CS156 (Introduction to AI), Fall 2021

Homework 6 submission

Roster Name: Nand Kishore Khuswaha

Student ID: 013920192

Email address: nandkishore.khuswaha@sjsu.edu (mailto:nandkishore.khuswaha@sjsu.edu)

Any special notes or anything you would like to communicate to me about this homework submission goes in here.

References and sources

1) Perceptron.Breast (class file)

Solution

Load libraries and set random number generator seed

In [1]:
```
import numpy as np
from sklearn.datasets import load_digits
import pandas as pd
from sklearn.linear_model import Perceptron
from sklearn.model_selection import train_test_split
from sklearn.linear_model import Perceptron
from sklearn.metrics import plot_confusion_matrix
import matplotlib.pyplot as plt
```

In [2]:
```
np.random.seed(42)
```

Code the solution

In [3]:
```
#Loading dataset from sklearn
dataset = load_digits()
```

In [4]:
```
X = dataset['images']
#to flatten the image
X= X.reshape(X.shape[0], -1)
#to normalize
X = X.astype("float32")/255
# print(X)
X.shape
```

Out[4]:
```
(1797, 64)
```

In [ ]:
```

```

In [5]:
```
X = dataset.data
X = X.astype("float32")/255
Y = dataset['target']

X.shape, Y.shape
```

Out[5]:
```
((1797, 64), (1797,))
```

In [ ]:
```

```

In [ ]:
```

```

In [6]:
```
#  convert categorical variables to a set of binary variables
df= pd.DataFrame(Y, columns = ['target'])
y_df = pd.get_dummies(df,columns=df.columns, prefix=df.columns)
y_df.head()
# print(y_df)

# type(y_df)
```

Out[6]:

	target_0	target_1	target_2	target_3	target_4	target_5	target_6	target_7	target_8	target_9
0	1	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0

In [7]:
```
# print(y_df.target_0)
# temp_y= y_df.target_0
# temp = y_df.iloc[:, 2].values
# print(temp)
```

In [8]:
```
# Training our model and computing accuracy
for i in [0,1, 2, 3, 4, 5,6,7,8,9]:
    temp_y= y_df.iloc[:, i].values
    X_train, X_test, Y_train, Y_test = train_test_split(X, temp_y, test_siz
    model = Perceptron(tol=1e-3, random_state=0).fit(X_train, Y_train)
    print("Model label:", i)

    print('Accuracy of perceptron on training set: {:.2f}'.format(model.scc
    print('Accuracy of perceptron on test set: {:.2f}'.format(model.score(X
```

```
Model label: 0
Accuracy of perceptron on training set: 0.99
Accuracy of perceptron on test set: 0.99
Model label: 1
Accuracy of perceptron on training set: 0.94
Accuracy of perceptron on test set: 0.94
Model label: 2
Accuracy of perceptron on training set: 1.00
Accuracy of perceptron on test set: 0.99
Model label: 3
Accuracy of perceptron on training set: 0.99
Accuracy of perceptron on test set: 0.98
Model label: 4
Accuracy of perceptron on training set: 0.98
Accuracy of perceptron on test set: 0.98
Model label: 5
Accuracy of perceptron on training set: 0.99
Accuracy of perceptron on test set: 0.99
Model label: 6
Accuracy of perceptron on training set: 0.99
Accuracy of perceptron on test set: 1.00
Model label: 7
Accuracy of perceptron on training set: 0.99
Accuracy of perceptron on test set: 0.99
Model label: 8
Accuracy of perceptron on training set: 0.92
Accuracy of perceptron on test set: 0.93
Model label: 9
Accuracy of perceptron on training set: 0.94
Accuracy of perceptron on test set: 0.93
```

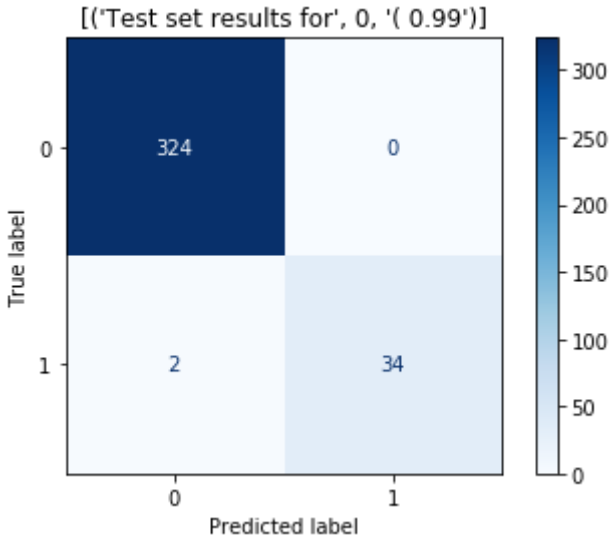
In [9]:
```
for i in [0,1, 2, 3, 4, 5,6,7,8,9]:
    temp_y= y_df.iloc[:, i].values
    X_train, X_test, Y_train, Y_test = train_test_split(X, temp_y, test_siz
    model = Perceptron(tol=1e-3, random_state=0).fit(X_train, Y_train)

    np.set_printoptions(precision=2)
    titles_options = [ ("Test set results for", i, '( {:.2f}'.format(model.


    disp = plot_confusion_matrix(model, X_test, Y_test, cmap=plt.cm.Blues)
    disp.ax_.set_title(titles_options)

#     print(titles_options)
#     print(disp.confusion_matrix)
    plt.show()
```

```
  warnings.warn(msg, category=FutureWarning)
```




```
/Users/becoming1/anaconda3/lib/python3.7/site-packages/sklearn/utils/depr
```

In [ ]:
```
```