ECE 539: Homework 3

Luke Yichao Zhang

September 23, 2024

1 Problem 1 - Confusion Matrix:

1.1 How many spam(P) results does this new test report?

$$\begin{aligned} \textit{Number of Predict } P &= TP + FP \\ &= 15 + 4 \\ &= 19 \end{aligned}$$

1.2 What percentage of actual spam emails are correctly identified as spam in this test?

$$\begin{aligned} \text{Percentage} &= \frac{TP}{TP + FN} \\ &= \frac{15}{15 + 5} \\ &= 75\% \end{aligned}$$

1.3 What is the false positive rate, defined as the fraction of products that are reported as defective but are actually non-defective, among all negative tests?

$$\begin{aligned} \text{False Positive Rate} &= \frac{FP}{TN + FP} \\ &= \frac{4}{4 + 16} \\ &= 20\% \end{aligned}$$

2 Problem 2 - Performance Metrics

2.1 (a) If b = 0.3, fill in the predicted label in the 4th row of the above table.

P (y(k) = 1—x(k))	0.05	0.15	0.40	0.55	0.25	0.45	0.48	0.62	0.67	0.75
True Label	0	0	0	0	1	1	1	1	1	1
Predicted Label $y(k)$	0	0	1	1	0	1	1	1	1	1

2.2 (b) Compute the confusion matrix C with b = 0.3.

Assume 0 is negative, and 1 is positive. Matrix C is as followed:

Actual/Predicted	0	1
0	2	2
1	1	5

2.3 (c) With b = 0.3, compute the following quantities: sensitivity (sen), specificity (spe), Pr. False Alarm (pfa), Pr. Miss (pmiss), precision (pre), recall, and accuracy.

$$\begin{aligned} Sensitivity &= Recall \\ &= \frac{TP}{TP+FN} \\ &= \frac{5}{6} \end{aligned}$$

$$\begin{aligned} Specificity &= \frac{TN}{TN+FP} \\ &= \frac{2}{4} = \frac{1}{2} \end{aligned}$$

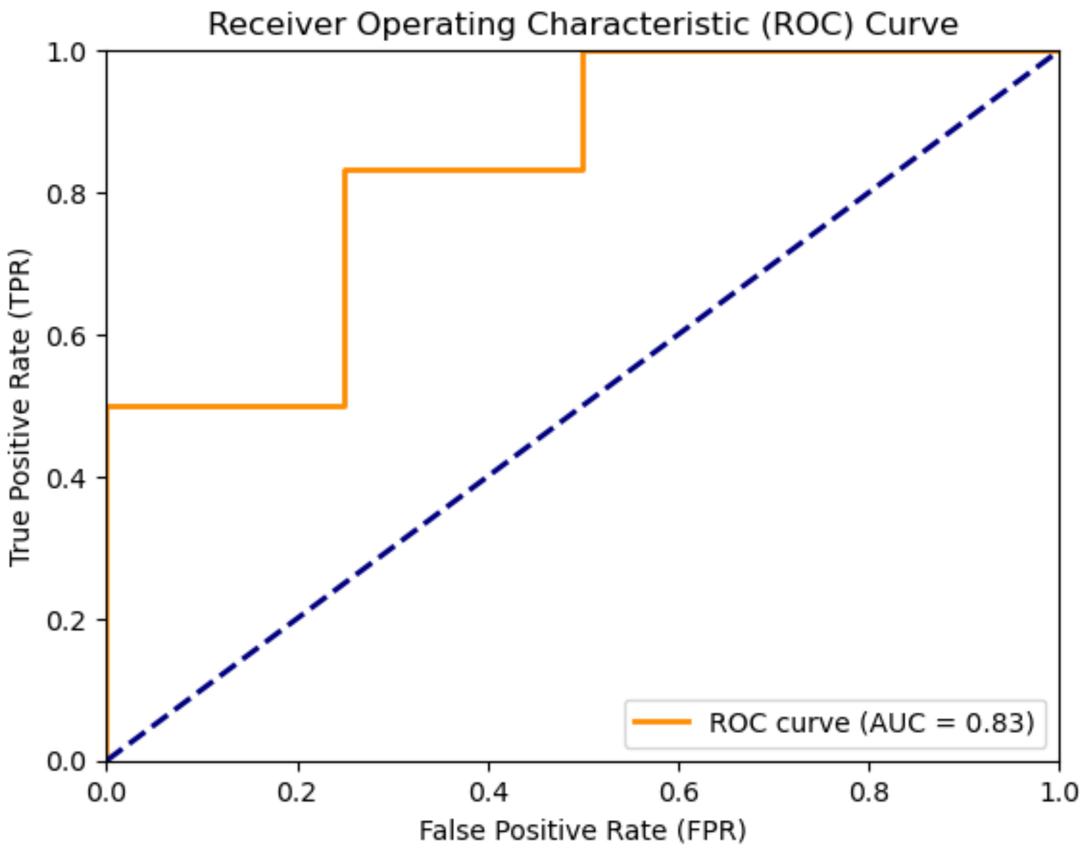
$$\begin{aligned} Pr.FalseAlarm &= 1 - Specificity \\ &= \frac{FP}{TN+FP} \\ &= \frac{2}{4} = \frac{1}{2} \end{aligned}$$

$$\begin{aligned} Pr.Miss &= 1 - Sensitivity \\ &= \frac{FN}{TP+FN} \\ &= \frac{1}{6} \end{aligned}$$

$$\begin{aligned} Precision &= \frac{TP}{TP+FP} \\ &= \frac{5}{7} \end{aligned}$$

$$\begin{aligned} Accuracy &= \frac{TP+TN}{TP+FP+TN+FN} \\ &= \frac{7}{10} \end{aligned}$$

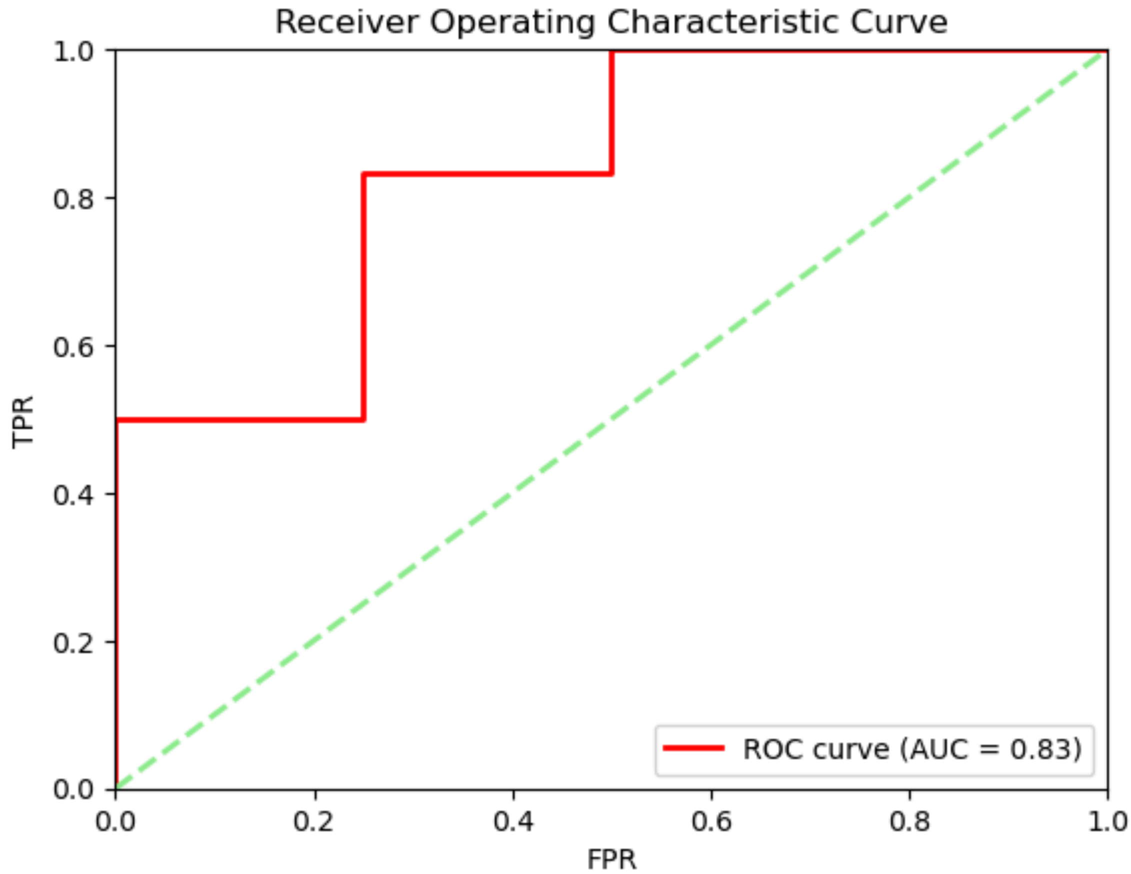
2.4 (d) For the value of threshold b varying from 0 to 1, compute the list of distinct pairs of (TPR, FPR) and then plot the ROC curve and calculate the area under the ROC curve (AUC).



(d) For the value of threshold b varying from 0 to 1, compute the list of distinct pairs of (TPR, FPR) and then plot the ROC curve and calculate the area under the ROC curve (AUC).

In [168…

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import roc_curve, auc

y_data = np.array([0.05, 0.15, 0.40, 0.55, 0.25, 0.45, 0.48, 0.62, 0.67, 0.75])
y_true = np.array([0, 0, 0, 0, 1, 1, 1, 1, 1, 1]) # true labels

# FPR, TPR thresholds
fpr, tpr, thresholds = roc_curve(y_true, y_data)

for b in thresholds:
    y_prob = np.where(y_data <= b, 0, 1)

# calculate AUC
roc_auc = auc(fpr, tpr)

# plot ROC
plt.figure()
plt.plot(fpr, tpr, color='red', lw=2, label='ROC curve (AUC = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color='lightgreen', lw=2, linestyle='--')  # 参考线
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.0])
plt.xlabel('FPR')
plt.ylabel('TPR')
plt.title('Receiver Operating Characteristic Curve')
plt.legend(loc="lower right")
plt.show()

# 输出 distinct pairs of (TPR, FPR)
distinct_pairs = list(set(zip(tpr, fpr)))
print("Distinct pairs of (TPR, FPR):")
for pair in distinct_pairs:
    print(pair)
```




```
Distinct pairs of (TPR, FPR):
(0.5, 0.25)
(0.5, 0.0)
(0.0, 0.0)
(1.0, 1.0)
(0.8333333333333334, 0.25)
(0.8333333333333334, 0.5)
(1.0, 0.5)
(0.16666666666666666, 0.0)
```

3 Problem 3 - PCA

In [94]:
```python
import torchvision
import torch
import matplotlib.pyplot as plt
import numpy as np

data_transform = torchvision.transforms.Compose ([
    torchvision.transforms.ToTensor(),
    lambda image : torch.floor ( image * 255 / 128) . squeeze (dim =0)
])

mnist_test = torchvision.datasets.MNIST(root = './ temp' , train = False , trans
image , label = mnist_test . __getitem__ (1010) # image is the image and label i
```

(a) Number of samples N = ?. Feature dimension = ?

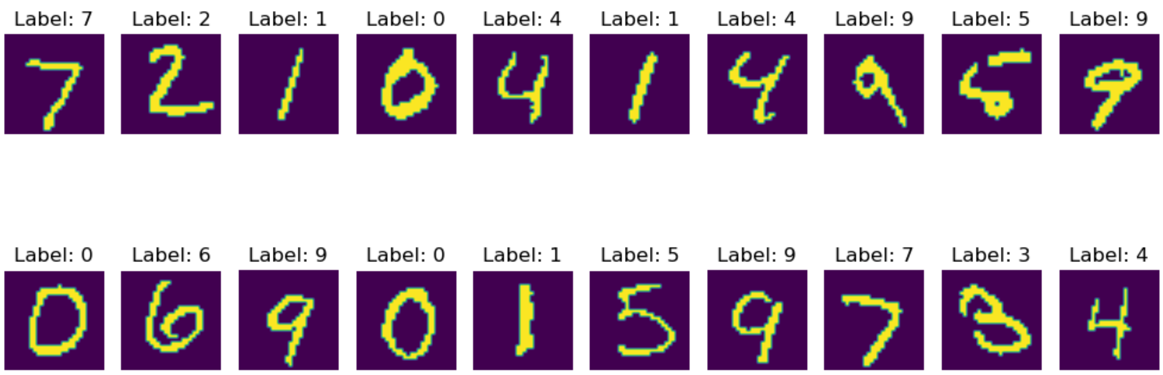
In [98]:
```python
num_of_sample = len(mnist_test)
feature_dim = 28 * 28

print(f'N = {num_of_sample}')
print(f'Feature Dim = {feature_dim}')
```

```
N = 10000
Feature Dim = 784
```

(b) Visualize the first 20 rows (samples). Each should be displayed as a 28 by 28 image. Refer to the d2l 22.9

In [100…

```python
plt.figure(figsize=(10, 5))

for i in range(20):
    image, label = mnist_test.__getitem__(i)

    plt.subplot(2, 10, i + 1)  # 2 by 10
    plt.imshow(image.numpy())  # Transform tensor format into numpy
    plt.title(f'Label: {label}')
    plt.axis('off')

plt.tight_layout()
plt.show()
```

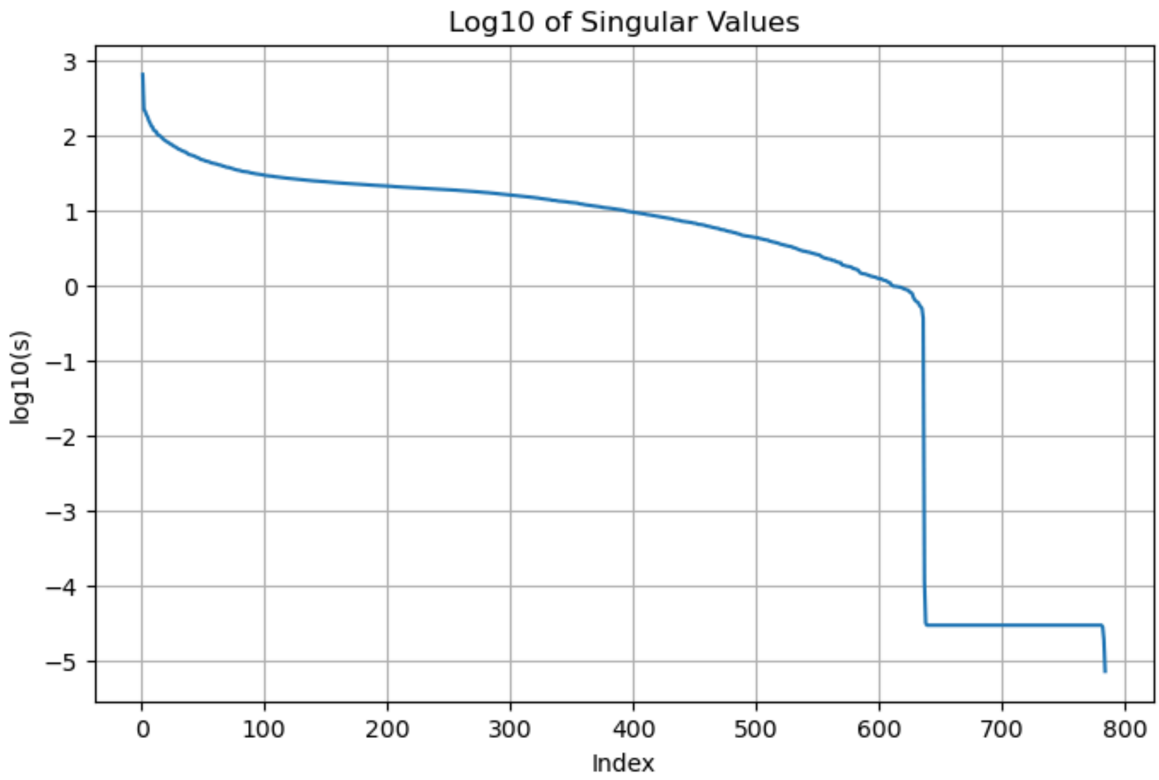


(c) Denote the N × d feature matrix as X. Perform SVD of X. Design the singular values as a vector s. Plot log10(s) over the range 1 to d

In [151…

```python
images = []
labels = []
for i in range(10000):
    image, label = mnist_test.__getitem__(i)
    images.append(image.view(-1))  # flatten image into matrix
    labels.append(label)

X = torch.stack(images)  # N x d feature matrix
print(X.shape)

U, S, V = torch.svd(X)   # SVD

s = S.numpy() # transform svd into array

# plot log10(S)
plt.figure(figsize=(8, 5))
plt.plot(np.arange(1, len(s) + 1), np.log10(s))
plt.title("Log10 of Singular Values")
plt.xlabel("Index")
plt.ylabel("log10(s)")
plt.grid(True)
plt.show()
```

```
torch.Size([10000, 784])
```

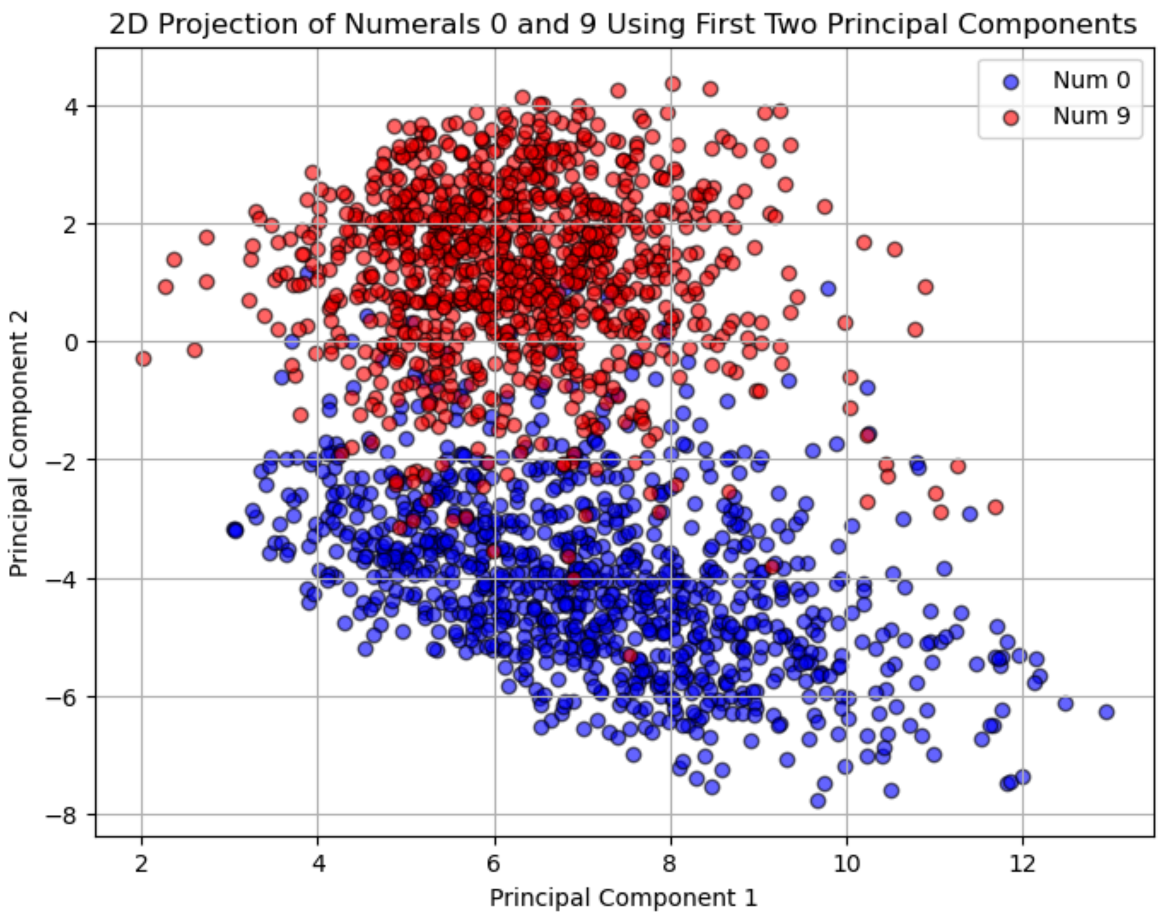


(d) Denote the first two principal components by a d × 2 matrix V . Use the first 2 principal components, projecting each row of the X matrix by computing Z = XV . Each row of Z is a 1 × 2 vector corresponding to a point in a 2D space spanned by the two columns of V . Give a scatter plot of these projected 2D points corresponding to numerals 0 and 9. Note that the numerals are class labels.

In [155…

```
# the first 2 PCA
V_2 = V[:, :2]  # 784 by 2
X_0, X_9 = [], []  # store 0, 9 projection

# find 0, 9
for i in range(num_of_sample):
    image, label = mnist_test.__getitem__(i)
    if label == 0 or label == 9:
        image_flat = image.view(1, -1)  # flatten image to 1 by 784
        Z = image_flat @ V_2  # projection
        if label == 0:
            X_0.append(Z.numpy())
        elif label == 9:
            X_9.append(Z.numpy())

# transform to arry
X_0 = np.array(X_0).reshape(-1, 2)
X_9 = np.array(X_9).reshape(-1, 2)

plt.figure(figsize=(8, 6))
plt.scatter(X_0[:, 0], X_0[:, 1], label='Num 0', alpha=0.6, color='blue', edgecc
plt.scatter(X_9[:, 0], X_9[:, 1], label='Num 9', alpha=0.6, color='red', edgecol
plt.title('2D Projection of Numerals 0 and 9 Using First Two Principal Component
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.legend()
```

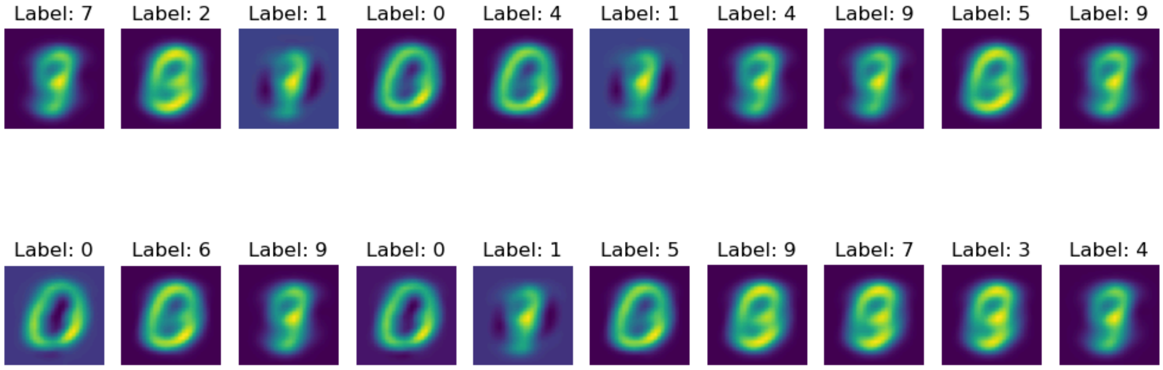
```
plt.grid(True)
plt.show()
```



(e) This one is for all 10 numerals. An approximation of the original feature matrix X may be estimated as: $\hat{X} = XVV^T = ZV^T$ Visualize the corresponding 28 by 28 patterns of the first 20 rows of $\hat{X}$

In [161…

```
Z = X @ V_2
X_hat = Z @ V_2.T

plt.figure(figsize=(10, 5))
for i in range(20):
    image, label = mnist_test.__getitem__(i)
    plt.subplot(2, 10, i + 1)
    image_hat = X_hat[i].view(28, 28).numpy()  # 重塑为28×28的图像
    plt.imshow(image_hat)
    plt.axis('off')
    plt.title(f'Label: {label}')

plt.tight_layout()
plt.show()
```


Label: 7
Label: 2
Label: 1
Label: 0
Label: 4
Label: 1
Label: 4
Label: 9
Label: 5
Label: 9
Label: 0
Label: 6
Label: 9
Label: 0
Label: 1
Label: 5
Label: 9
Label: 7
Label: 3
Label: 4