## CS156 (Introduction to AI), Spring 2021

## **Homework 6 submission**

Roster Name: Neeval Kumar

Preferred Name (if different): Chosen Name

Student ID: 011877086

Any special notes or anything you would like to communicate to me about this homework submission

Email address: kumar.neeval@gmail.com

goes in here. References and sources

## where you grabbed some code examples.

**Solution** 

List all your references and sources here. This includes all sites/discussion boards/blogs/posts/etc.

## In [131... import numpy as np

import pandas as pd from sklearn import datasets

Load libraries and set random number generator seed

```
import matplotlib.pyplot as plt
           import seaborn as sns
           from sklearn.model selection import train test split
           \textbf{from} \text{ sklearn.preprocessing } \textbf{import} \text{ StandardScaler}
           from sklearn.linear_model import Perceptron
           from sklearn.metrics import plot_confusion matrix
In [132... np.random.seed(42)
         Load the dataset and flatten numbers
```

In [133... digits = datasets.load digits()

In [135... def replaceLabels(Y, label):

X = digits.data Y = digits.target

for index, value in enumerate(Y):

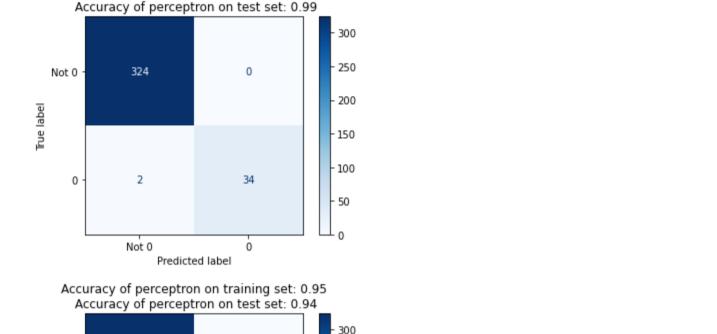
```
class_names = digits.target_names
          X.shape, Y.shape, class names
Out[133... ((1797, 64), (1797,), array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]))
In [134...] X = X.astype("float32") / 255
         Just a method to replace the variables in Y based on the class label we are looping
```

```
through. Basically the 0 or not 0 aspect in our confusion matrix. If it is the label, then
we replace the element with 1, otherwise 0.
```

if value == label: Y[index] = 1else: Y[index] = 0return Y

```
Loop through each unique class label, and replace the Y with the labels from the
method before. Then split and create a model and perform a confusion matrix on that
specific model.
 digits df = pd.DataFrame(X,columns=digits.feature names)
 for index, label in enumerate(class names):
     scaled Y = replaceLabels(list(Y), label)
     X train, X test, Y train, Y test = train test split(digits df, scaled Y, test size
     model = Perceptron(tol=1e-3, random state=0)
     model.fit(X train, Y train)
```

title = 'Accuracy of perceptron on training set: {:.2f}'.format(model.score(X tra: display\_labels = ['Not ' + str(label), str(label)]
disp = plot\_confusion\_matrix(model, X\_test, Y\_test, display labels = display labels, cmap=plt.cm.Blues) disp.ax .set title(title) plt.show() Accuracy of perceptron on training set: 0.99



250

200

150

250

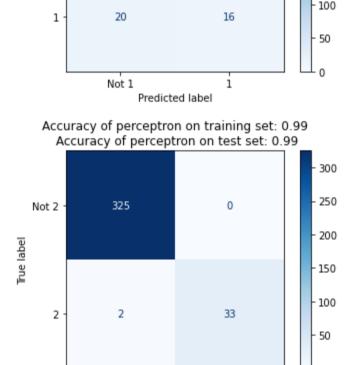
50

300

250

300

0



Predicted label

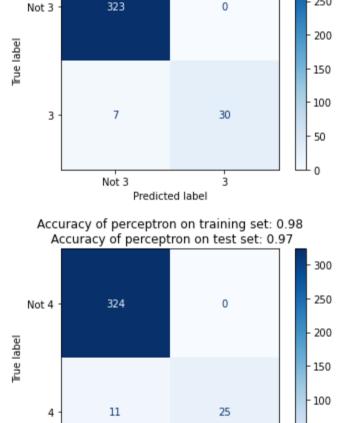
Accuracy of perceptron on training set: 0.98 Accuracy of perceptron on test set: 0.98

324

Not 2

Not 1

Frue label



Not 4

323

Not 5

Predicted label

Accuracy of perceptron on training set: 0.99 Accuracy of perceptron on test set: 0.99

0

