

MNIST_clf

March 8, 2021

1 MNIST CLASSIFICATION

1.1 Importing Data

```
[4]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from config import CONFIG

import sys
sys.path.insert(0, "../src")
sys.path.insert(1, "../../GaussBayes/bayesTheorem/src/")
```

```
[5]: %load_ext autoreload
```

```
[27]: %autoreload 2
from helper import mnist_show_me, confusion_matrix
from KNN import *
from GaussNB import *
```

```
[7]: mnist_train = pd.read_csv(CONFIG.data / "MNIST_train.csv")
print(f"train data:")
display(mnist_train.head())
print(f"data shape: {mnist_train.shape}")
```

train data:

	Unnamed: 0	index	labels	0	1	2	3	4	5	6	...	774	775	776	777	\
0	0	0	5	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	
2	2	2	4	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	
4	4	4	9	0	0	0	0	0	0	0	...	0	0	0	0	

	778	779	780	781	782	783
0	0	0	0	0	0	0
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0

```
4    0    0    0    0    0    0
```

```
[5 rows x 787 columns]
```

```
data shape: (60000, 787)
```

```
[32]: X_train, y_train = mnist_train.iloc[:, 3:].to_numpy(), mnist_train.iloc[:, 2].  
      ↪to_numpy()  
      X_train = X_train / 255
```

```
[9]: mnist_test = pd.read_csv(CONFIG.data / "MNIST_test.csv")  
     print(f"test data:")  
     display(mnist_train.head())  
     print(f"test data shape: {mnist_test.shape}")
```

```
test data:
```

	Unnamed: 0	index	labels	0	1	2	3	4	5	6	...	774	775	776	777	\
0	0	0	5	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	
2	2	2	4	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	
4	4	4	9	0	0	0	0	0	0	0	...	0	0	0	0	

	778	779	780	781	782	783
0	0	0	0	0	0	0
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0

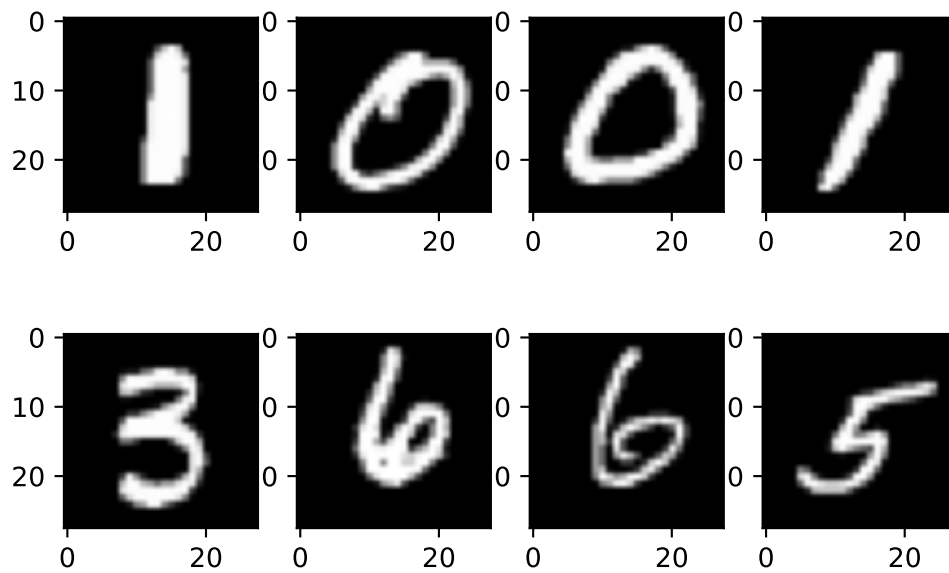
```
[5 rows x 787 columns]
```

```
test data shape: (10000, 787)
```

```
[33]: X_test, y_test = mnist_test.iloc[:, 3:].to_numpy(), mnist_test.iloc[:, 2].  
      ↪to_numpy()  
      X_test = X_test / 255
```

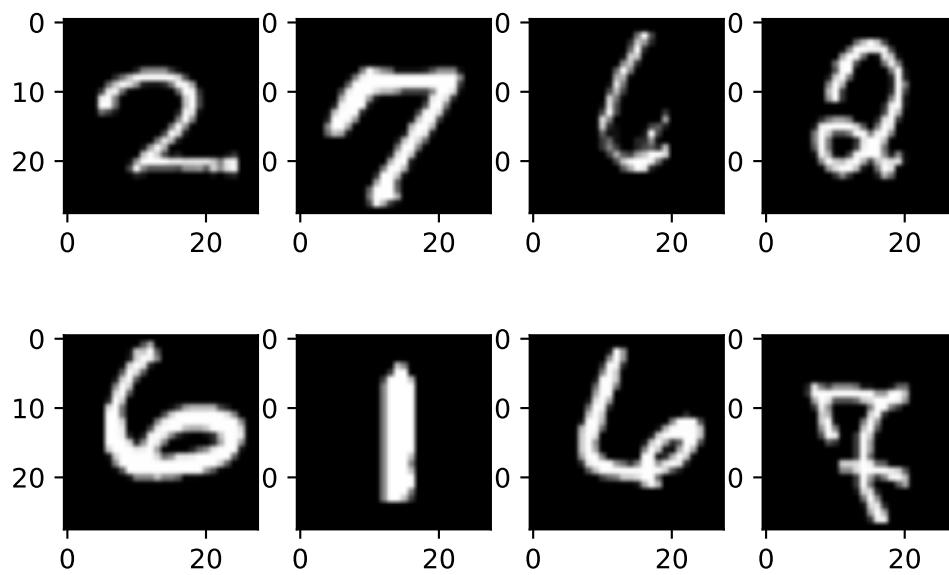
```
[28]: train_indices = np.random.randint(0, len(X_train), 8)  
      list_of_images = [x.reshape(28, 28) for x in X_train[train_indices]]  
      mnist_show_me(list_of_images, title="Train Images")
```

Train Images



```
[29]: indices = np.random.randint(0, len(X_test), 8)
list_of_test_images = [x.reshape(28, 28) for x in X_test[indices]]
mnist_show_me(list_of_test_images, title="Test Images")
```

Test Images



1.2 GAUSSIAN BAYES CLASSIFIER

```
[34]: gauss_clf = GaussBayes()

gauss_clf.fit(X_train, y_train)
gauss_preds = gauss_clf.predict(X_test)
```

```
[35]: def accuracy(y_test, preds):
        return np.mean(y_test==preds)
print(f"Model's accuracy: {accuracy(y_test, gauss_preds)}")
```

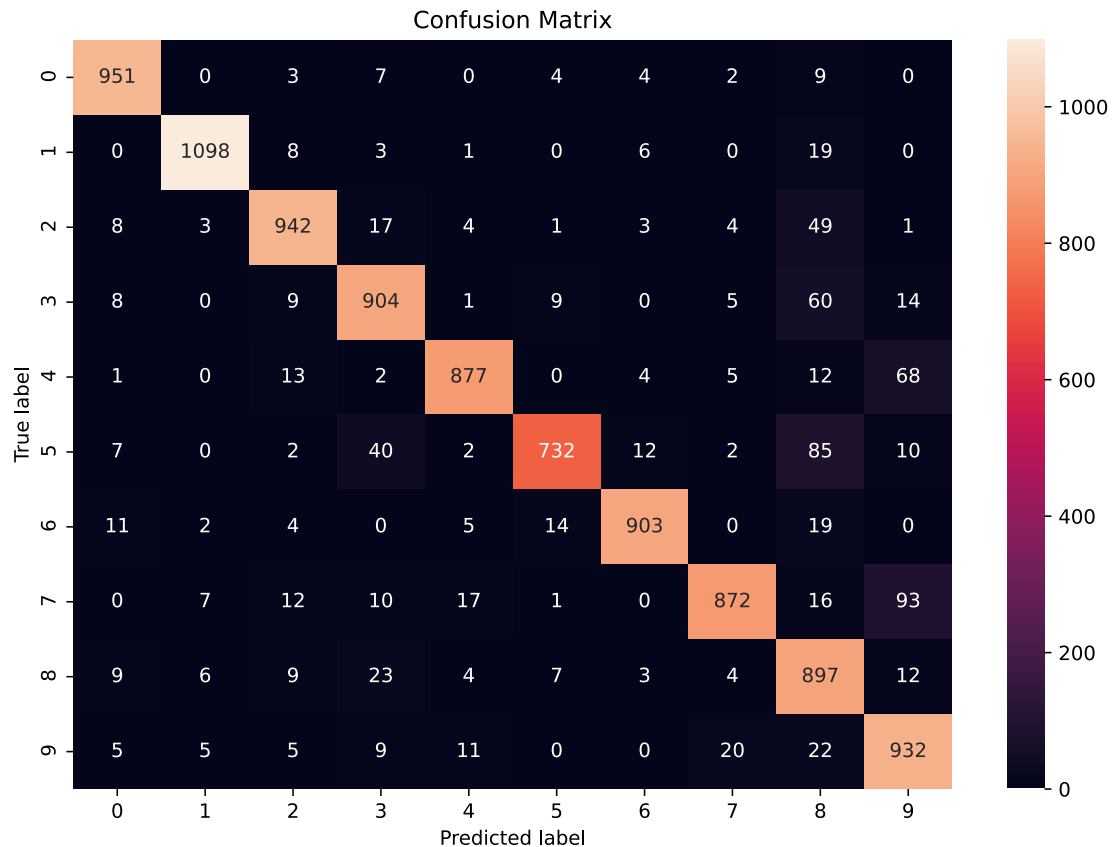
Model's accuracy: 0.9108

```
[40]: from sklearn.model_selection import KFold
kf = KFold(n_splits=5)
acc = []
for train_indices, test_indices in kf.split(X_train):
    xtrain, xtest = X_train[train_indices], X_train[test_indices]
    ytrain, ytest = y_train[train_indices], y_train[test_indices]
    gauss_clf = GaussBayes()
    gauss_clf.fit(xtrain, ytrain)
    preds = gauss_clf.predict(xtest)
    score = accuracy(ytest, preds)
    acc.append(score)
print(f"Model's accuracy: {np.mean(acc):.3f} +/- {np.std(acc)}")
```

Model's accuracy: 0.905 +/- 0.0027090383697705167

```
[43]: confusion_matrix(y_true=y_test, y_preds=gauss_preds)
plt.title("Confusion Matrix")
```

```
[43]: Text(0.5, 1.0, 'Confusion Matrix')
```



1.3 KNN Classifier

```
[18]: import time
```

```
[15]: knn = KNNClassifier()
```

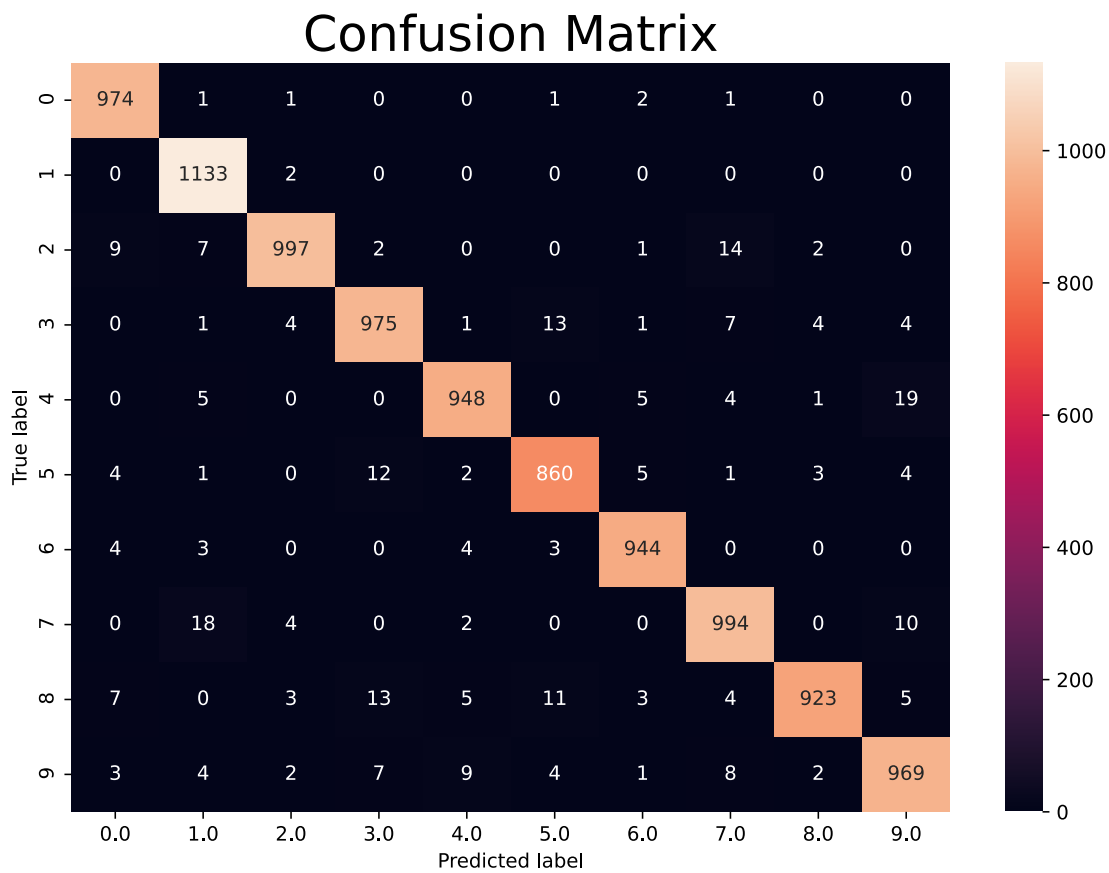
```
[ ]: start = time.time()
      knn.fit(X_train, y_train)
      preds = knn.predict(X_test, K=3)
      accuracy(y_test, preds)
      knn_time = time.time() - start
      print(f"processing time: {knn_time:.3f}")
```

```
[23]: np.mean(y_test==preds)
```

```
[23]: 0.9717
```

```
[26]: confusion_matrix(y_test, preds)
      plt.title("Confusion Matrix", fontsize=25)
```

```
[26]: Text(0.5, 1.0, 'Confusion Matrix')
```



```
[ ]:
```