

## ▼ CS156 (Introduction to AI), Fall 2022

### Homework 6 submission

Roster Name: Preet LNU

Student ID: 014755741

Email address: [preet.lnu@sjsu.edu](mailto:preet.lnu@sjsu.edu)

### ▼ References and sources

Perceptron.Breast

### ▼ Solution

#### ▼ Load libraries and set random number generator seed

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import Perceptron
from sklearn.metrics import plot_confusion_matrix

from sklearn.model_selection import train_test_split

np.random.seed(42)
```

Code the solution

#### ▼ Load the dataset

```

digits = datasets.load_digits()
X = digits.data
X = X.astype("float32") / 255
Y = digits.target
class_names = digits.target_names
X.shape, Y.shape, class_names

((1797, 64), (1797,), array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]))

digits_df = pd.DataFrame(X, columns=digits.feature_names)
digits_df['output_digit'] = Y
digits_df.head()

```

	pixel_0_0	pixel_0_1	pixel_0_2	pixel_0_3	pixel_0_4	pixel_0_5	pixel_0_6	pi
0	0.0	0.0	0.019608	0.050980	0.035294	0.003922		0.0
1	0.0	0.0	0.000000	0.047059	0.050980	0.019608		0.0
2	0.0	0.0	0.000000	0.015686	0.058824	0.047059		0.0
3	0.0	0.0	0.027451	0.058824	0.050980	0.003922		0.0
4	0.0	0.0	0.000000	0.003922	0.043137	0.000000		0.0

5 rows x 65 columns



## ▼ Converting digits to binary classifier

```

df_numeric = pd.get_dummies(digits_df, columns=['output_digit'], prefix=['output_digit'])

converted = ['output_digit_0', 'output_digit_1', 'output_digit_2', 'output_digit_3',
            'output_digit_4', 'output_digit_5', 'output_digit_6', 'output_digit_7',
            'output_digit_8', 'output_digit_9']

df_numeric

```

	pixel_0_0	pixel_0_1	pixel_0_2	pixel_0_3	pixel_0_4	pixel_0_5	pixel_0_6
<b>0</b>	0.0	0.0	0.019608	0.050980	0.035294	0.003922	0.000000
<b>1</b>	0.0	0.0	0.000000	0.047059	0.050980	0.019608	0.000000
<b>2</b>	0.0	0.0	0.000000	0.015686	0.058824	0.047059	0.000000
<b>3</b>	0.0	0.0	0.027451	0.058824	0.050980	0.003922	0.000000
<b>4</b>	0.0	0.0	0.000000	0.003922	0.043137	0.000000	0.000000
...	...	...	...	...	...	...	...
<b>1792</b>	0.0	0.0	0.015686	0.039216	0.050980	0.023529	0.000000
<b>1793</b>	0.0	0.0	0.023529	0.062745	0.050980	0.043137	0.003922
<b>1794</b>	0.0	0.0	0.003922	0.043137	0.058824	0.003922	0.000000

Train a single layer perceptron model on the training data and compute accuracy of the model on the test data. Output a non-normalized confusion matrix for the test set prediction results.

```
for i in range(10):
    Y_new = df_numeric[converted[i]]
    X_train, X_test, Y_train, Y_test = train_test_split(X, Y_new, test_size=0.2, random_state=i)

    model = Perceptron(tol=1e-3, random_state=0).fit(X_train, Y_train)
    print('Accuracy of perceptron on training set: {:.2f}'.format(model.score(X_train, Y_train)))
    print('Accuracy of perceptron on test set: {:.2f}'.format(model.score(X_test, Y_test)))

    model_acc = "{:.2f}".format(model.score(X_test, Y_test))
    title = "Test set results for " + converted[i] + " (" + model_acc + ")"
    print('\n')

    disp = plot_confusion_matrix(model, X_test, Y_test,
                                  display_labels=['Not ' + str(i), str(i)],
                                  cmap=plt.cm.Blues,
                                  normalize=None)

    disp.ax_.set_title(title)

    print(title)
    print(disp.confusion_matrix)

plt.show()
```

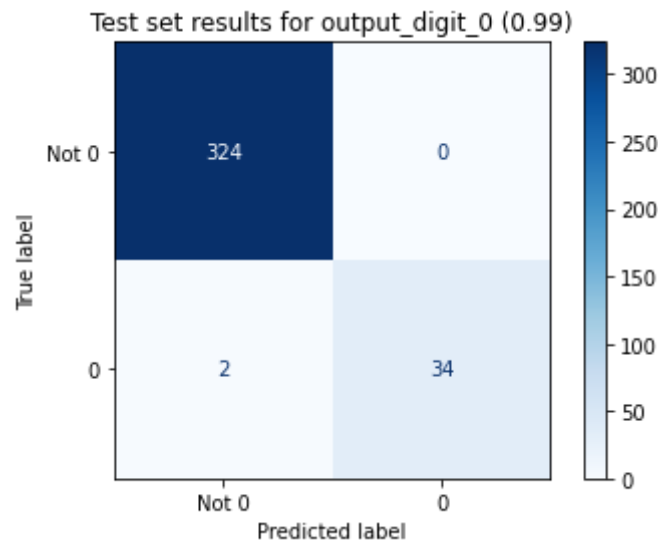
Accuracy of perceptron on training set: 0.99

Accuracy of perceptron on test set: 0.99

Test set results for output\_digit\_0 (0.99)

```
[[324  0]
 [ 2 34]]
```

/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: warnings.warn(msg, category=FutureWarning)



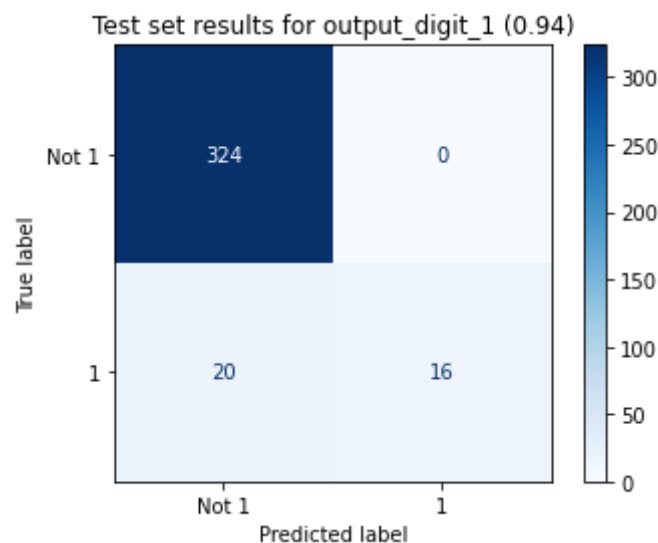
Accuracy of perceptron on training set: 0.95

Accuracy of perceptron on test set: 0.94

Test set results for output\_digit\_1 (0.94)

```
[[324  0]
 [20 16]]
```

/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: warnings.warn(msg, category=FutureWarning)



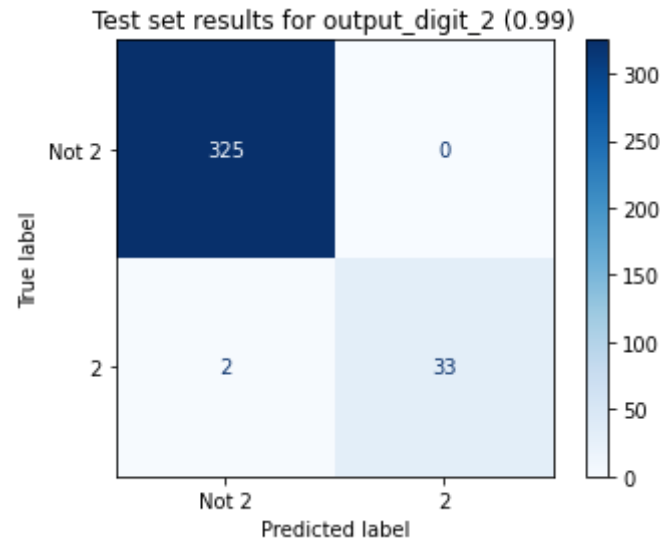
Accuracy of perceptron on training set: 0.99

Accuracy of perceptron on test set: 0.99

Test set results for output\_digit\_2 (0.99)

```
[[325  0]
```

```
[ 2 33]]
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning:
warnings.warn(msg, category=FutureWarning)
```

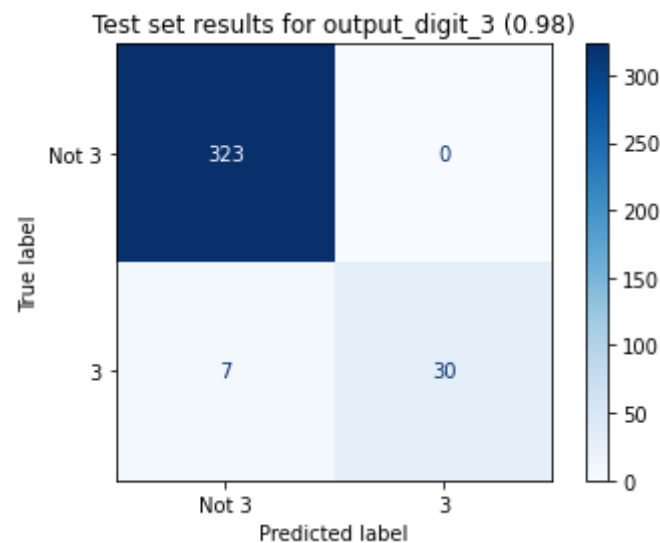


Accuracy of perceptron on training set: 0.98

Accuracy of perceptron on test set: 0.98

Test set results for output\_digit\_3 (0.98)

```
[[323  0]
 [ 7 30]]
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning:
warnings.warn(msg, category=FutureWarning)
```



Accuracy of perceptron on training set: 0.98

Accuracy of perceptron on test set: 0.97

Test set results for output\_digit\_4 (0.97)

```
[[324  0]
 [11 25]]
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning:
warnings.warn(msg, category=FutureWarning)
```

