hw1_1

August 29, 2018

0.1 Importing packages

```
In [1]: import numpy as np
```

0.2 Parameters

```
In [2]: x_test = [-1,-1]
```

0.3 Loading the dataset

```
In [3]: x_train = np.transpose(np.genfromtxt('X.csv',delimiter=','))    # Loading training datas
        y_train = np.genfromtxt('Y.csv',delimiter=',')                  #loading the correspodin
        print(x_train)
```

```
[[ 1.67705596 -0.91408325]
 [-1.91652611 -1.30496213]
 [ 1.65105414 -1.08038307]
 ...
 [-1.02465078  0.25885031]
 [ 0.730965    0.54656083]
 [ 1.51459784  0.7214577 ]]
```

0.4 Sperating the training set based on the labels

```
In [4]: x_train_1 = []
        x_train_0 = []
        for i in range(0,1000):
            if y_train[i] == 1:
                x_train_1.append(x_train[i])     # appending into new array for labels = 1
            else:
                x_train_0.append(x_train[i])     # appending into new array for labels = -1
        x_train_1 = np.asarray(x_train_1)        #converting the lists to arrays
        x_train_0 = np.asarray(x_train_0)
        print(x_train_1.shape)
        print(x_train_0.shape)
```

```
(493, 2)
(507, 2)
```

0.5 Fitting the distributions(finding mean and variances)

```
In [5]: mean_1 = np.mean(x_train_1,axis = 0)
        mean_0 = np.mean(x_train_0,axis = 0)
        print(mean_1)
        print(mean_0)

        var_1 = np.var(x_train_1,axis = 0)
        var_0 = np.var(x_train_0,axis = 0)
        print(var_1)
        print(var_0)
```

```
[0.83139867 0.05820254]
[-0.81491941 -0.03831222]
[0.3699126  0.93099005]
[0.36732626 0.967272  ]
```

1 Testing

1.0.1 Finding the probabilities

```
In [6]: p_1 = np.exp(-((x_test-mean_1)**2)/(2*var_1))/np.sqrt(2*np.pi*var_1)
        p_0 = np.exp(-((x_test-mean_0)**2)/(2*var_0))/np.sqrt(2*np.pi*var_0)

        print(p_1)
        print(p_0)
```

```
[0.00704649 0.22659658]
[0.62825235 0.25148582]
```

1.1 Estimating the label

```
In [7]: if np.prod(p_1)*(x_train_1.shape[0]/1000) > np.prod(p_0)*(x_train_0.shape[0]/1000):
            print("label = ",1)
        else:
            print("label = ",-1)
```

```
label =  -1
```

2 Observations

- Navie bayes classifier has been implemented
- The training set(each dimension) is assumed to be guassian

The predictions on the test data given is as follows: - for [1,-1] the estimated label is "1" - for [1,1] the estimated label is "1" - for [-1,-1] the estimated label is "-1" - for [-1,1] the estimated label is "-1"