

```
In [31]: import numpy as np
from sklearn.model_selection import train_test_split

rawData = np.load('data_train_correct.npy')
rawLabels = np.load('labels_train_corrected.npy')
```

```
In [32]: rawData = rawData.T
print(rawData[4].shape)

(90000,)
```

```
In [33]: data = []
labels = []
for i in range(9032):
    if (rawLabels[i] != 10):
        data.append(rawData[i])
        labels.append(rawLabels[i])

data = np.array(data)
labels = np.array(labels)

print(np.shape(data))
print(np.shape(labels))

(8914, 90000)
(8914,)
```

```
In [53]: X_train, X_test, y_train, y_test = train_test_split(data, labels, test_size=0.2, random
print(X_train.shape, X_test.shape)

(7131, 90000) (1783, 90000)
```

```
In [51]: from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics

for k in range(1, 10):
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train, y_train)
    accuracy = knn.score(X_test, y_test)
    print('k value:', k, 'Accuracy:', accuracy)

k value: 1 Accuracy: 0.4554122265844083
k value: 2 Accuracy: 0.4178351093662367
k value: 3 Accuracy: 0.42344363432417276
k value: 4 Accuracy: 0.4195176668536175
k value: 5 Accuracy: 0.41615255187885586
k value: 6 Accuracy: 0.4094223219293326
k value: 7 Accuracy: 0.4038137969713965
k value: 8 Accuracy: 0.3841839596186203
k value: 9 Accuracy: 0.3869882220975883
```

```
In [52]: from sklearn.metrics import classification_report
knn = KNeighborsClassifier(n_neighbors=1)
```

```
knn.fit(X_train, y_train)
y_pred = knn.predict(X_test)
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0.0	0.46	0.37	0.41	183
1.0	0.50	0.39	0.44	166
2.0	0.47	0.53	0.50	194
3.0	0.37	0.76	0.50	180
4.0	0.43	0.52	0.47	192
5.0	0.46	0.41	0.44	173
6.0	0.60	0.46	0.52	180
7.0	0.63	0.40	0.49	179
8.0	0.34	0.27	0.30	174
9.0	0.46	0.43	0.44	162
accuracy			0.46	1783
macro avg	0.47	0.45	0.45	1783
weighted avg	0.47	0.46	0.45	1783

```
In [60]: from sklearn.metrics import confusion_matrix

y_train_pred = knn.predict(X_train)
```

```
In [58]: confusion_matrix(y_train, y_train_pred)
```

```
Out[58]: array([[702,  0,  0,  0,  0,  0,  0,  0,  0,  0],
 [  0, 730,  0,  0,  0,  0,  0,  0,  0,  0],
 [  0,  0, 699,  0,  0,  0,  0,  0,  0,  0],
 [  0,  0,  0, 712,  0,  0,  0,  0,  0,  0],
 [  0,  0,  0,  0, 688,  0,  0,  0,  0,  0],
 [  0,  0,  0,  0,  0, 724,  0,  0,  0,  0],
 [  0,  0,  0,  0,  0,  0, 715,  0,  0,  0],
 [  0,  0,  0,  0,  0,  0,  0, 706,  0,  0],
 [  0,  0,  0,  0,  0,  0,  0,  0, 719,  0],
 [  0,  0,  0,  0,  0,  0,  0,  0,  0, 736]], dtype=int64)
```

```
In [59]: confusion_matrix(y_test, y_pred)
```

```
Out[59]: array([[ 67, 10, 14, 12, 13, 35,  8,  2, 13,  9],
 [ 12, 65, 13, 18, 16,  7,  9,  1, 13, 12],
 [  3,  2,103, 57,  9,  4,  0,  5,  5,  6],
 [  3,  0, 14,136, 16,  3,  2,  0,  4,  2],
 [  4,  5,  9, 52, 99,  3,  4,  1,  6,  9],
 [ 17, 12, 15, 15, 15, 71,  9,  0,  9, 10],
 [  7,  2, 12, 31, 15,  5, 83,  6,  9, 10],
 [  7, 11, 13, 19, 15,  5,  7, 72, 21,  9],
 [ 14, 16, 16, 15, 12, 12,  6, 23, 47, 13],
 [ 12,  7,  8, 10, 19,  8, 11,  5, 13,  6]], dtype=int64)
```