In [6]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [7]:

```
train_data = pd.read_csv('/content/MNIST_train.csv')
test_data = pd.read_csv('/content/MNIST_test.csv')
train_data.shape
test_data.shape
```

Out[7]:

```
(10000, 787)
```

In [8]:

```
X=train_data.to_numpy()
X_test=test_data.to_numpy()
```

In [9]:

```
y=X[:,2]
y_test=X_test[:,2]
```

In [10]:

```
y
y_test
```

Out[10]:

```
array([7, 2, 1, ..., 4, 5, 6])
```

In [11]:

```
set(y)
set(y_test)
```

Out[11]:

```
{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
```

In [12]:

```
X=X[:,3:]
X_test=X_test[:,3:]
```

In [13]:

```
X.shape
X_test.shape
```

Out[13]:

```
(10000, 784)
```

In [14]:

```
X = X / 255
X_test = X_test/ 255
```

In [15]:

```
from scipy.stats import multivariate_normal as mvn
```

In [16]:

```
class GaussNB():
  def fit(self,X,y,epsilon=1e-1):
    self.likelihoods=dict()
    self.priors=dict()
    self.K=set(y.astype(int))
    print(self.K)

    for k in self.K:
      X_k=X[y==k]#
      print(X_k.shape)
      self.likelihoods[k]={"mean":X_k.mean(axis=0),"cov":X_k.var(axis=0)+epsilon}
      self.priors[k]=len(X_k)/len(X)


  def predict(self,X):
    N, D = X.shape
    P_hat= np.zeros((N,len(self.K)))

    for k, l in self.likelihoods.items():
      P_hat[:,k]=mvn.logpdf(X,l["mean"],l["cov"])+np.log(self.priors[k])

    return P_hat.argmax(axis=1)
```

In [17]:

```
gnb = GaussNB()
```

In [18]:

```
gnb.fit(X, y)
```

```
{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
(5923, 784)
(6742, 784)
(5958, 784)
(6131, 784)
(5842, 784)
(5421, 784)
(5918, 784)
(6265, 784)
(5851, 784)
(5949, 784)
```

In [19]:
```python
y_hat=gnb.predict(X)
```

In [20]:
```python
y_hat_test=gnb.predict(X_test)
```

In [21]:
```python
def accuracy(y,y_hat):
  return np.mean(y==y_hat)
```

In [22]:
```python
accuracy(y,y_hat)
```

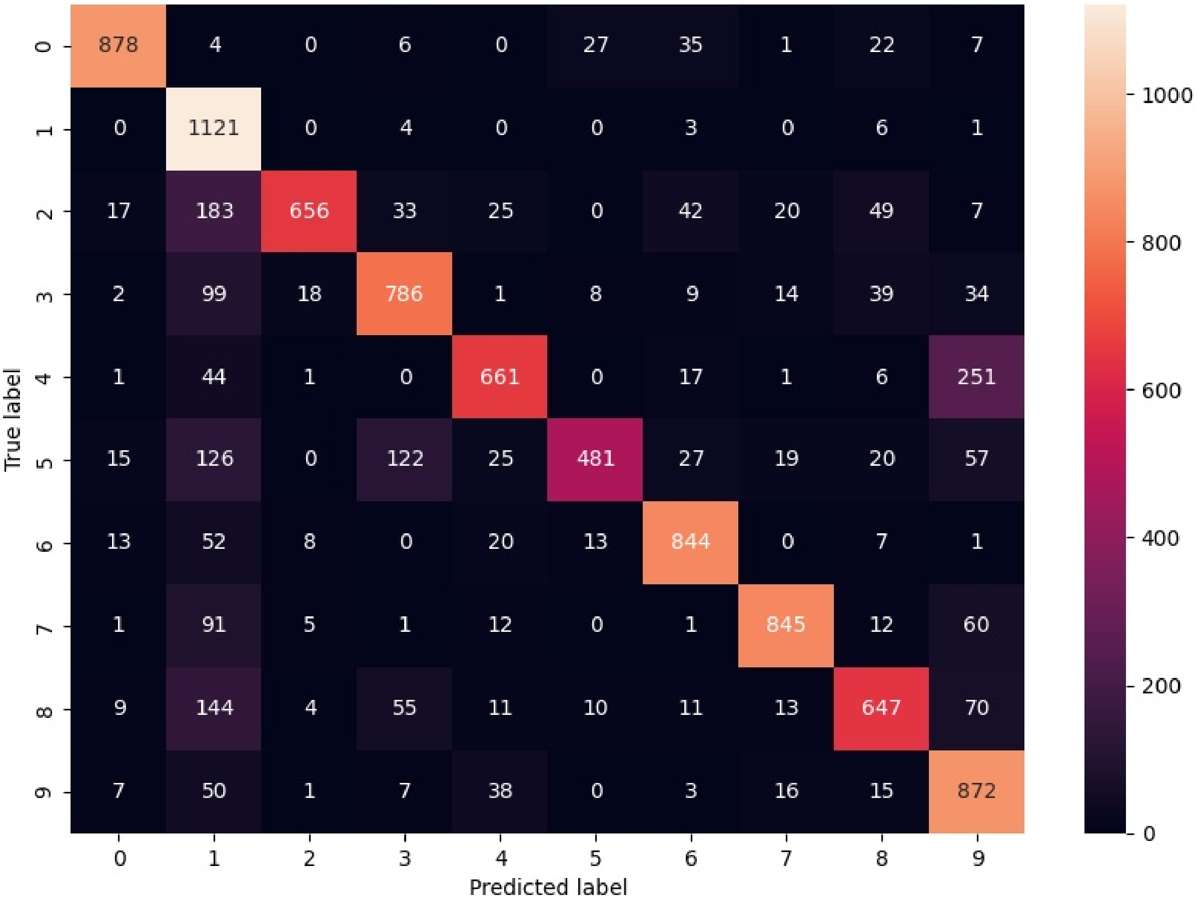
Out[22]:
```
0.7651333333333333
```

In [23]:
```python
accuracy(y_test,y_hat_test)
```

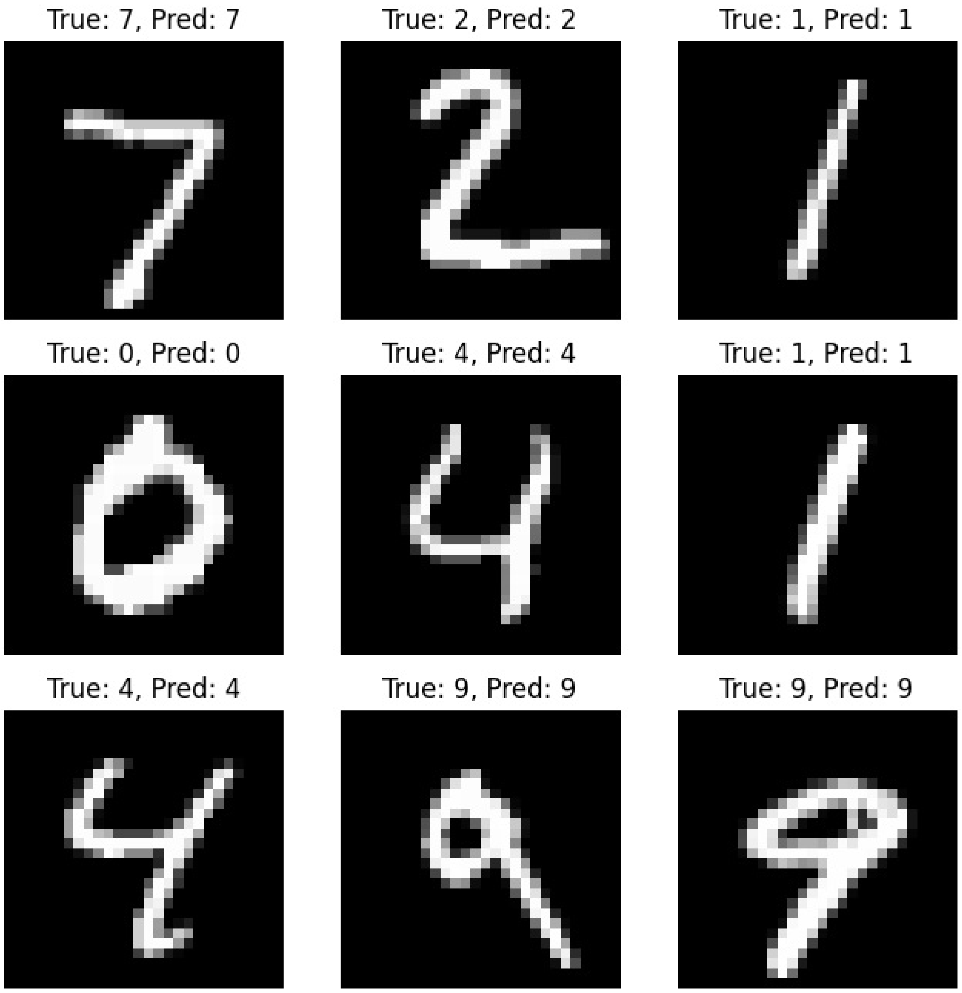
Out[23]:
```
0.7791
```

In [24]:
```python
import seaborn as sns
plt.figure(figsize=(10,7))
y_actu = pd.Series(y_test, name='Actual')
y_pred = pd.Series(y_hat_test, name='Predicted')
cm = pd.crosstab(y_actu, y_pred)
ax = sns.heatmap(cm, annot=True, fmt="d")
plt.ylabel('True label')
plt.xlabel('Predicted label')
```

Out[24]:
```
Text(0.5, 47.7222222222222, 'Predicted label')
```



In [25]:
```python
correct_indices = np.where(y_pred == y_test[:60000])[0]
fig, axes = plt.subplots(3, 3, figsize=(8, 8))
for i, ax in enumerate(axes.flat):
    if i < len(correct_indices):
        index = correct_indices[i]
        # Reshape the image data to 28x28
        image = X_test[index].reshape(28, 28)
        ax.imshow(image, cmap='gray')
        ax.set_title(f"True: {y_test[index]}, Pred: {y_pred[index]}")
        ax.axis('off')
    else:
        ax.axis('off')
plt.show()
```



True: 7, Pred: 7

True: 2, Pred: 2

True: 1, Pred: 1

True: 0, Pred: 0

True: 4, Pred: 4

True: 1, Pred: 1

True: 4, Pred: 4

True: 9, Pred: 9

True: 9, Pred: 9