 20.69 %
Couple 1-8 : Error Rate = 11.25 %
Couple 5-8 : Error Rate = 10.64 %
Couple 1-4 : Error Rate = 9.33 %
Couple 3-5 : Error Rate = 9.09 %
Couple 1-5 : Error Rate = 8.82 %
```

- Greater Variability for "0": The larger number of images for the digit "0" in the dataset 319 of 1707 may introduce greater variability in terms of writing styles. This increased variability can make it more challenging for the algorithm to capture a consistent and distinctive representation of this digit.
- Dataset Imbalance: Overrepresentation of a certain digit in the dataset could impact the learning process and subsequent classification.
- Impact on U Matrices: During SVD, a higher number of images might lead to a more complex or varied U matrix for that digit, affecting the algorithm's ability to accurately distinguish that digit from others.

Top 15 triples with highest misclassifications rates:

Couple (0, 8, 9) : Error Rate = 57.98 %

Couple (0, 7, 8) : Error Rate = 54.76 %

Couple (0, 2, 7) : Error Rate = 54.35 %

Couple (0, 6, 7) : Error Rate = 50.78 %

Couple (0, 4, 9) : Error Rate = 50.43 %

Couple (0, 2, 9) : Error Rate = 50.38 %

Couple (0, 5, 7) : Error Rate = 49.57 %

Couple (0, 2, 8) : Error Rate = 48.12 %

Couple (0, 3, 8) : Error Rate = 47.90 %

Couple (0, 2, 6) : Error Rate = 47.41 %

Couple (0, 6, 8) : Error Rate = 47.15 %

Couple (0, 4, 5) : Error Rate = 46.23 %

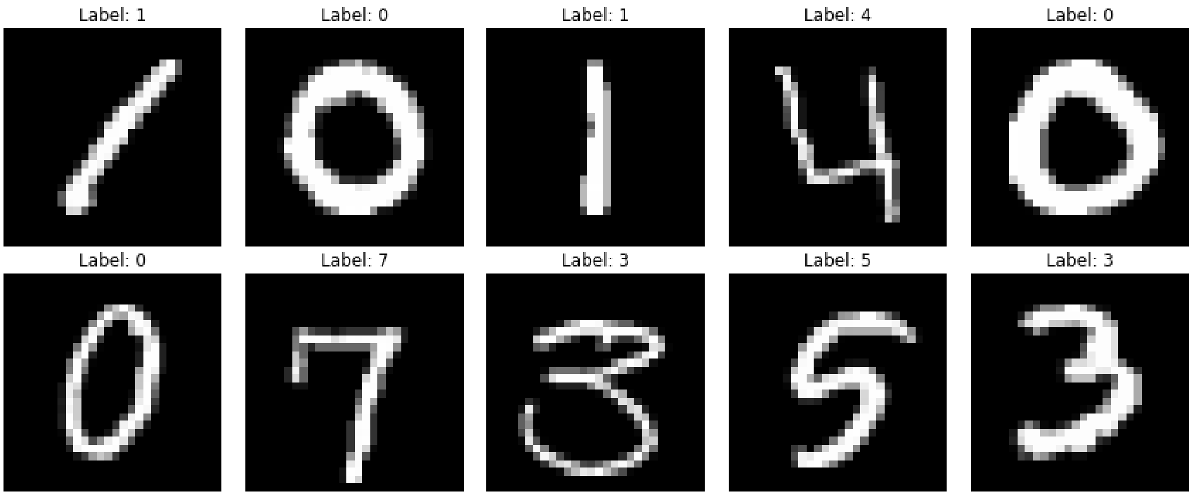
Couple (0, 7, 9) : Error Rate = 45.97 %

Couple (0, 4, 7) : Error Rate = 45.90 %

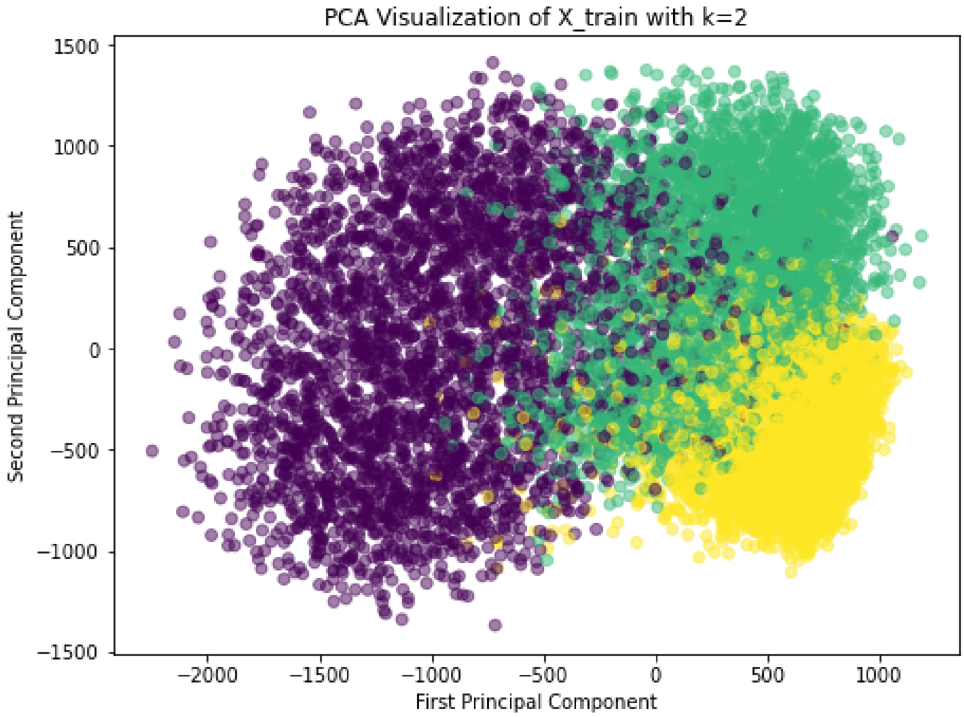
Couple (0, 2, 3) : Error Rate = 45.80 %

HomeWork 2_3

```
Shape of the data: (42000, 785)
Shape of X: (784, 42000)
Shape of Y: (42000,)
```

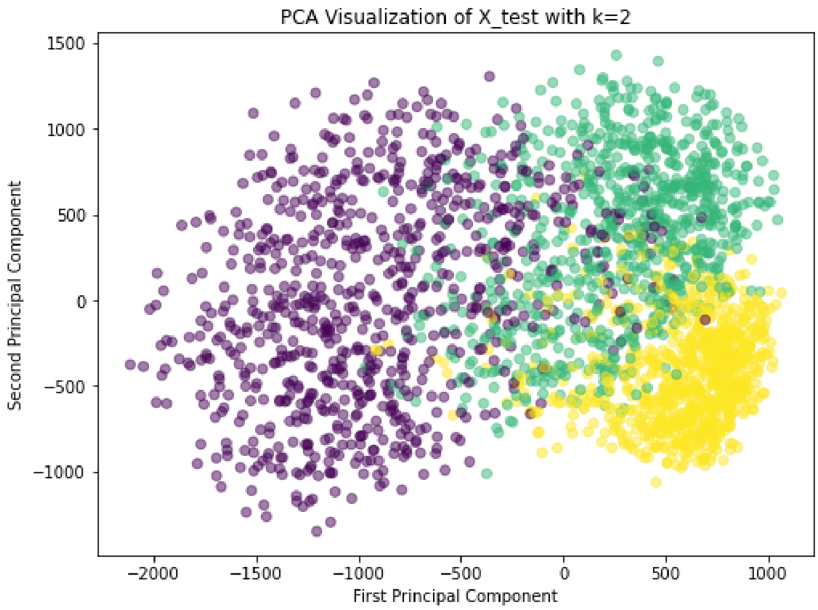



```
Shape of filtered X: (784, 12457)
Shape of filtered Y: (12457,)
X_train shape: (784, 9965)
Y_train shape: (9965,)
X_test shape: (784, 2492)
Y_test shape: (2492,)
Shape of centered data: (784, 9965)
Norm of the centroid of the centered data (should be close to 0): 1.44e-12
Shape of U: (784, 784)
Shape of s: (784,)
Shape of VT: (784, 9965)
Shape of U_k (first 2 columns of U): (784, 2)
Shape of Z_k (projected data in 2 dimensions): (2, 9965)
```

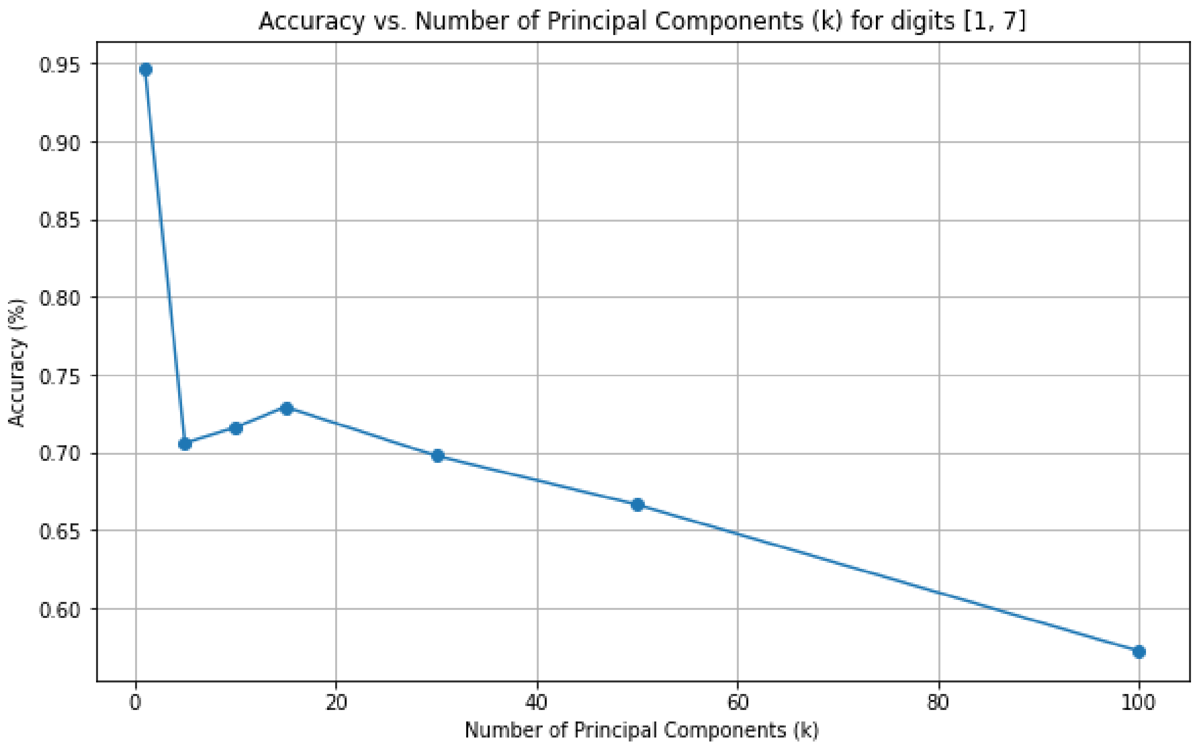


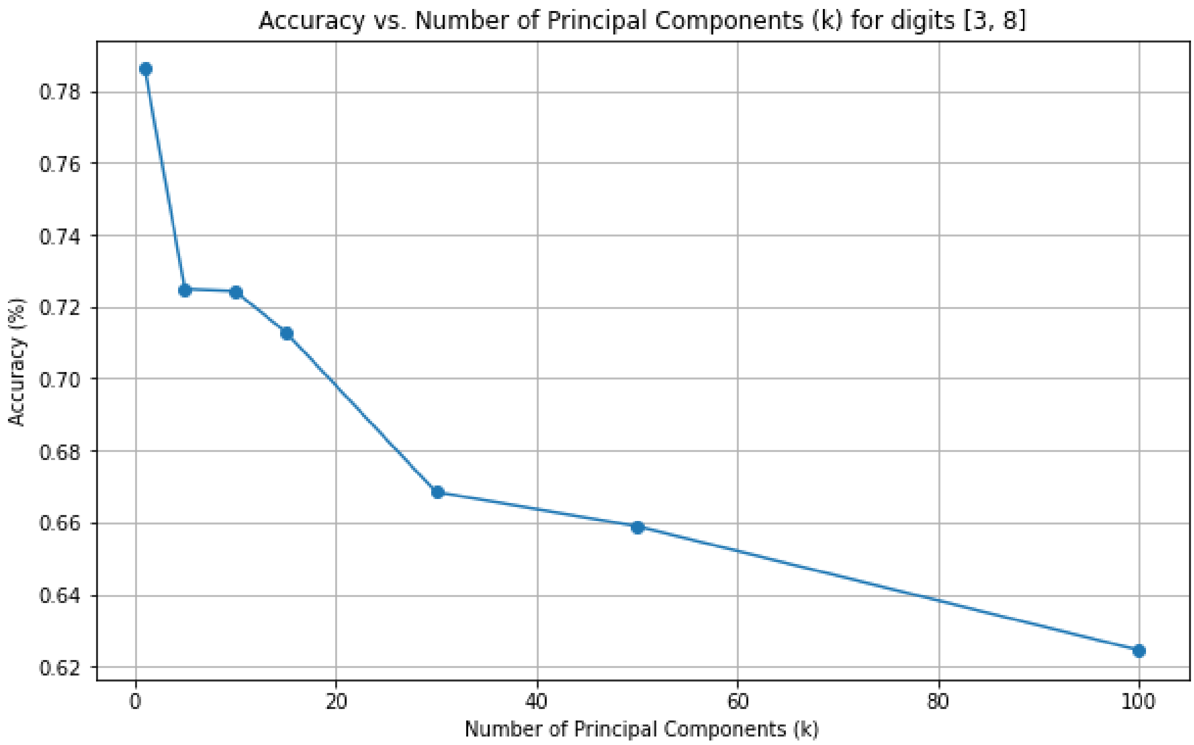
Average distances for training clusters: [738.4934739517306, 537.8551605675603, 365.6178024112473]

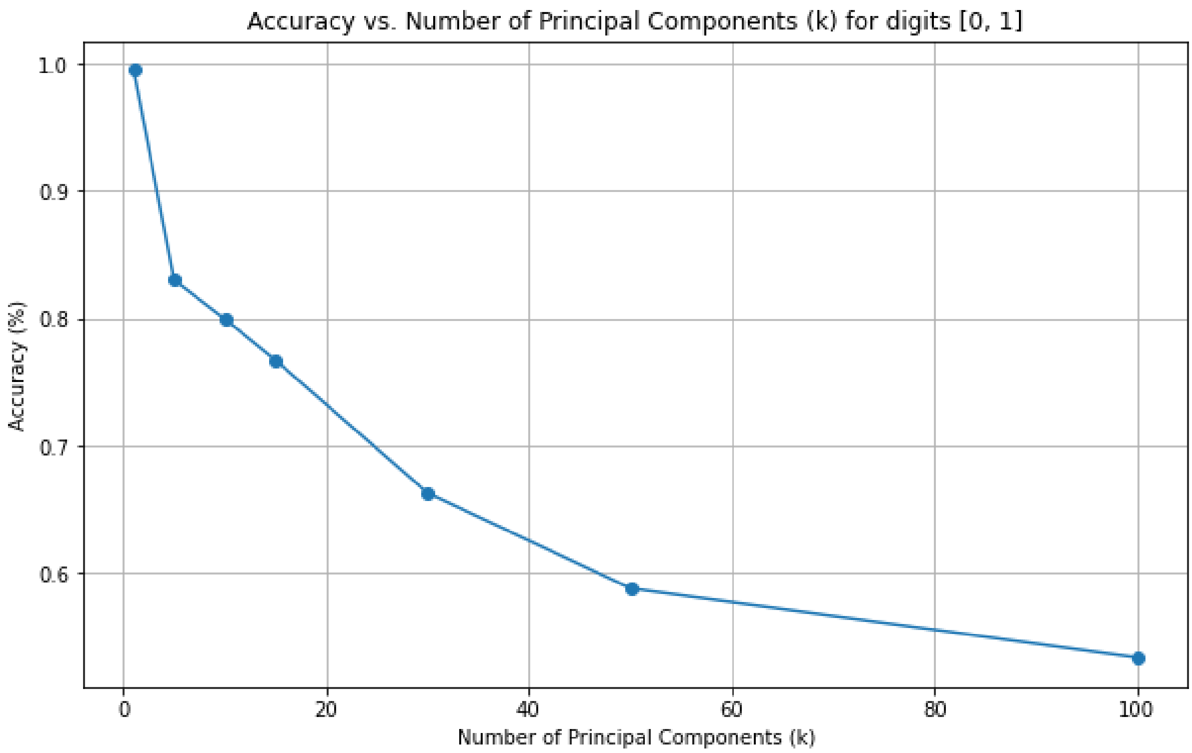
```
(784, 2492)
Centroid of X_test is : 37.709718819286195.
(2, 2492)
```

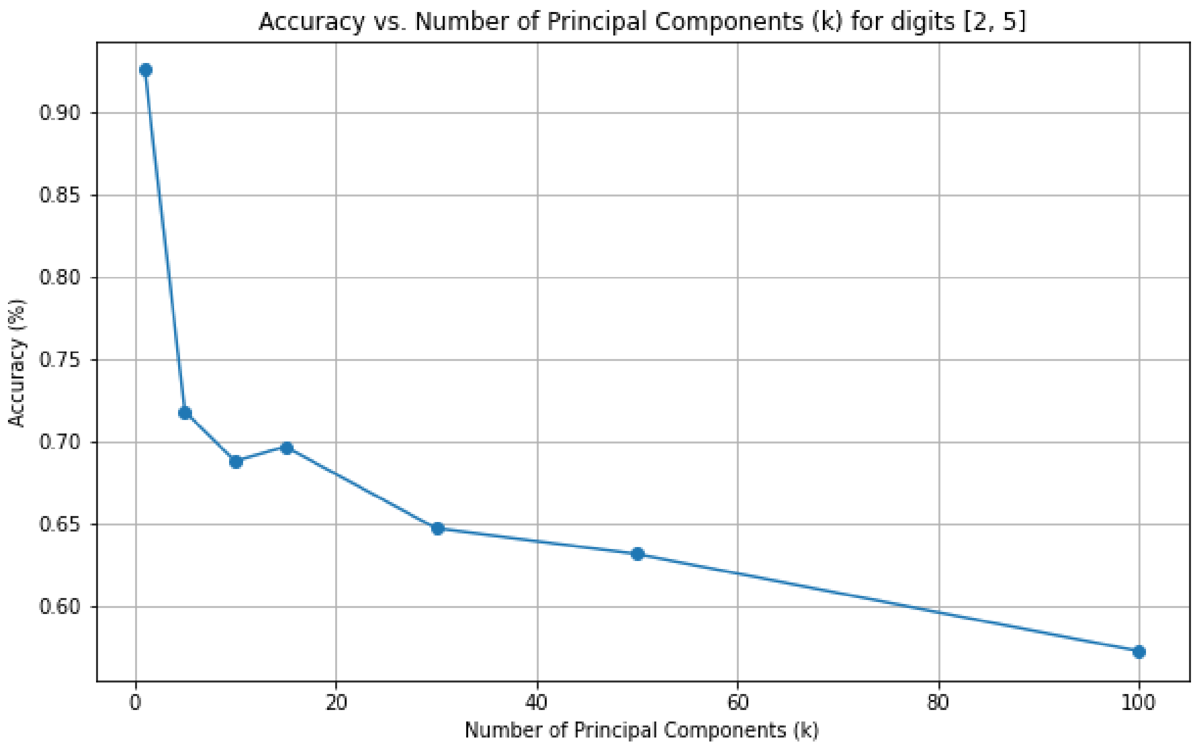


Average distances for test clusters: [736.480765945645, 559.9044888051794, 368.1653382835122]


Accuracy vs. Number of Principal Components (k) for digits [1, 7]
Accuracy (%)
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0
20
40
60
80
100
Number of Principal Components (k)


Accuracy vs. Number of Principal Components (k) for digits [3, 8]
Accuracy (%)
0.78
0.76
0.74
0.72
0.70
0.68
0.66
0.64
0.62
0
20
40
60
80
100
Number of Principal Components (k)


Accuracy vs. Number of Principal Components (k) for digits [0, 1]
Accuracy (%)
1.0
0.9
0.8
0.7
0.6
0
20
40
60
80
100
Number of Principal Components (k)


Accuracy vs. Number of Principal Components (k) for digits [2, 5]
Accuracy (%)
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0
20
40
60
80
100
Number of Principal Components (k)

Accuracy Variation with Number of Principal Components k

As the number of principal components k used in PCA increases, the accuracy of the associated classifier might initially improve, capturing more relevant information from the data. However, this increase in accuracy is not always indefinite. After reaching a certain number of components, the accuracy might start to decrease or stabilize. This can occur due to several factors:

- **Overfitting**: Including too many components might lead to the model fitting noise in the data rather than the actual underlying patterns, which can degrade its performance on test data.

- **Increased Model Complexity**: More components increase the model's complexity, potentially making it harder to generalize to new, unseen data.

- **Irrelevant Information**: Subsequent principal components may not significantly contribute to class discrimination and might introduce irrelevant or distracting information.

Finding the optimal number of principal components in PCA involves balancing dimensionality reduction and retaining relevant information for accurate classification.