# **HW10** PCA and KMeans

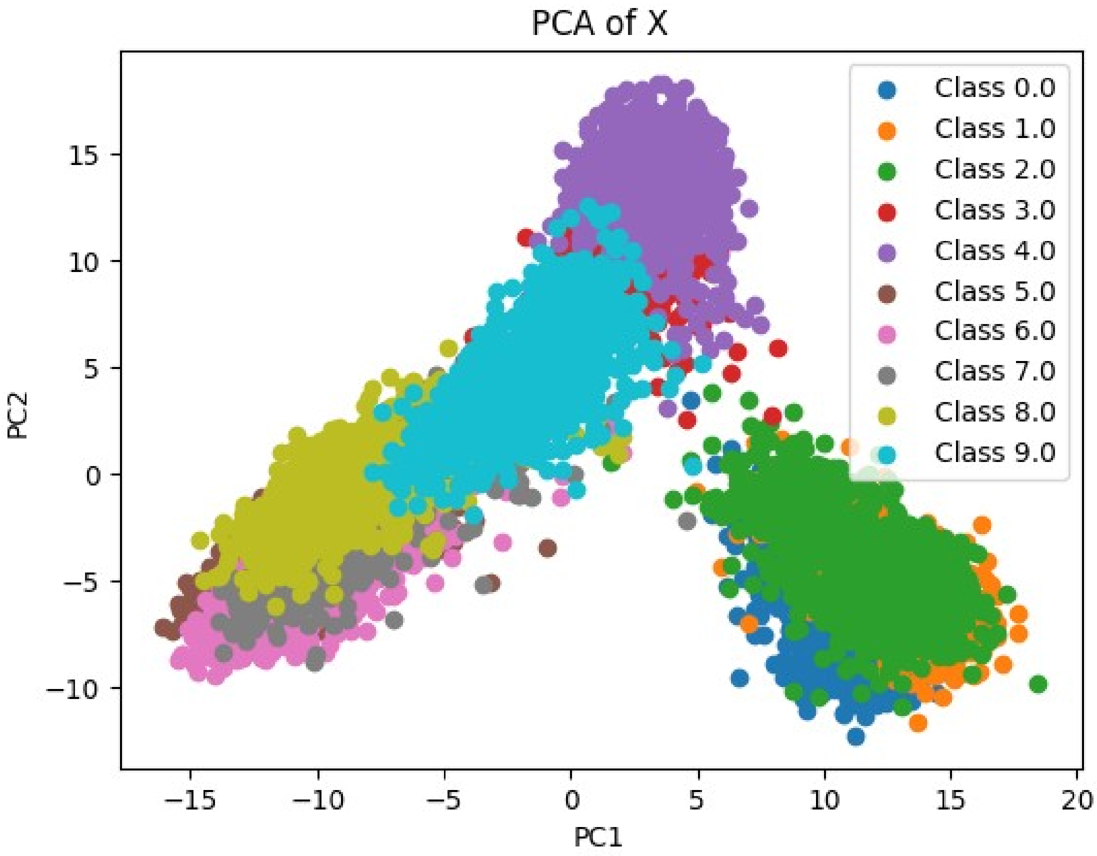
## As always start with importing what we need and finding our data locations.

```python
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.io
from scipy.spatial.distance import cdist
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
from sklearn.metrics.cluster import contingency_matrix
from sklearn.metrics import adjusted_rand_score
from scipy.optimize import linear_sum_assignment
import pandas as pd

import os

os.chdir("C:/Users/rique/Downloads/datasets")

def load_dataset():

    ds = scipy.io.loadmat("data_clust.mat")
    X = ds['x']
    y = ds['y'].flatten()

    return X, y

X, y = load_dataset()

def pca(X,y):
    model = PCA(n_components=2)
    model = model.fit_transform(X)

    plt.figure()
    for i in np.unique(y):
        plt.scatter(model[y == i, 0], model[y == i, 1], label=f'Class {i}')

    plt.xlabel('PC1')
    plt.ylabel('PC2')
    plt.title('PCA of X')
    plt.legend()
    plt.show()
```

```python
def Kmeans(X,y, init, iterations):
    accuracies = []
    aris = []
    for i in range(iterations):
        kmeans = KMeans(n_clusters=10, init=init, n_init=1,
random_state=i)
        y_pred = kmeans.fit_predict(X)

        cont_matrix = contingency_matrix(y, y_pred)
        row_ind, col_ind = linear_sum_assignment(-cont_matrix)
        permuted_cont_matrix = cont_matrix[:, col_ind]

        accuracy = np.sum(np.diag(permuted_cont_matrix)) /
np.sum(permuted_cont_matrix)
        ari = adjusted_rand_score(y, y_pred)

        accuracies.append(accuracy)
        aris.append(ari)

        # Display results
        print(f"Iteration {i+1} with {init}: Accuracy =
{accuracy:.4f}, Adjusted Rand Index = {ari:.4f}")

        # Plot original and permuted contingency matrices
        plt.figure(figsize=(12, 6))
        plt.subplot(1, 2, 1)
        sns.heatmap(cont_matrix, annot=True, fmt='d', cmap='gray')
        plt.title(f'Original Contingency Matrix: Iteration {i+1}')
        plt.xlabel('Predicted')
        plt.ylabel('True')

        plt.subplot(1, 2, 2)
        sns.heatmap(permuted_cont_matrix, annot=True, fmt='d',
cmap='gray')
        plt.title(f'Permuted Contingency Matrix: Iteration {i+1}')
        plt.xlabel('Predicted (permuted)')
        plt.ylabel('True')
        plt.show()

    return np.mean(accuracies), np.mean(aris)
```

# a) Using PCA

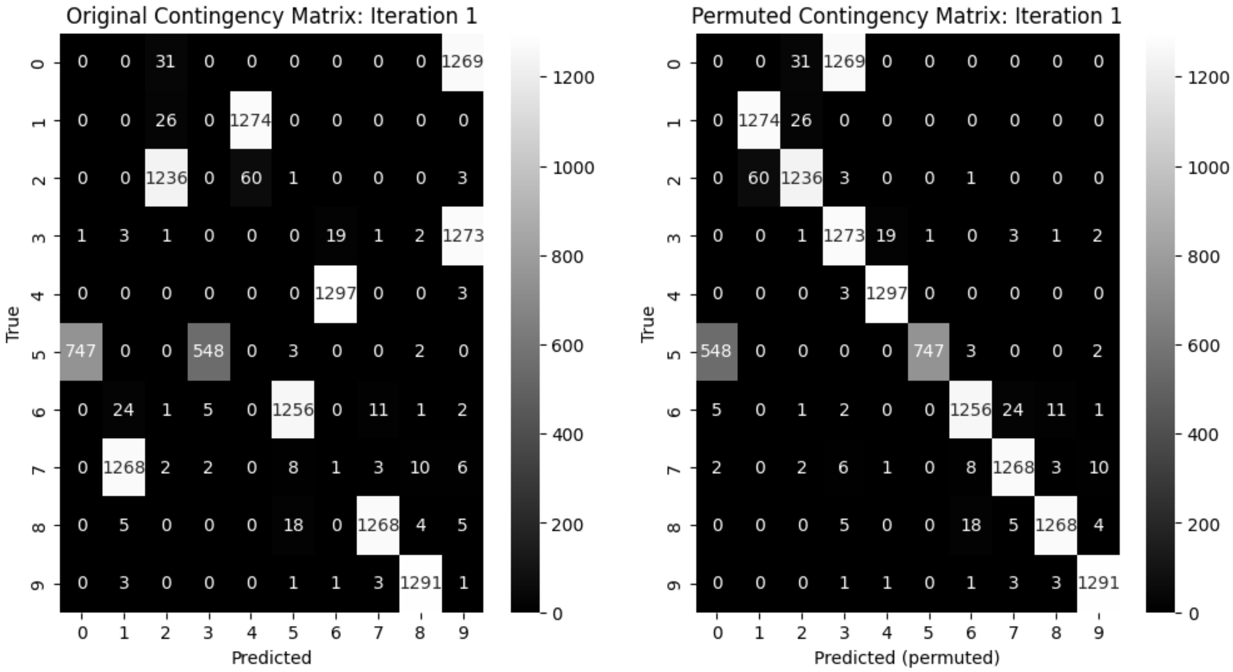
```python
pca(X,y)
```



## b) & c) using random initalizer

```
random_km_one = Kmeans(X,y,'random',1)
```

```
Iteration 1 with random: Accuracy = 0.8392, Adjusted Rand Index =
0.8308
```



Original Contingency Matrix: Iteration 1

|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 31 | 0 | 0 | 0 | 0 | 0 | 0 | 1269 |
| 1 | 0 | 0 | 26 | 0 | 1274 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 1236 | 0 | 60 | 1 | 0 | 0 | 0 | 3 |
| 3 | 1 | 3 | 1 | 0 | 0 | 0 | 19 | 1 | 2 | 1273 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 1297 | 0 | 0 | 3 |
| 5 | 747 | 0 | 0 | 548 | 0 | 3 | 0 | 0 | 2 | 0 |
| 6 | 0 | 24 | 1 | 5 | 0 | 1256 | 0 | 11 | 1 | 2 |
| 7 | 0 | 1268 | 2 | 2 | 0 | 8 | 1 | 3 | 10 | 6 |
| 8 | 0 | 5 | 0 | 0 | 0 | 18 | 0 | 1268 | 4 | 5 |
| 9 | 0 | 3 | 0 | 0 | 0 | 1 | 1 | 3 | 1291 | 1 |

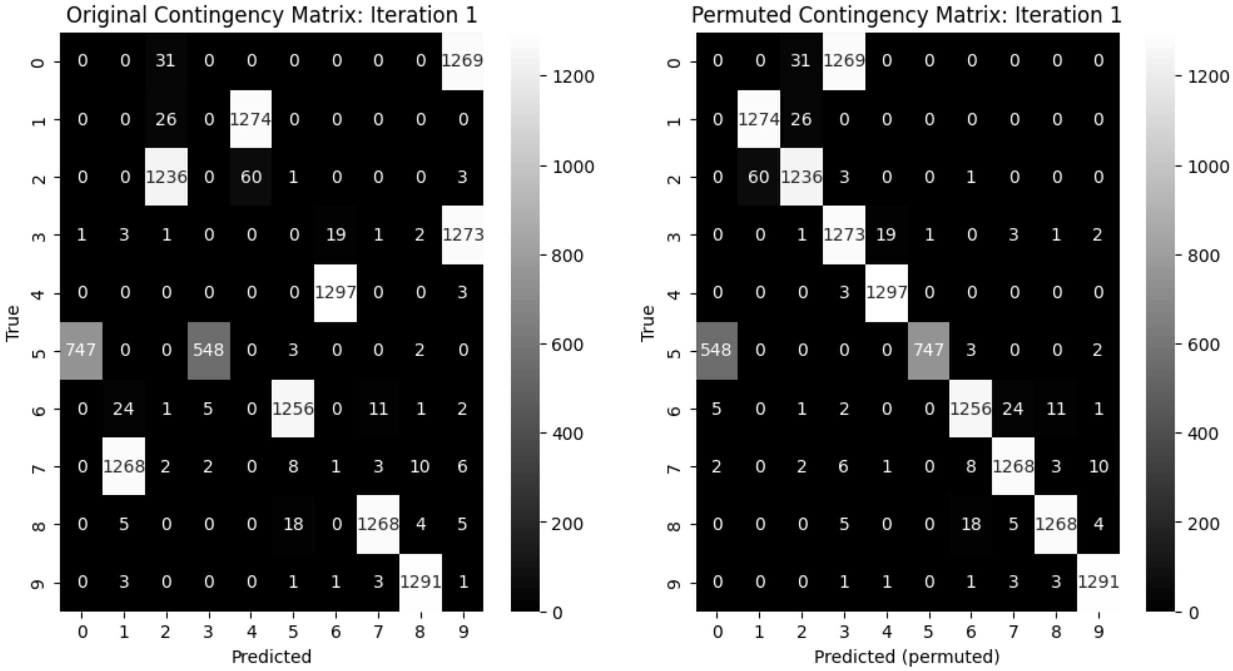
Permuted Contingency Matrix: Iteration 1

|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 31 | 1269 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1274 | 26 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 60 | 1236 | 3 | 0 | 0 | 1 | 0 | 0 | 0 |
| 3 | 0 | 0 | 1 | 1273 | 19 | 1 | 0 | 3 | 1 | 2 |
| 4 | 0 | 0 | 0 | 3 | 1297 | 0 | 0 | 0 | 0 | 0 |
| 5 | 548 | 0 | 0 | 0 | 0 | 747 | 3 | 0 | 0 | 2 |
| 6 | 5 | 0 | 1 | 2 | 0 | 0 | 1256 | 24 | 11 | 1 |
| 7 | 2 | 0 | 2 | 6 | 1 | 0 | 8 | 1268 | 3 | 10 |
| 8 | 0 | 0 | 0 | 5 | 0 | 0 | 18 | 5 | 1268 | 4 |
| 9 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 3 | 3 | 1291 |

# d) using random initalizer for 5 rounds

```
random_km_five = Kmeans(X,y,'random',5)
```

```
Iteration 1 with random: Accuracy = 0.8392, Adjusted Rand Index =
0.8308
```



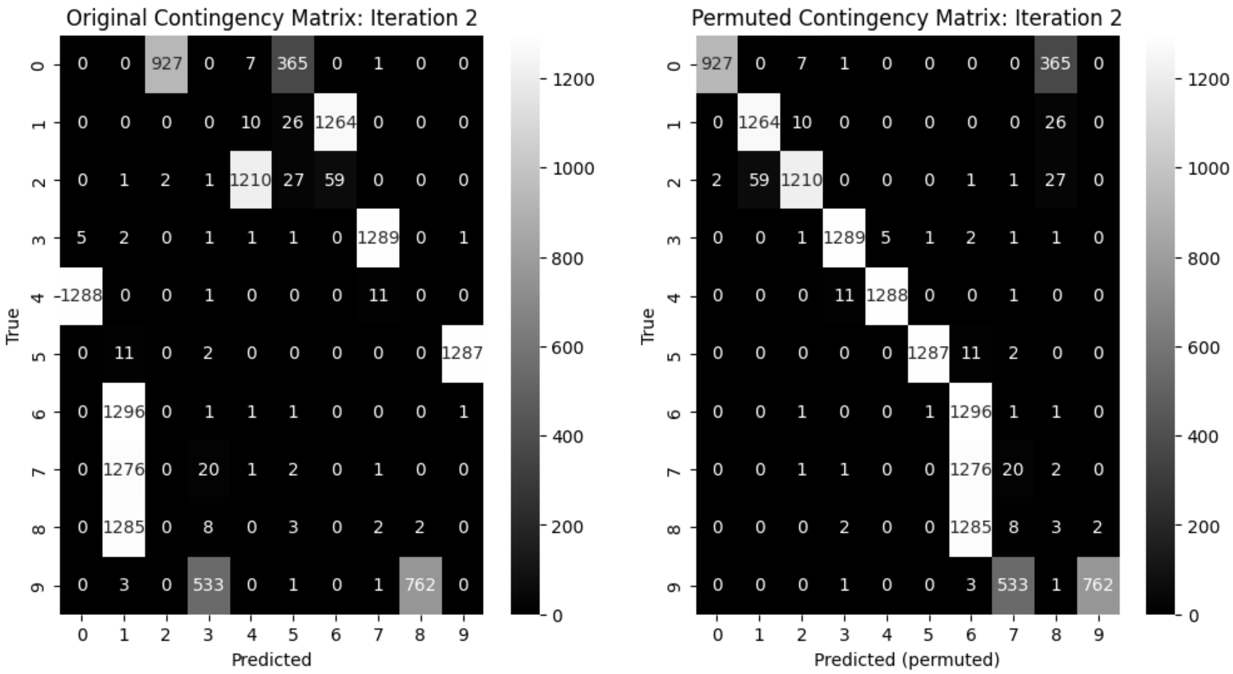
Original Contingency Matrix: Iteration 1

|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 31 | 0 | 0 | 0 | 0 | 0 | 0 | 1269 |
| 1 | 0 | 0 | 26 | 0 | 1274 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 1236 | 0 | 60 | 1 | 0 | 0 | 0 | 3 |
| 3 | 1 | 3 | 1 | 0 | 0 | 0 | 19 | 1 | 2 | 1273 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 1297 | 0 | 0 | 3 |
| 5 | 747 | 0 | 0 | 548 | 0 | 3 | 0 | 0 | 2 | 0 |
| 6 | 0 | 24 | 1 | 5 | 0 | 1256 | 0 | 11 | 1 | 2 |
| 7 | 0 | 1268 | 2 | 2 | 0 | 8 | 1 | 3 | 10 | 6 |
| 8 | 0 | 5 | 0 | 0 | 0 | 18 | 0 | 1268 | 4 | 5 |
| 9 | 0 | 3 | 0 | 0 | 0 | 1 | 1 | 3 | 1291 | 1 |

Permuted Contingency Matrix: Iteration 1

|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 31 | 1269 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1274 | 26 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 60 | 1236 | 3 | 0 | 0 | 1 | 0 | 0 | 0 |
| 3 | 0 | 0 | 1 | 1273 | 19 | 1 | 0 | 3 | 1 | 2 |
| 4 | 0 | 0 | 0 | 3 | 1297 | 0 | 0 | 0 | 0 | 0 |
| 5 | 548 | 0 | 0 | 0 | 0 | 747 | 3 | 0 | 0 | 2 |
| 6 | 5 | 0 | 1 | 2 | 0 | 0 | 1256 | 24 | 11 | 1 |
| 7 | 2 | 0 | 2 | 6 | 1 | 0 | 8 | 1268 | 3 | 10 |
| 8 | 0 | 0 | 0 | 5 | 0 | 0 | 18 | 5 | 1268 | 4 |
| 9 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 3 | 3 | 1291 |

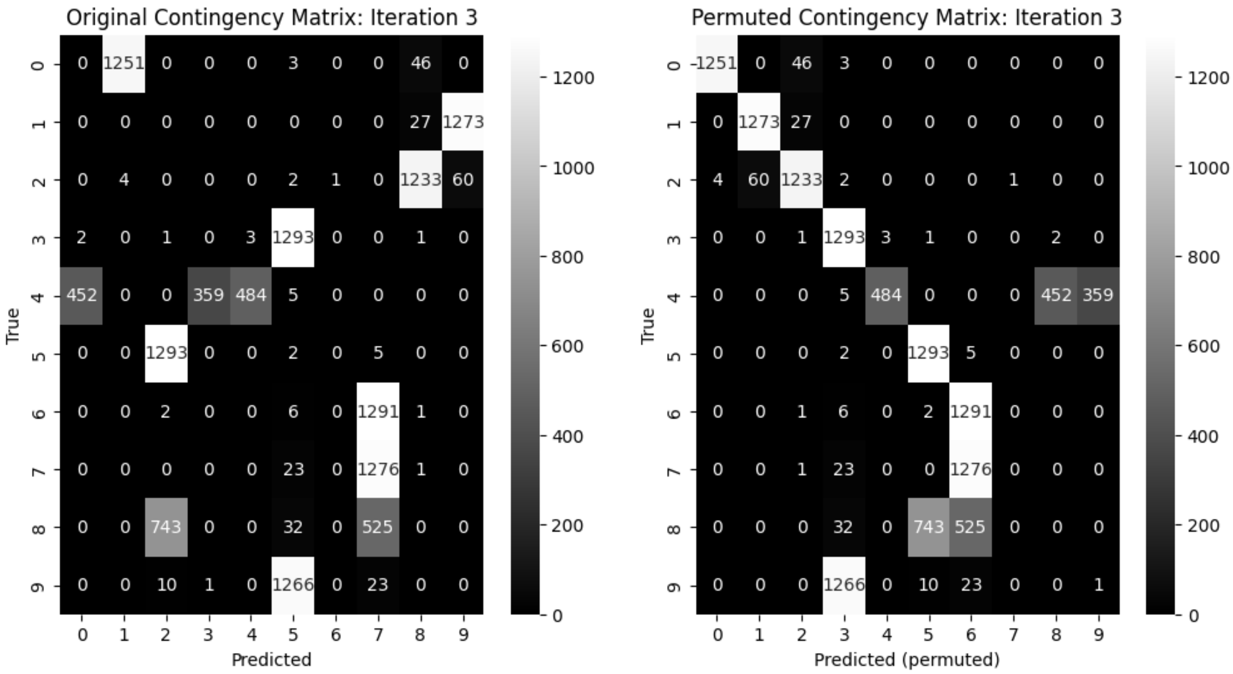
```
Iteration 2 with random: Accuracy = 0.7189, Adjusted Rand Index =
0.6647
```



Original Contingency Matrix: Iteration 2

Permuted Contingency Matrix: Iteration 2

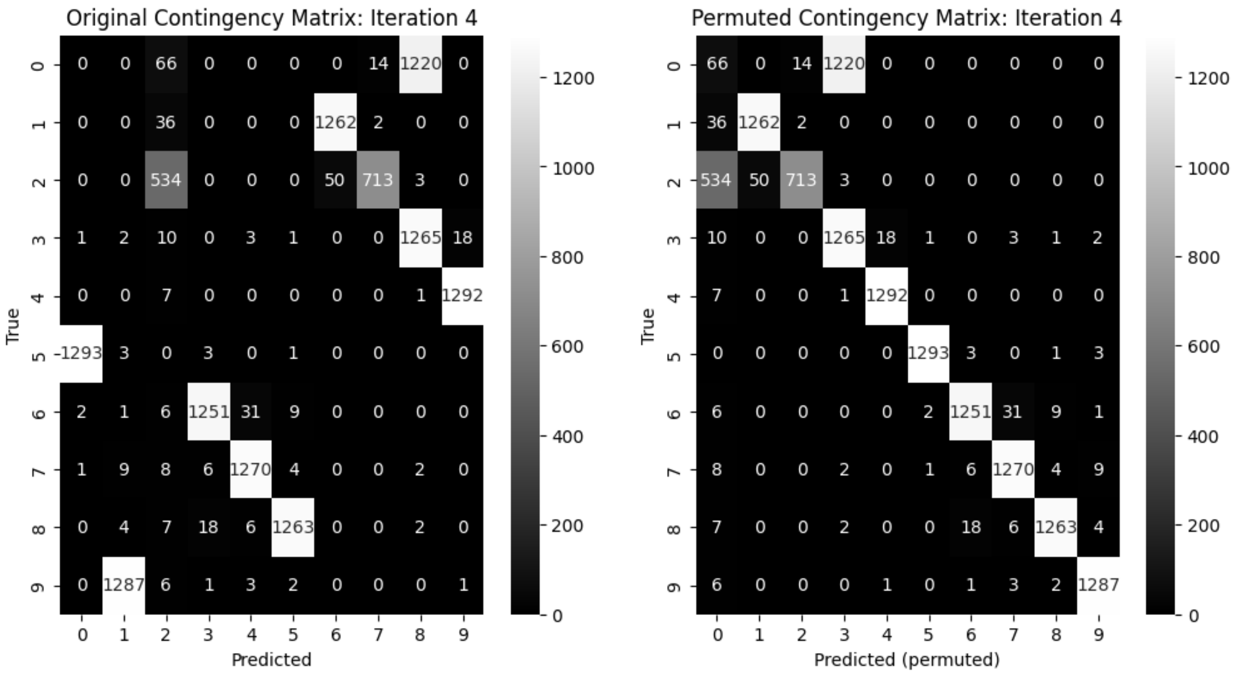
```
Iteration 3 with random: Accuracy = 0.6245, Adjusted Rand Index =
0.6141
```



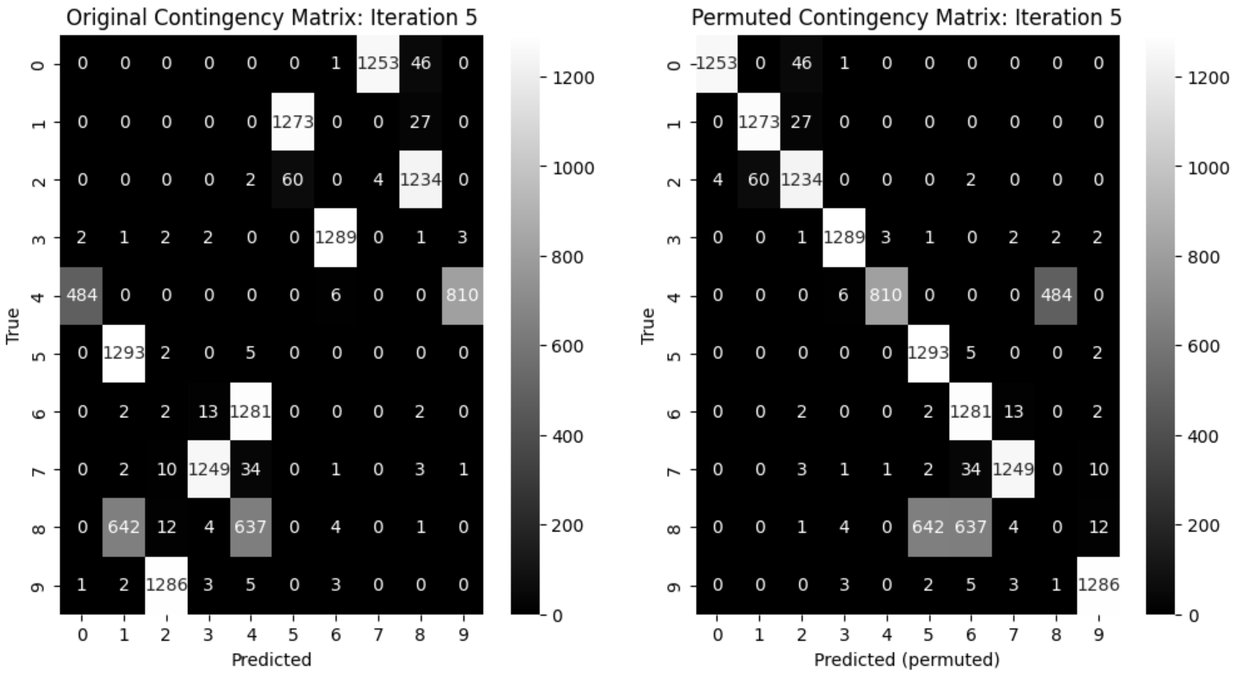
Original Contingency Matrix: Iteration 3

Permuted Contingency Matrix: Iteration 3

```
Iteration 4 with random: Accuracy = 0.8432, Adjusted Rand Index =
0.8311
```



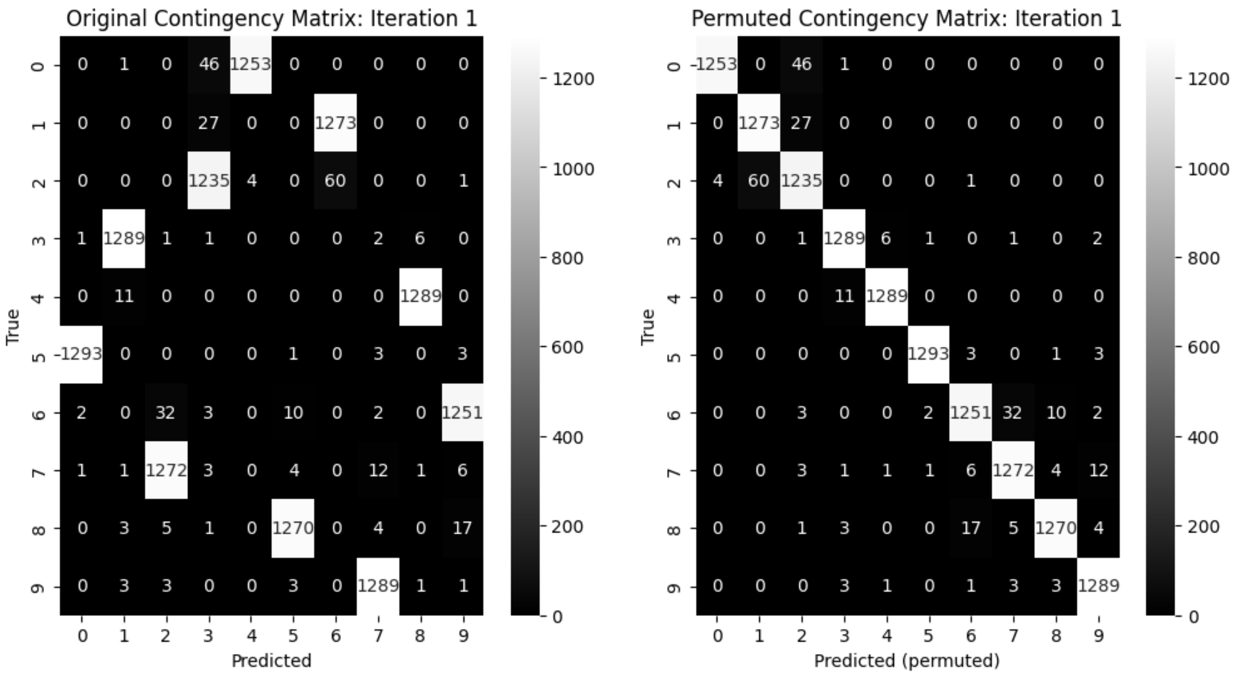
```
Iteration 5 with random: Accuracy = 0.8437, Adjusted Rand Index =
0.8013
```



# e) Using kmeans++ initializer

```
plusplus_km_five = Kmeans(X,y,'k-means++',5)
```

```
Iteration 1 with k-means++: Accuracy = 0.9780, Adjusted Rand Index =
0.9523
```



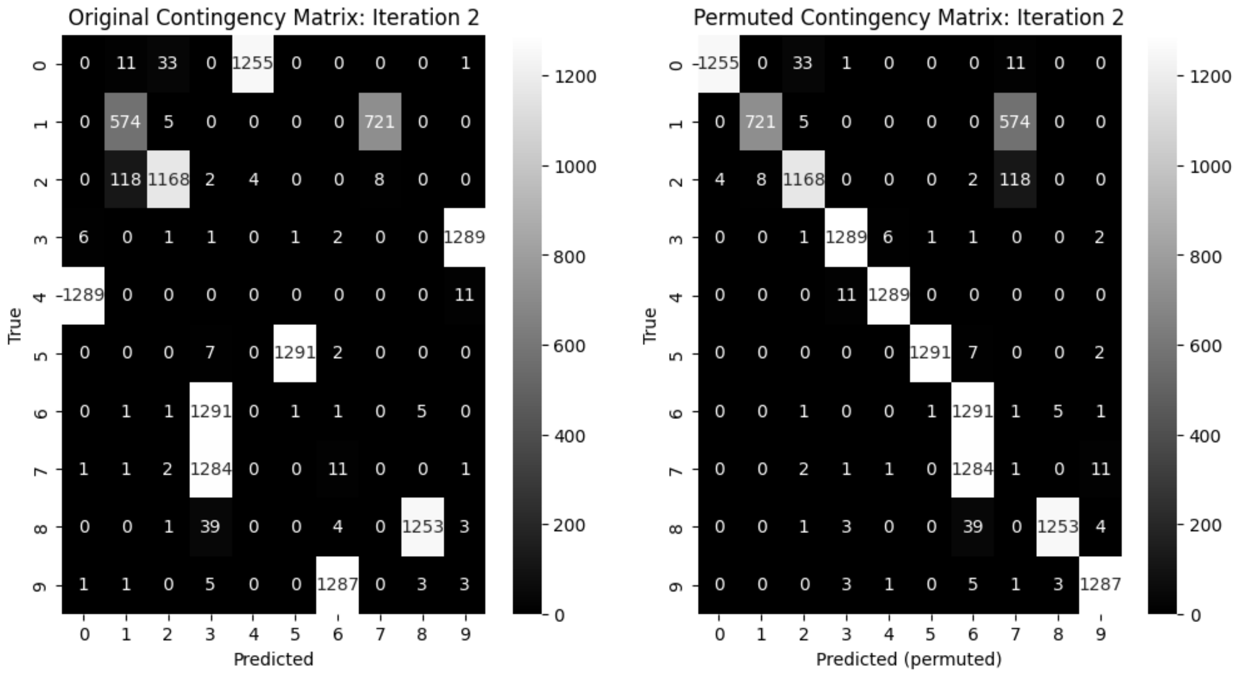
Original Contingency Matrix: Iteration 1

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 46 | 1253 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 27 | 0 | 0 | 1273 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 1235 | 4 | 0 | 60 | 0 | 0 | 1 |
| 3 | 1 | 1289 | 1 | 1 | 0 | 0 | 0 | 2 | 6 | 0 |
| 4 | 0 | 11 | 0 | 0 | 0 | 0 | 0 | 0 | 1289 | 0 |
| 5 | 1293 | 0 | 0 | 0 | 0 | 1 | 0 | 3 | 0 | 3 |
| 6 | 2 | 0 | 32 | 3 | 0 | 10 | 0 | 2 | 0 | 1251 |
| 7 | 1 | 1 | 1272 | 3 | 0 | 4 | 0 | 12 | 1 | 6 |
| 8 | 0 | 3 | 5 | 1 | 0 | 1270 | 0 | 4 | 0 | 17 |
| 9 | 0 | 3 | 3 | 0 | 0 | 3 | 0 | 1289 | 1 | 1 |

Permuted Contingency Matrix: Iteration 1

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1253 | 0 | 46 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1273 | 27 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 4 | 60 | 1235 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 3 | 0 | 0 | 1 | 1289 | 6 | 1 | 0 | 1 | 0 | 2 |
| 4 | 0 | 0 | 0 | 11 | 1289 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 1293 | 3 | 0 | 1 | 3 |
| 6 | 0 | 0 | 3 | 0 | 0 | 2 | 1251 | 32 | 10 | 2 |
| 7 | 0 | 0 | 3 | 1 | 1 | 1 | 6 | 1272 | 4 | 12 |
| 8 | 0 | 0 | 1 | 3 | 0 | 0 | 17 | 5 | 1270 | 4 |
| 9 | 0 | 0 | 0 | 3 | 1 | 0 | 1 | 3 | 3 | 1289 |

```
Iteration 2 with k-means++: Accuracy = 0.8342, Adjusted Rand Index =
0.8275
```



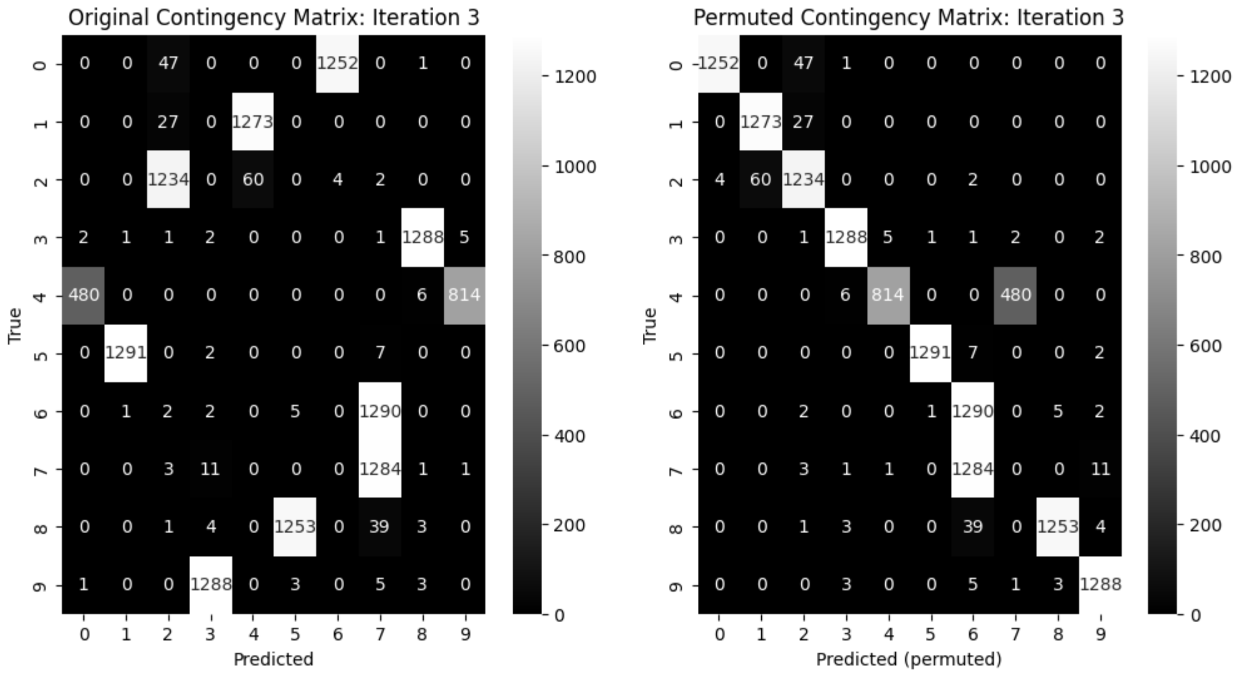
Original Contingency Matrix: Iteration 2

| True \ Predicted | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 11 | 33 | 0 | 1255 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 574 | 5 | 0 | 0 | 0 | 0 | 721 | 0 | 0 |
| 2 | 0 | 118 | 1168 | 2 | 4 | 0 | 0 | 8 | 0 | 0 |
| 3 | 6 | 0 | 1 | 1 | 0 | 1 | 2 | 0 | 0 | 1289 |
| 4 | 1289 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 11 |
| 5 | 0 | 0 | 0 | 7 | 0 | 1291 | 2 | 0 | 0 | 0 |
| 6 | 0 | 1 | 1 | 1291 | 0 | 1 | 1 | 0 | 5 | 0 |
| 7 | 1 | 1 | 2 | 1284 | 0 | 0 | 11 | 0 | 0 | 1 |
| 8 | 0 | 0 | 1 | 39 | 0 | 0 | 4 | 0 | 1253 | 3 |
| 9 | 1 | 1 | 0 | 5 | 0 | 0 | 1287 | 0 | 3 | 3 |

Permuted Contingency Matrix: Iteration 2

| True \ Predicted (permuted) | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1255 | 0 | 33 | 1 | 0 | 0 | 0 | 11 | 0 | 0 |
| 1 | 0 | 721 | 5 | 0 | 0 | 0 | 0 | 574 | 0 | 0 |
| 2 | 4 | 8 | 1168 | 0 | 0 | 0 | 2 | 118 | 0 | 0 |
| 3 | 0 | 0 | 1 | 1289 | 6 | 1 | 1 | 0 | 0 | 2 |
| 4 | 0 | 0 | 0 | 11 | 1289 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 1291 | 7 | 0 | 0 | 2 |
| 6 | 0 | 0 | 1 | 0 | 0 | 1 | 1291 | 1 | 5 | 1 |
| 7 | 0 | 0 | 2 | 1 | 1 | 0 | 1284 | 1 | 0 | 11 |
| 8 | 0 | 0 | 1 | 3 | 0 | 0 | 39 | 0 | 1253 | 4 |
| 9 | 0 | 0 | 0 | 3 | 1 | 0 | 5 | 1 | 3 | 1287 |

```
Iteration 3 with k-means++: Accuracy = 0.8448, Adjusted Rand Index =
0.8307
```



Original Contingency Matrix: Iteration 3

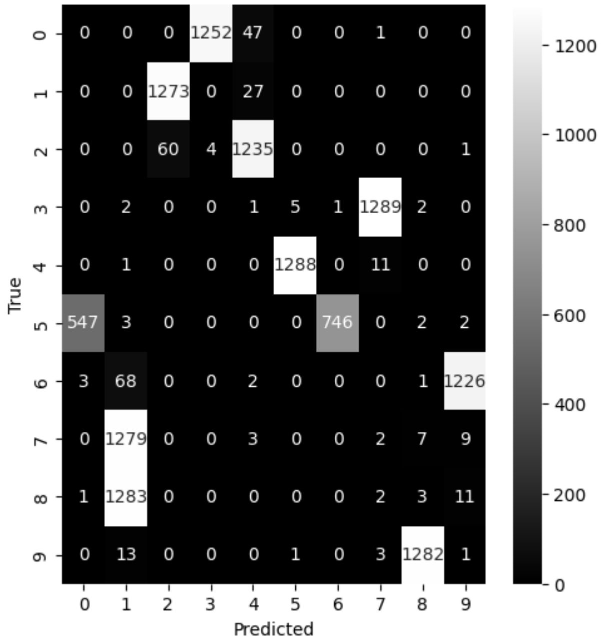
| True \ Predicted | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 47 | 0 | 0 | 0 | 1252 | 0 | 1 | 0 |
| 1 | 0 | 0 | 27 | 0 | 1273 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 1234 | 0 | 60 | 0 | 4 | 2 | 0 | 0 |
| 3 | 2 | 1 | 1 | 2 | 0 | 0 | 0 | 1 | 1288 | 5 |
| 4 | 480 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 814 |
| 5 | 0 | 1291 | 0 | 2 | 0 | 0 | 0 | 7 | 0 | 0 |
| 6 | 0 | 1 | 2 | 2 | 0 | 5 | 0 | 1290 | 0 | 0 |
| 7 | 0 | 0 | 3 | 11 | 0 | 0 | 0 | 1284 | 1 | 1 |
| 8 | 0 | 0 | 1 | 4 | 0 | 1253 | 0 | 39 | 3 | 0 |
| 9 | 1 | 0 | 0 | 1288 | 0 | 3 | 0 | 5 | 3 | 0 |

Permuted Contingency Matrix: Iteration 3

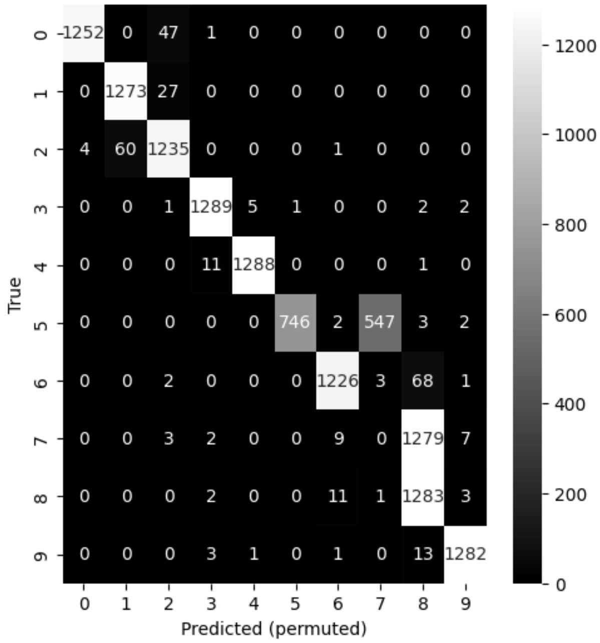
| True \ Predicted (permuted) | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1252 | 0 | 47 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1273 | 27 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 4 | 60 | 1234 | 0 | 0 | 0 | 2 | 0 | 0 | 0 |
| 3 | 0 | 0 | 1 | 1288 | 5 | 1 | 1 | 2 | 0 | 2 |
| 4 | 0 | 0 | 0 | 6 | 814 | 0 | 0 | 480 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 1291 | 7 | 0 | 0 | 2 |
| 6 | 0 | 0 | 2 | 0 | 0 | 1 | 1290 | 0 | 5 | 2 |
| 7 | 0 | 0 | 3 | 1 | 1 | 0 | 1284 | 0 | 0 | 11 |
| 8 | 0 | 0 | 1 | 3 | 0 | 0 | 39 | 0 | 1253 | 4 |
| 9 | 0 | 0 | 0 | 3 | 0 | 0 | 5 | 1 | 3 | 1288 |

```
Iteration 4 with k-means++: Accuracy = 0.8365, Adjusted Rand Index =
0.8208
```

Original Contingency Matrix: Iteration 4

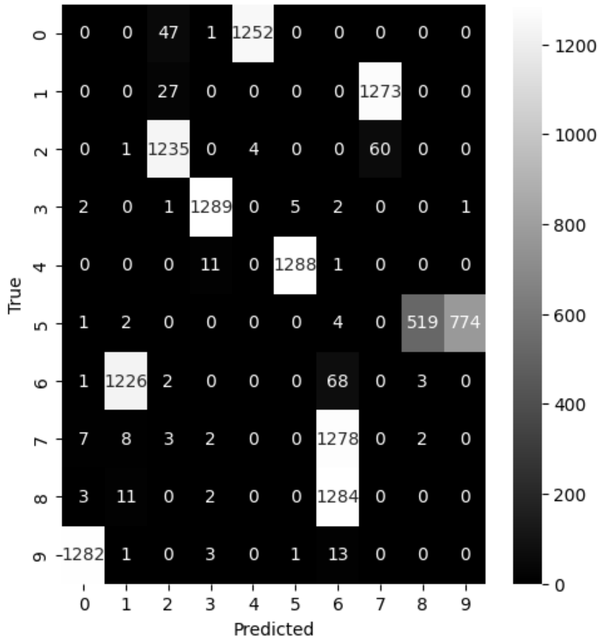


Permuted Contingency Matrix: Iteration 4

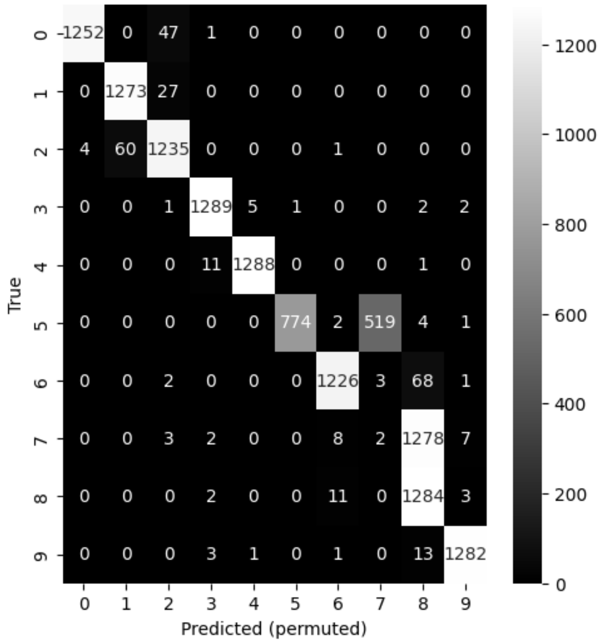


```
Iteration 5 with k-means++: Accuracy = 0.8388, Adjusted Rand Index =
0.8212
```

Original Contingency Matrix: Iteration 5



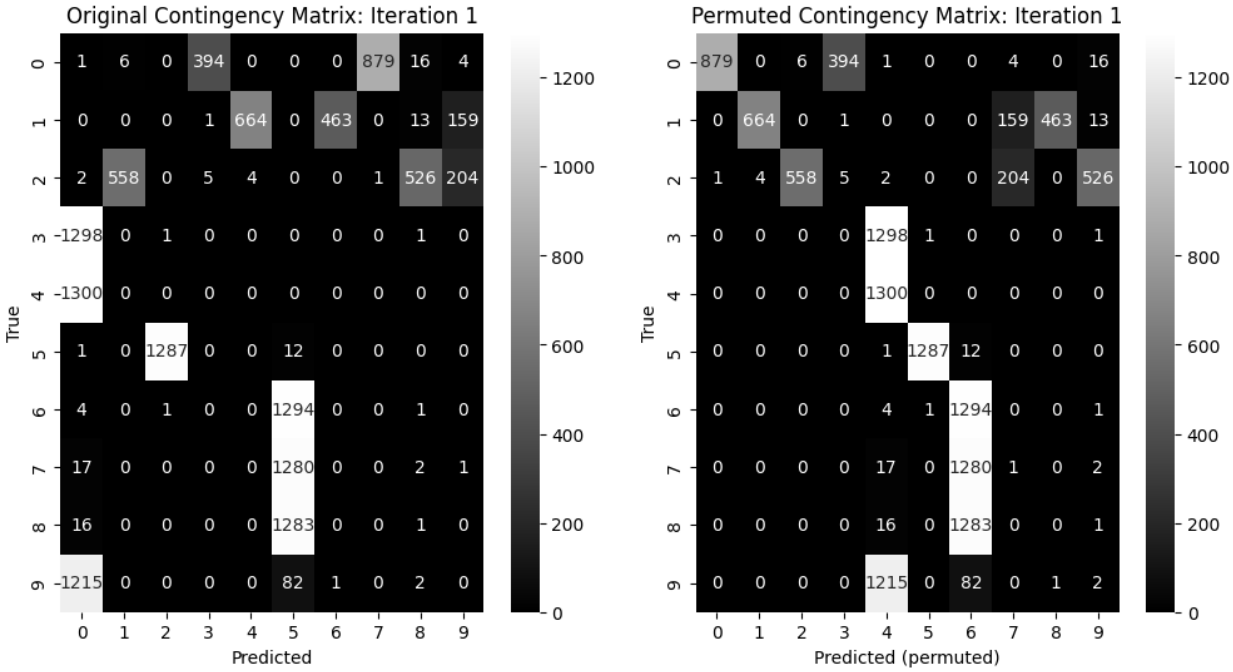
Permuted Contingency Matrix: Iteration 5



# f) Using furthest point initializer

```
def furthest_point(X, k):

    n_samples, n_features = X.shape
    centers = np.empty((k, n_features))
    center_idx = np.random.choice(n_samples)
    centers[0] = X[center_idx]

    for i in range(1, k):
        distances = cdist(X, centers[:i], 'euclidean')
        min_distances = np.min(distances, axis=1)
        centers[i] = X[np.argmax(min_distances)]

    return centers

furthest_point_five = Kmeans(X, y, furthest_point(X, 10), 5)
```

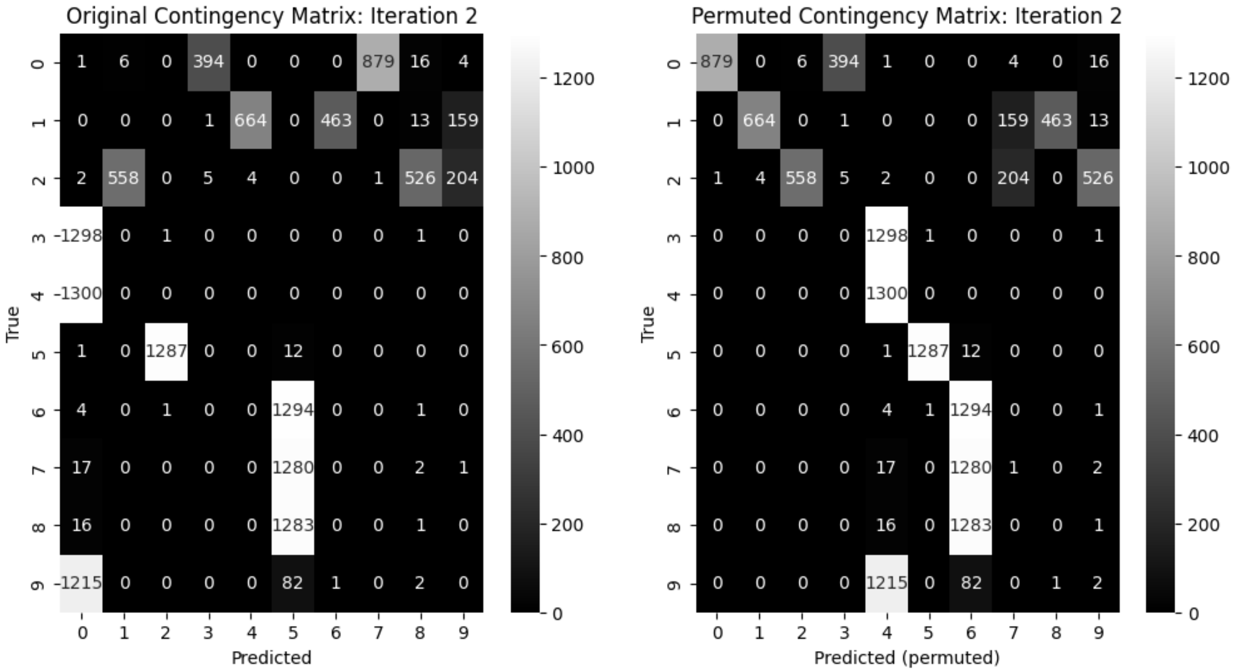
```
Iteration 1 with [[ 1.08708119  0.10274044  1.3284713  ...  0.20177411
-0.73461658
  -2.06957603]
 [-2.08751225 -0.28992596 -1.90656745 ... -2.14951348  0.5614475
  -0.5089283 ]
 [-0.91873735 -0.30468169  0.30459017 ...  0.03175154  0.53389227
   1.11217272]
 ...
 [ 0.58948386  1.6152029  -0.07295995 ... -0.80592066  0.10592526
   0.17286082]
 [ 0.67048806  0.45769775 -1.88142967 ... -0.19062555  0.97133142
  -1.16675484]
 [-0.99684846  0.08060685 -0.77995998 ... -2.38692307 -3.12664652
  -4.00407934]]: Accuracy = 0.4604, Adjusted Rand Index = 0.4611
```



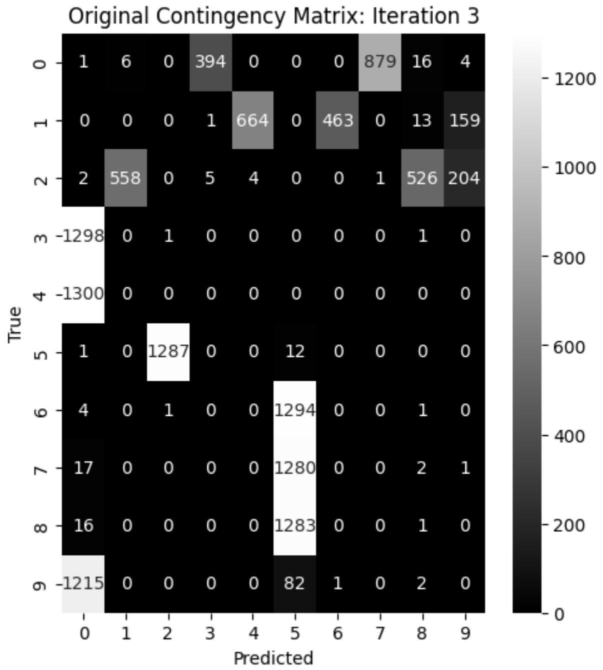
Original Contingency Matrix: Iteration 1

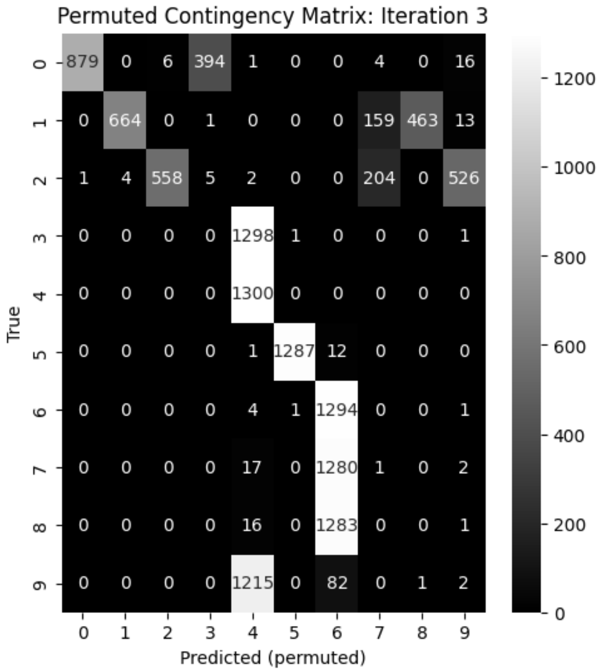
Permuted Contingency Matrix: Iteration 1

```
Iteration 2 with [[ 1.08708119  0.10274044  1.3284713  ...  0.20177411
-0.73461658
  -2.06957603]
 [-2.08751225 -0.28992596 -1.90656745 ... -2.14951348  0.5614475
  -0.5089283 ]
 [-0.91873735 -0.30468169  0.30459017 ...  0.03175154  0.53389227
   1.11217272]
 ...
 [ 0.58948386  1.6152029  -0.07295995 ... -0.80592066  0.10592526
   0.17286082]
 [ 0.67048806  0.45769775 -1.88142967 ... -0.19062555  0.97133142
  -1.16675484]
 [-0.99684846  0.08060685 -0.77995998 ... -2.38692307 -3.12664652
  -4.00407934]]: Accuracy = 0.4604, Adjusted Rand Index = 0.4611
```

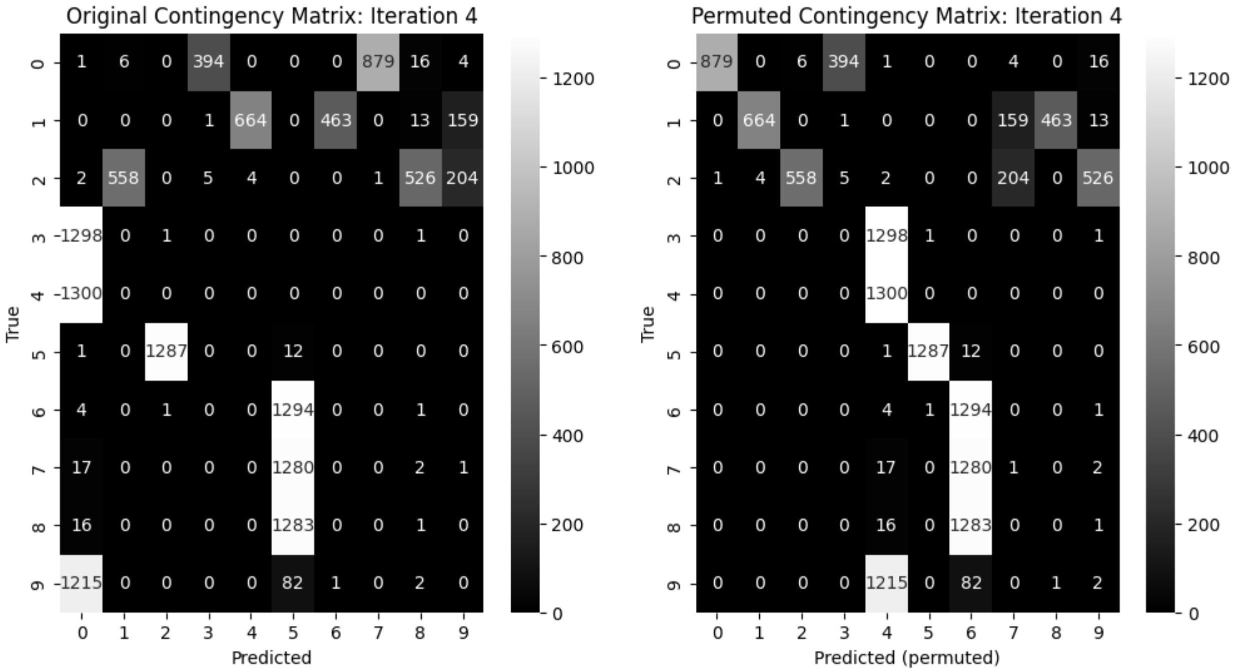


```
Iteration 3 with [[ 1.08708119  0.10274044  1.3284713  ...  0.20177411
-0.73461658
  -2.06957603]
 [-2.08751225 -0.28992596 -1.90656745 ... -2.14951348  0.5614475
  -0.5089283 ]
 [-0.91873735 -0.30468169  0.30459017 ...  0.03175154  0.53389227
   1.11217272]
 ...
 [ 0.58948386  1.6152029  -0.07295995 ... -0.80592066  0.10592526
   0.17286082]
 [ 0.67048806  0.45769775 -1.88142967 ... -0.19062555  0.97133142
  -1.16675484]
 [-0.99684846  0.08060685 -0.77995998 ... -2.38692307 -3.12664652
  -4.00407934]]: Accuracy = 0.4604, Adjusted Rand Index = 0.4611
```

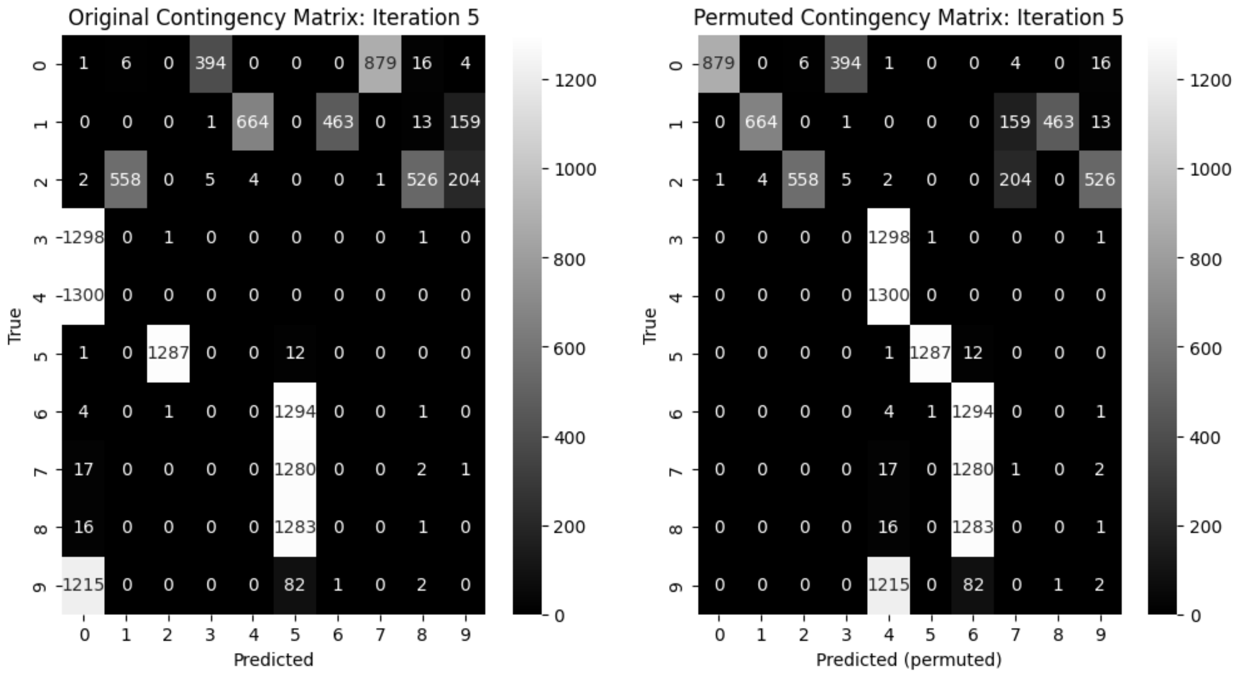




```
Iteration 4 with [[ 1.08708119  0.10274044  1.3284713  ...  0.20177411
-0.73461658
  -2.06957603]
 [-2.08751225 -0.28992596 -1.90656745 ... -2.14951348  0.5614475
  -0.5089283 ]
 [-0.91873735 -0.30468169  0.30459017 ...  0.03175154  0.53389227
   1.11217272]
 ...
 [ 0.58948386  1.6152029  -0.07295995 ... -0.80592066  0.10592526
   0.17286082]
 [ 0.67048806  0.45769775 -1.88142967 ... -0.19062555  0.97133142
  -1.16675484]
 [-0.99684846  0.08060685 -0.77995998 ... -2.38692307 -3.12664652
  -4.00407934]]: Accuracy = 0.4604, Adjusted Rand Index = 0.4611
```



```
Iteration 5 with [[ 1.08708119  0.10274044  1.3284713  ...  0.20177411
-0.73461658
  -2.06957603]
 [-2.08751225 -0.28992596 -1.90656745 ... -2.14951348  0.5614475
  -0.5089283 ]
 [-0.91873735 -0.30468169  0.30459017 ...  0.03175154  0.53389227
   1.11217272]
 ...
 [ 0.58948386  1.6152029  -0.07295995 ... -0.80592066  0.10592526
   0.17286082]
 [ 0.67048806  0.45769775 -1.88142967 ... -0.19062555  0.97133142
  -1.16675484]
 [-0.99684846  0.08060685 -0.77995998 ... -2.38692307 -3.12664652
  -4.00407934]]: Accuracy = 0.4604, Adjusted Rand Index = 0.4611
```



# g) Table of average results

```
Table = pd.DataFrame({
    'Init Method': ['Random', 'K-means++', 'Furthest Point'],
    'Avg Accuracy': [random_km_five[0], plusplus_km_five[0],
furthest_point_results[0]],
    'Avg Adjusted Rand Index': [random_km_five[1],
plusplus_km_five[1], furthest_point_results[1]]
})

print(Table)
```

```
      Init Method  Avg Accuracy  Avg Adjusted Rand Index
0          Random      0.773923                 0.748387
1       K-means++      0.866477                 0.850513
2  Furthest Point      0.461077                 0.433430
```