

## תרגול 9 - זיהוי ספרות בכתב ידי scikit-learn

**שלב א'** (חקירת מבנה הנתונים mnist\_784 תוך שימוש scikit-learn)  
להלן תוכנית הכתובה בשפת python הרץ אותה וענה על השאלות הבאות:

```
import numpy as np
import matplotlib.pyplot as plt
from pathlib import Path
from sklearn.datasets import fetch_openml

datapath = Path("mnist_784.npz")
if not(datapath.exists()):
    print("downloading file...")
    x_mnist, y_mnist = fetch_openml("mnist_784", version=1,
                                    return_X_y=True, data_home=".")
    x=np.array(x_mnist,dtype="u8")
    y=np.array(y_mnist,dtype="u8")
    np.savez(datapath,x=x,y=y)
    del x_mnist, y_mnist
print("Loading file...")
data = np.load(datapath)
print("\ntype: ", type(data))
print("\ntype: ", data.files)
x_mnist = data["x"]
y_mnist = data["y"]

plt.figure()
for i in range(200):
    plt.subplot(10,20,i+1)
    plt.axis("off")
    plt.imshow(x_mnist[i].reshape((28,28)),cmap="gray", vmin=0, vmax=255)
plt.show()
```

1. מה פלט התוכנית?
2. האם זמן הרצה בפעם הראשונה שונה מזמן ההרצה בפעם השנייה? נמק את תשובתך.
3. הסבר מה עושה הפעולה exists שבמחלקה Path מתוך הספרייה pathlib.

<https://www.geeksforgeeks.org/python-os-path-exists-method/>

4. הסבר מה עושות הפעולות np.load ו- np.savez

<https://www.kite.com/python/docs/numpy.lib.npyio.NpzFile>

5. מה צריך לשנות בתוכנית במטרה להציג יותר תמונות מספרים על המסך.

**שלב ב'** (אימון רשת לצורך סיווג ספרות מתוך בסיס נתונים mnist\_784 תוך שימוש scikit-learn)

המלצה למבנה רשת:

```
mlp = MLPClassifier(  
    hidden_layer_sizes=(50,),  
    max_iter=40,  
    solver='sgd',  
    verbose=True,  
    learning_rate_init=0.1)
```

כדי לבדוק את ביצועי המערכת ניתן להיעזר בפעולות הבאות:

```
print("Training set score: %f" % mlp.score(train_data, train_lbl))  
print("Test set score: %f" % mlp.score(test_data, test_lbl))
```

דוגמה לפלט רצוי:

```
Iteration 29, loss = 0.00628542  
Iteration 30, loss = 0.00462353  
Iteration 31, loss = 0.00402229  
Iteration 32, loss = 0.00366734  
Iteration 33, loss = 0.00301430  
Iteration 34, loss = 0.00295780  
Iteration 35, loss = 0.00282633  
Iteration 36, loss = 0.00254278  
Iteration 37, loss = 0.00228591  
Iteration 38, loss = 0.00217346  
Iteration 39, loss = 0.00211823  
Iteration 40, loss = 0.00198818  
Training set score: 1.000000  
Test set score: 0.976000
```

**שלב ג'** (שיפור יכולות רשת לסיווג ספרות תוך שימוש scikit-learn)

להלן 3 פלטים של confusion\_matrix שנדגמו בשלוש הרצות שונות של התוכנה. מי מבין 3 המטריצות הבאות נותן את הביצועים הטובים ביותר:

<pre>[[ 968  1  0  1  0  4  4  1  1  0]  [  1128  2  1  0  1  1  0  2  0]  [  1  1015  2  1  0  1  5  7  0]  [  0  0  4  988  0  8  0  4  4  2]  [  1  0  3  0  962  0  4  2  3  7]  [  2  0  0  6  0  869  7  0  7  1]  [  3  2  0  1  4  4  943  0  1  0]  [  0  2  13  0  0  1  0 1003  5  4]  [  2  0  3  2  3  3  3  2  954  2]  [  0  2  0  3  12  4  1  5  10  972]]</pre>	<pre>[[ 965  0  1  1  2  0  6  2  3  0]  [  1122  2  2  0  1  2  1  5  0]  [  3  1 1002  4  4  0  3  5  10  0]  [  0  0  3  987  0  10  0  3  4  3]  [  2  2  3  2  948  1  7  4  2  11]  [  2  0  0  11  2  864  5  1  6  1]  [  3  3  1  1  1  8  937  1  2  1]  [  0  4  14  10  1  0  0  983  6  10]  [  6  2  0  3  4  5  3  3  943  5]  [  3  4  0  7  10  7  0  7  8  963]]</pre>	<pre>[[  0  0  0  980  0  0  0  0  0  0]  [  0  0  0 1135  0  0  0  0  0  0]  [  0  0  0 1032  0  0  0  0  0  0]  [  0  0  0 1010  0  0  0  0  0  0]  [  0  0  0  982  0  0  0  0  0  0]  [  0  0  0  892  0  0  0  0  0  0]  [  0  0  0  958  0  0  0  0  0  0]  [  0  0  0 1028  0  0  0  0  0  0]  [  0  0  0  974  0  0  0  0  0  0]  [  0  0  0 1009  0  0  0  0  0  0]]</pre>
---	--	---

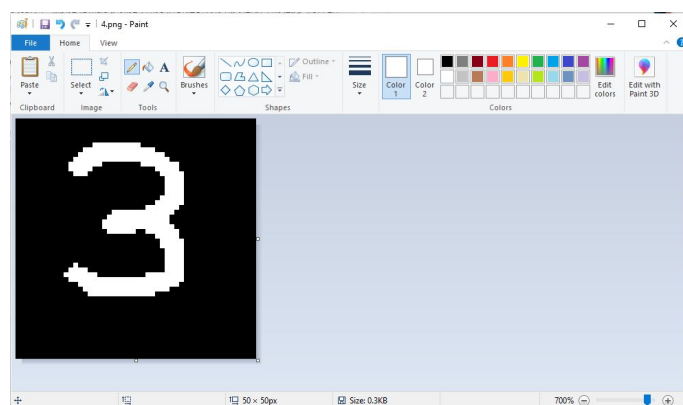
היעזרו ב- confusion\_matrix במטרה להגיע לביצועים טובים יותר ביכולת הסיווג של המכנה.

```
from sklearn.metrics import confusion_matrix  
  
.....  
print(confusion_matrix(test_lbl, mlp.predict(test_data)))
```

**שלב ד' (סיווג ספרות מתמונות אישיות תוך שימוש scikit-learn)**

ממשו קוד הקולט תמונה המכילה ספרה בודדת. על התוכנית לסווג את התמונה תוך שימוש באלגוריתם שמימשנו עד כה.

לדוגמה עבור:



נקבל:

```
Predict data: [3]
```