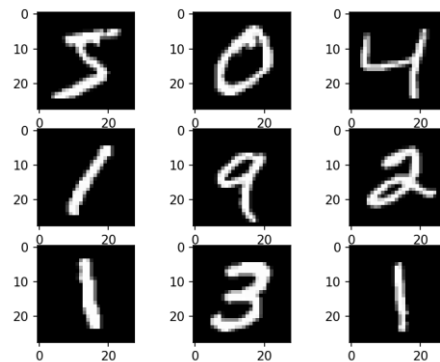Valerie Melland
Statistical Learning
Final Homework PCA
4/24/2023

For this project I used the MNIST Dataset with 60,000 images of handwritten digits and 10,000 testing images. I imported the dataset and printed out the first nine images to check if the import worked correctly.
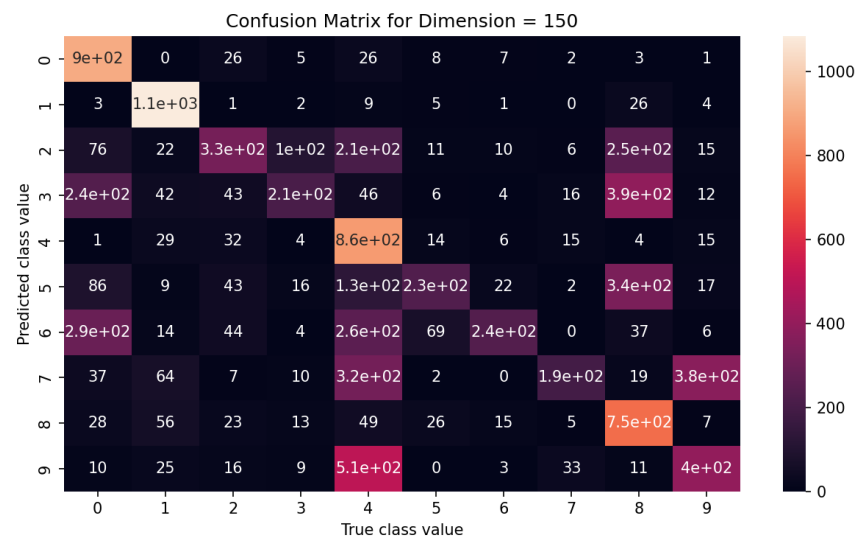

I developed and used a Naïve Bayes classifier for matching the images to their labels. First, I had to reshape the training and testing data to make it easier to classify. After this, I trained the model on the MNIST Dataset with the complete dimensions, 28*28, or 784. Below is a heatmap of the confusion
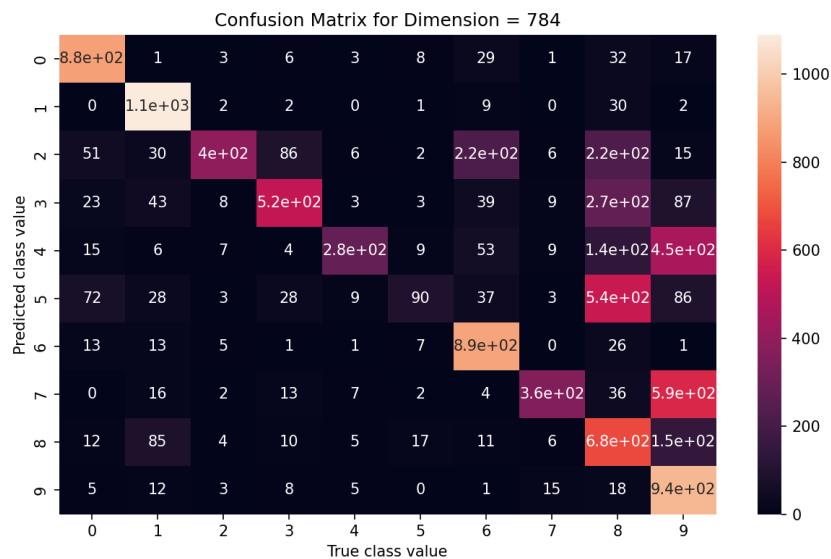
matrix for dimension 784. This shows the number of images classified as a number from 1 through 9. The diagonal line in the middle are the correctly classified images. The accuracy rating of the first classification at dimension 784 was 61.46% for the training set and 61.29% for the testing set. For this classification for images of dimension 784, it seems like the classifier had a harder time classifying the numbers 6, 8, and 9. Next, I used PCA to reduce the dimension of the input from 784 to 150 and then

compared the accuracy of the two classifiers. Below is another heatmap of the confusion matrix after reducing the dimension of the input to 150. The accuracy rating of this classification was 52.2% for the training accuracy and 51.93% for the testing accuracy. I thought that reducing the dimension would help

Confusion Matrix for Dimension = 150

	0	1	2	3	4	5	6	7	8	9
0	9e+02	0	26	5	26	8	7	2	3	1
1	3	1.1e+03	1	2	9	5	1	0	26	4
2	76	22	3.3e+02	1e+02	2.1e+02	11	10	6	2.5e+02	15
3	2.4e+02	42	43	2.1e+02	46	6	4	16	3.9e+02	12
4	1	29	32	4	8.6e+02	14	6	15	4	15
5	86	9	43	16	1.3e+02	2.3e+02	22	2	3.4e+02	17
6	2.9e+02	14	44	4	2.6e+02	69	2.4e+02	0	37	6
7	37	64	7	10	3.2e+02	2	0	1.9e+02	19	3.8e+02
8	28	56	23	13	49	26	15	5	7.5e+02	7
9	10	25	16	9	5.1e+02	0	3	33	11	4e+02

(y-axis: Predicted class value; x-axis: True class value)

the accuracy rating of the classifier, but it didn't so that is interesting. It looks like the classifier had a hard time classifying the number 4 and the number 8.

I decided to try reducing the dimension to 50 and comparing the accuracy of the model against the test for dimension of 150. The heatmap below shows the confusion matrix for dimension 50. The training

Confusion Matrix for Dimension = 50

	0	1	2	3	4	5	6	7	8	9
0	9.2e+02	0	27	3	3	7	11	0	13	1
1	0	1.1e+03	3	0	1	3	4	1	60	3
2	42	17	5.7e+02	52	57	9	23	6	2.4e+02	15
3	36	18	48	5.8e+02	1	60	2	12	2.4e+02	13
4	3	19	53	2	8.2e+02	6	4	12	11	50
5	28	5	43	27	25	5.8e+02	9	6	1.5e+02	15
6	75	9	55	0	42	27	6.9e+02	0	56	2
7	15	42	17	8	54	1	0	4.2e+02	32	4.4e+02
8	13	33	38	22	12	30	3	8	8e+02	13
9	17	16	32	7	1.2e+02	2	0	18	23	7.7e+02

(y-axis: Predicted class value; x-axis: True class value)
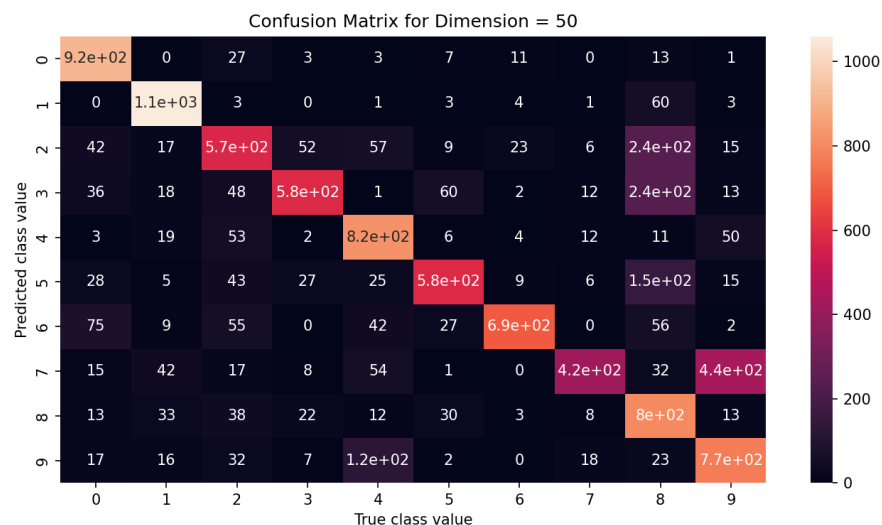
accuracy was 72.26% and the testing accuracy was 72.25%. This classifier seemed to work the best out of the three tries. There still looks like there is some confusion when classifying the number 8 and the number 9, but there are less outliers in the heatmap, so I think that reducing the dimension to 50 helped when classifying the images when compared the reducing the dimension to 150.

This project was very interesting and fun. I think next, I will try comparing the Naïve Bayes classifier to the other types of classifiers for fun. Overall, I think that my classifier worked pretty well.