## Importing libraries and datasets

In [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import multivariate_normal as mvn
import seaborn as sn
sn.set()
```

In [4]:
```
train_data=pd.read_csv('C:/Users/Consultant/Documents/machine-learning/MNIST_train.csv')
test_data=pd.read_csv('C:/Users/Consultant/Documents/machine-learning/MNIST_test.csv')
```

## working on the training data

In [5]:
```
train_data
```

Out[5]:

| | Unnamed: 0 | index | labels | 0 | 1 | 2 | 3 | 4 | 5 | 6 | ... | 774 | 775 | 776 | 777 | 778 | 779 | 780 | 781 | 782 | 783 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **1** | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **2** | 2 | 2 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **3** | 3 | 3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4** | 4 | 4 | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **59995** | 59995 | 59995 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **59996** | 59996 | 59996 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **59997** | 59997 | 59997 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **59998** | 59998 | 59998 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **59999** | 59999 | 59999 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

60000 rows × 787 columns

In [6]:
```
train_data = train_data.drop(train_data.columns[[0, 1]], axis='columns')
```

In [7]:
```
train_data
```

Out[7]:

| | labels | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | ... | 774 | 775 | 776 | 777 | 778 | 779 | 780 | 781 | 782 | 783 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **2** | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **3** | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4** | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **59995** | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **59996** | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **59997** | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **59998** | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **59999** | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

60000 rows × 785 columns

In [8]:
```
#null values
train_data.isnull().sum()
```

Out[8]:
```
labels    0
0         0
1         0
2         0
3         0
         ..
779       0
780       0
781       0
782       0
783       0
Length: 785, dtype: int64
```

In [9]:
```
y=train_data['labels']
y
```

Out[9]:
```
0        5
1        0
2        4
3        1
4        9
        ..
59995    8
59996    3
59997    5
59998    6
59999    8
Name: labels, Length: 60000, dtype: int64
```

In [10]:
```
X=train_data.drop(['labels'],axis=1)
X
```

Out[10]:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... | 774 | 775 | 776 | 777 | 778 | 779 | 780 | 781 | 782 | 783 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **2** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **3** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **59995** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **59996** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **59997** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **59998** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **59999** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

60000 rows × 784 columns

In [11]:
```
y.shape
```

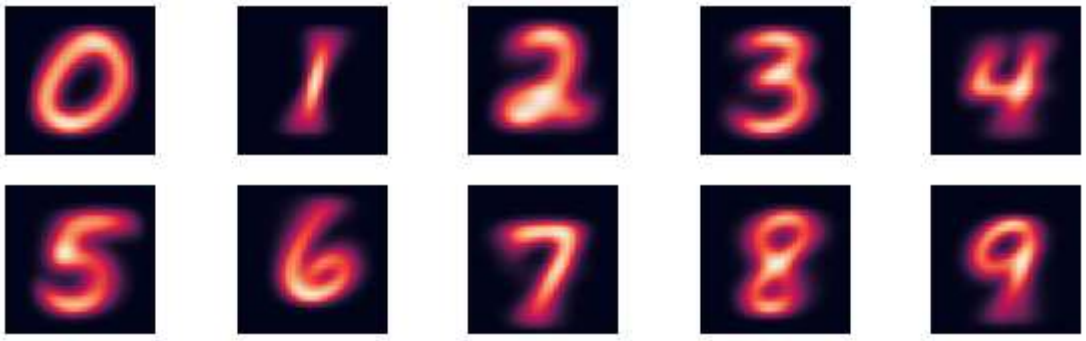
Out[11]:
```
(60000,)
```

In [12]:
```
X.shape
```

Out[12]:
```
(60000, 784)
```

## Plot the labels

In [13]:
```
plt.figure(figsize=(10,3))
for i in range(10):
    avgImg = np.average(train_data.loc[train_data["labels"]==i].drop(["labels"], axis = 1),0)
    plt.subplot(2, 5, i+1)
    plt.imshow(avgImg.reshape((28,28)))
    plt.axis('off')
```



## Gaussian Bayes Classifier (non Naive)

In [14]:
```
class GaussBayes():

    def fit(self, x, y, epsilon = 1e3):
        self.likelihoods ={}
        self.priors = {}
        self.k = set(y.astype(int))

        for k in self.k:
            x_k = x[y==k,:]
            N_k, D = x_k.shape
            mu_k = x_k.mean(axis = 0)

            self.likelihoods[k] = {"mean": x_k.mean(axis = 0), "cov": (1/(N_k-1)) *
                                    np.matmul((x_k - mu_k).T, x_k - mu_k)+ epsilon * np.identity(D)} #Use identit
            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 [15]:
```
X = X.to_numpy()
y = y.to_numpy()
```

In [16]:
```
GB = GaussBayes()
```

In [17]:
```
GB.fit(X,y)
```

In [18]:
```
preds = GB.predict(X)
```

In [19]:
```
preds
```

Out[19]:
```
array([5, 0, 4, ..., 5, 6, 8], dtype=int64)
```

In [20]:
```
def accuracy(y,y_hat):
    return np.mean(y==y_hat)
```

In [21]:
```
accuracy(y,preds)
```

Out[21]:
```
0.95805
```

## working on the test data

In [22]:
```
test_data
```

Out[22]:

| | Unnamed: 0 | index | labels | 0 | 1 | 2 | 3 | 4 | 5 | 6 | ... | 774 | 775 | 776 | 777 | 778 | 779 | 780 | 781 | 782 | 783 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **1** | 1 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **2** | 2 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **3** | 3 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4** | 4 | 4 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **9995** | 9995 | 9995 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **9996** | 9996 | 9996 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **9997** | 9997 | 9997 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **9998** | 9998 | 9998 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **9999** | 9999 | 9999 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

10000 rows × 787 columns

In [23]:
```
test_data = test_data.drop(test_data.columns[[0, 1]], axis='columns')
```

In [24]:
```
test_data
```

Out[24]:

| | labels | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | ... | 774 | 775 | 776 | 777 | 778 | 779 | 780 | 781 | 782 | 783 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **1** | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **2** | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **3** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4** | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **9995** | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **9996** | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **9997** | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **9998** | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **9999** | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

10000 rows × 785 columns

In [25]:
```
#null values
train_data.isnull().sum()
```

Out[25]:
```
labels    0
0         0
1         0
2         0
3         0
         ..
779       0
780       0
781       0
782       0
783       0
Length: 785, dtype: int64
```

In [26]:
```
y_test=test_data['labels']
y_test
```

Out[26]:
```
0       7
1       2
2       1
3       0
4       4
       ..
9995    2
9996    3
9997    4
9998    5
9999    6
Name: labels, Length: 10000, dtype: int64
```

In [27]:
```
X_test=test_data.drop(['labels'],axis=1)
X_test
```

Out[27]:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... | 774 | 775 | 776 | 777 | 778 | 779 | 780 | 781 | 782 | 783 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **2** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **3** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **9995** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **9996** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **9997** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **9998** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **9999** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

10000 rows × 784 columns

In [28]:
```
X_test = X_test.to_numpy()
y_test = y_test.to_numpy()
```

In [29]:
```
preds_test = GB.predict(X_test)
```

In [30]:
```
preds_test
```

Out[30]:
```
array([7, 2, 1, ..., 4, 5, 6], dtype=int64)
```

In [31]:
```
accuracy(y_test,preds_test)
```

Out[31]:
```
0.9511
```

In [32]:
```
preds_test
```

Out[32]:
```
array([7, 2, 1, ..., 4, 5, 6], dtype=int64)
```

## Confusion matrix

In [33]:
```
actual=pd.Series(y_test.tolist(),name='actual')
predictions=pd.Series(preds_test.tolist(),name='predictions')
confusion_matrix=pd.crosstab(actual,predictions)
confusion_matrix
```

Out[33]:

| predictions | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| actual | | | | | | | | | | |
| **0** | 968 | 0 | 1 | 1 | 0 | 0 | 3 | 1 | 6 | 0 |
| **1** | 0 | 1120 | 6 | 1 | 0 | 0 | 5 | 0 | 3 | 0 |
| **2** | 3 | 3 | 973 | 9 | 3 | 0 | 1 | 6 | 33 | 1 |
| **3** | 5 | 0 | 4 | 943 | 0 | 12 | 0 | 5 | 31 | 10 |
| **4** | 0 | 2 | 3 | 0 | 938 | 0 | 4 | 2 | 3 | 30 |
| **5** | 2 | 0 | 2 | 20 | 0 | 820 | 12 | 3 | 25 | 8 |
| **6** | 6 | 3 | 1 | 0 | 3 | 9 | 928 | 0 | 8 | 0 |
| **7** | 0 | 11 | 13 | 2 | 10 | 1 | 0 | 944 | 5 | 42 |
| **8** | 8 | 5 | 7 | 14 | 3 | 4 | 2 | 4 | 918 | 9 |
| **9** | 5 | 6 | 4 | 8 | 6 | 1 | 0 | 6 | 14 | 959 |

In [ ]: