STAT 576 Homework 2

In [1]:

```
# Import all necessary libraries.

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder
from sklearn.manifold import MDS, Isomap, LocallyLinearEmbedding, TSNE
from adjustText import adjust_text
```

In [2]:

```
# Load the data and make the target variable. I used the head() functions to get glimp
# variables and data looked like prior to algorithm fitting/transformations. We use sa

mnist_data = pd.read_csv("C://Users/coryg/Onedrive/Desktop/STAT_576_F24/mnist_hw.csv")
mnist_data.head()
mnist_data = mnist_data
mnist_data.head()

class_target = mnist_data.iloc[:, -1]
mnist_mat = mnist_data.iloc[:, 0:]
mnist_mat.head()
class_target.head()
```

Out[2]:

```
0    0
1    1
2    2
3    3
4    4
Name: target, dtype: int64
```

In [3]:

```
# Problem 1: Feature Extraction by MDS

mds_mnist_res = MDS(n_components=2, random_state=42).fit_transform(mnist_mat)
X_sample1 = np.random.choice(len(mds_mnist_res), size = 200, replace = False)
mds_sample = mds_mnist_res[X_sample1]
y_sample1 = class_target.iloc[X_sample1]
```

```
c:\Users\coryg\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\mani
fold\_mds.py:298: FutureWarning: The default value of `normalized_stress` will change
to `'auto'` in version 1.4. To suppress this warning, manually set the value of `norm
alized_stress`.
  warnings.warn(
```

In [4]:

```
# Problem 2: Feature Extraction by IsoMAP

isomap_mnist_res = Isomap(n_neighbors=10, n_components=2).fit_transform(mnist_mat)
X_sample2 = np.random.choice(len(isomap_mnist_res), size = 200, replace = False)
isomap_sample = isomap_mnist_res[X_sample2]
y_sample2 = class_target.iloc[X_sample2]
```

In [5]:

```
# Problem 3: Feature Extraction by LLE

lle_mnist_res = LocallyLinearEmbedding(n_neighbors=10, n_components=2).fit_transform(m
X_sample3 = np.random.choice(len(lle_mnist_res), size = 200, replace = False)
lle_sample = lle_mnist_res[X_sample3]
y_sample3 = class_target.iloc[X_sample3]
```

In [6]:

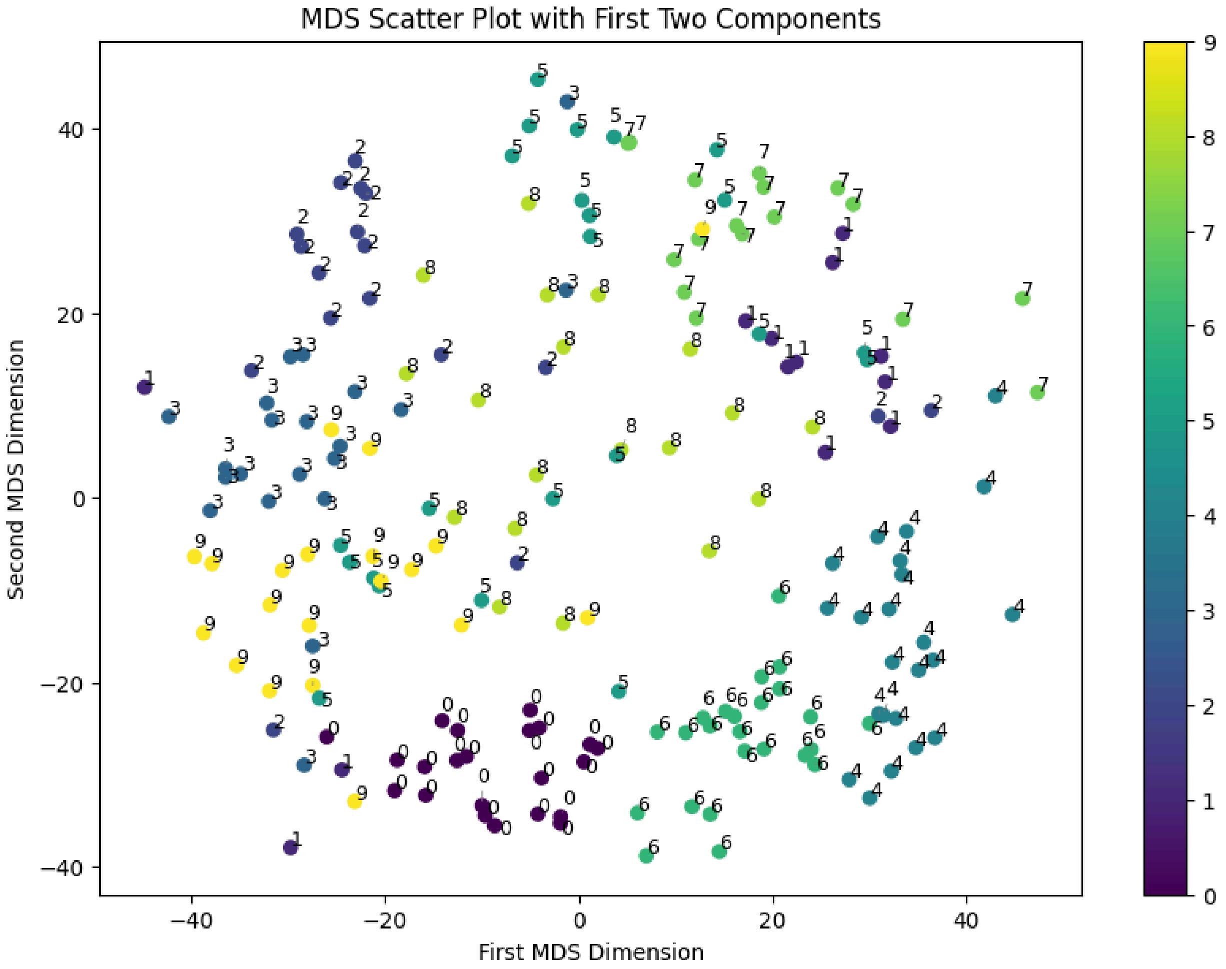
```
# Problem 4: Feature Extraction bt t-SNE

tsne_mnist_res = TSNE(n_components=2, perplexity=5, random_state=0).fit_transform(mnis
X_sample4 = np.random.choice(len(tsne_mnist_res), size = 200, replace = False)
tsne_sample = tsne_mnist_res[X_sample4]
y_sample4 = class_target.iloc[X_sample4]
```

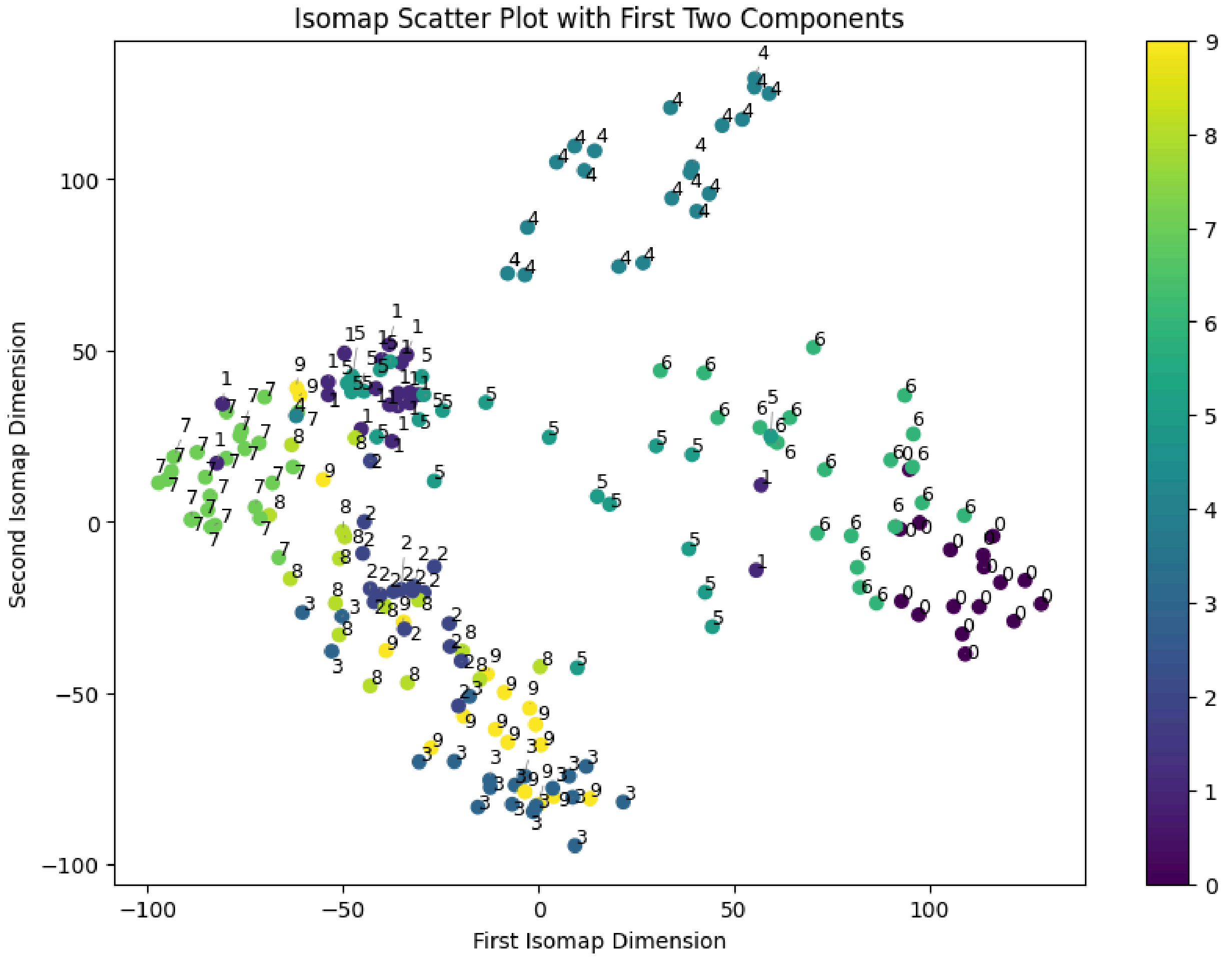
In [9]:

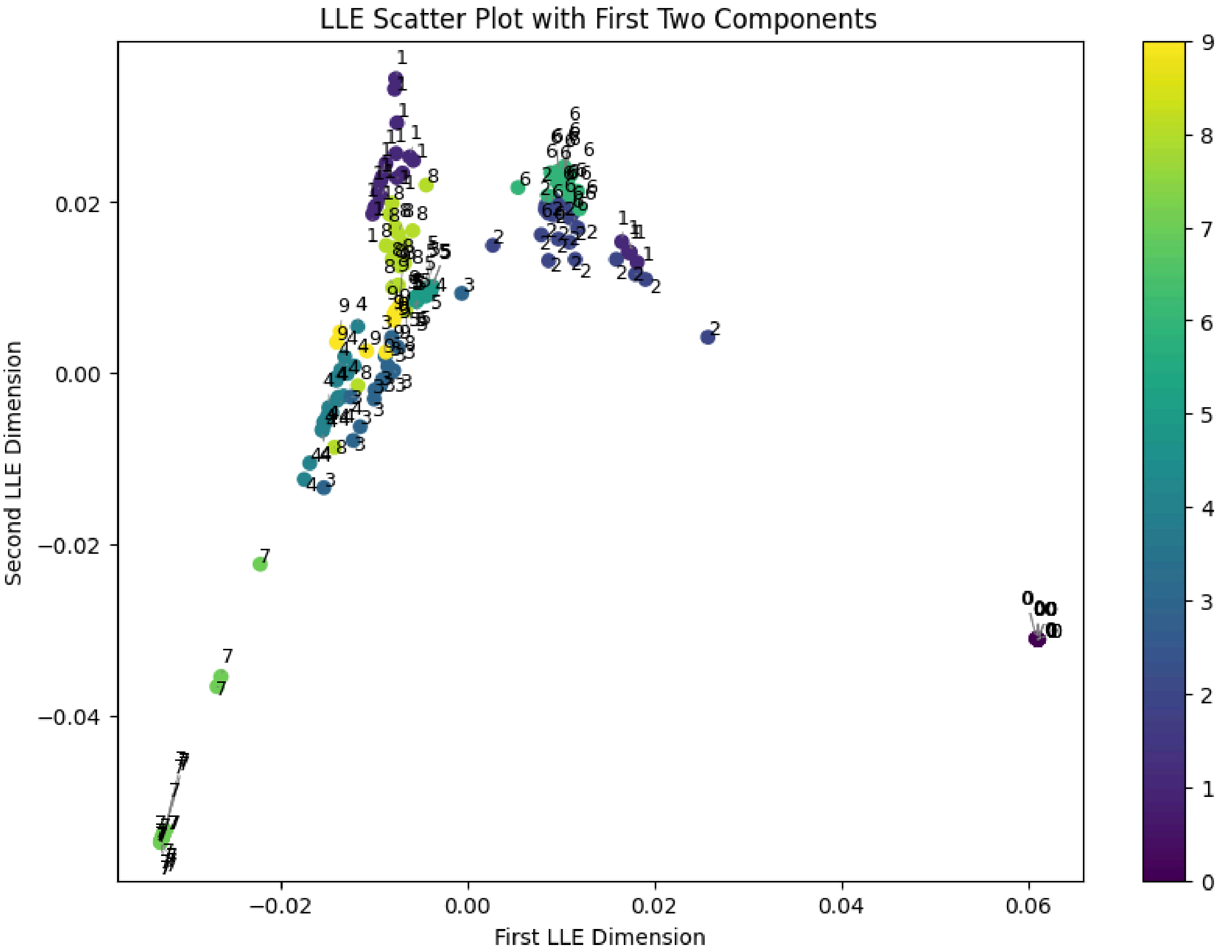
```
def plot_scatter(data, sample_target, method_name):
    label_encoder = LabelEncoder()
    target_encoded = label_encoder.fit_transform(sample_target)
    plt.figure(figsize=(10, 7))
    scatter = plt.scatter(data[:, 0], data[:, 1], c=target_encoded, cmap='viridis')
    plt.title(f'{method_name} Scatter Plot with First Two Components')
    plt.xlabel(f'First {method_name} Dimension')
    plt.ylabel(f'Second {method_name} Dimension')

    text_list = [plt.text(data[i, 0], data[i, 1], str(sample_target.iloc[i]), fontsize
    adjust_text(text_list, arrowprops=dict(arrowstyle='-', color='gray', lw=0.5))
    plt.colorbar(scatter)
    plt.show()
```

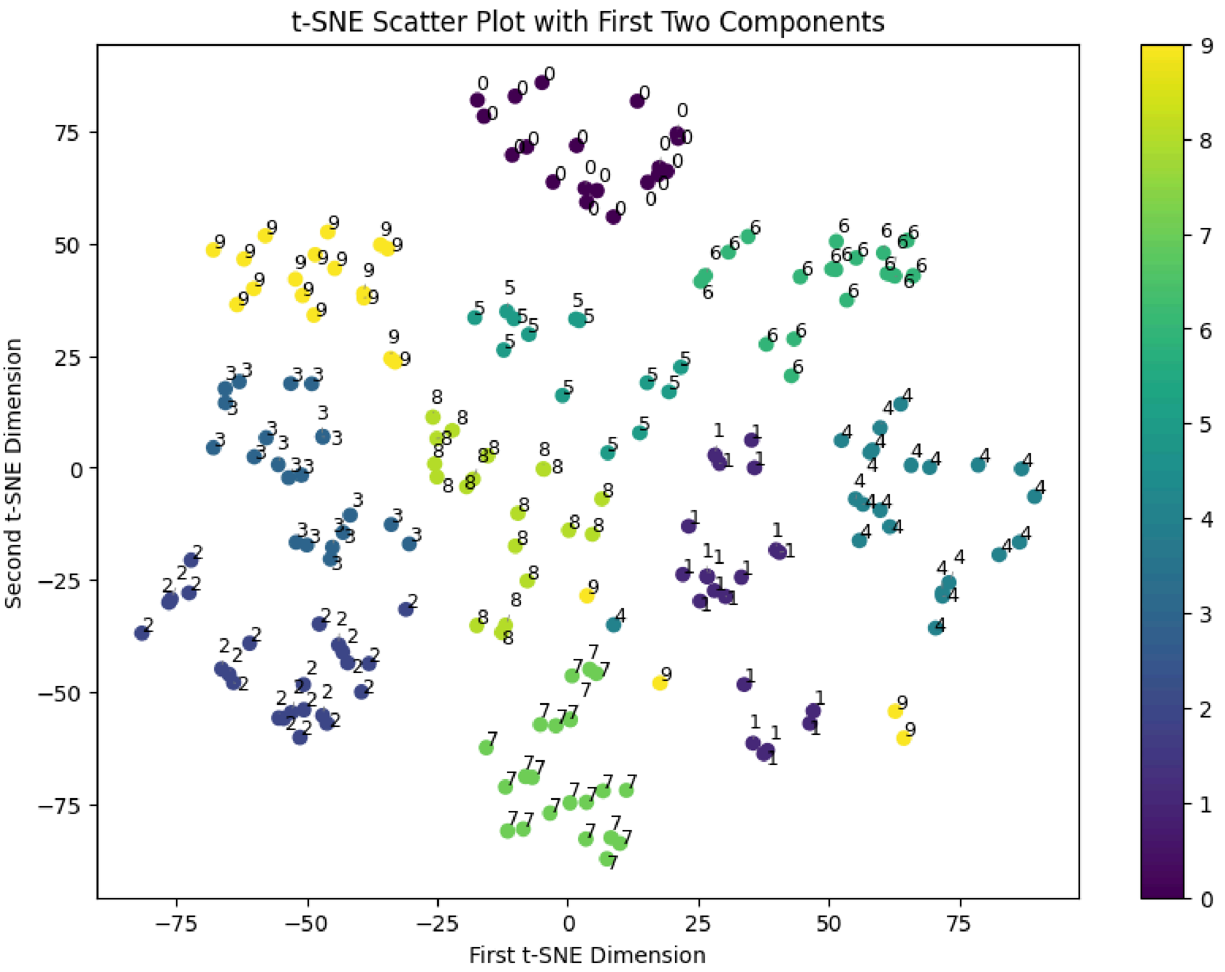
In [10]:

```
plot_scatter(mds_sample, y_sample1, 'MDS')
plot_scatter(isomap_sample, y_sample2, 'Isomap')
plot_scatter(lle_sample, y_sample3, 'LLE')
plot_scatter(tsne_sample, y_sample4, 't-SNE')
```




Isomap Scatter Plot with First Two Components
First Isomap Dimension
Second Isomap Dimension
−100
−50
0
50
100
−100
−50
0
50
100
0
1
2
3
4
5
6
7
8
9


LLE Scatter Plot with First Two Components
First LLE Dimension
Second LLE Dimension
−0.02
0.00
0.02
0.04
0.06
0.02
0.00
−0.02
−0.04
0
1
2
3
4
5
6
7
8
9



In [ ]:

```
# Survey Extra Credit

# 1. How long does it take to finish this homework?

# It took me about one day to finish this homework because I made sure to better brush
# compiling and processing the output of my code. In the first homework, I struggled a
# that I took a hiatus on using Python to learn R and SQL.

# 2. What problem was the easiest to implement?

# The problem that was easiest to implement was the MDS problem since the algorithm pa
# as specifying the number of components and the random state.

# 3. What problem was the most difficult to implement and why?

# The problem that was most difficult to implement was drawing the scatterplot for eac
# my memory error on my machine occurred because I did not take a random sample after
# implementation.

# 4. Do you have the confidence to implement all of these algorithms to your real prob

# Yes, I have the confidence to implement these algorithms to both my real problems wo
# extraction and for future projects.
```