02 MAYIS 2017


# ISTANBUL TECHNICAL UNIVERSITY

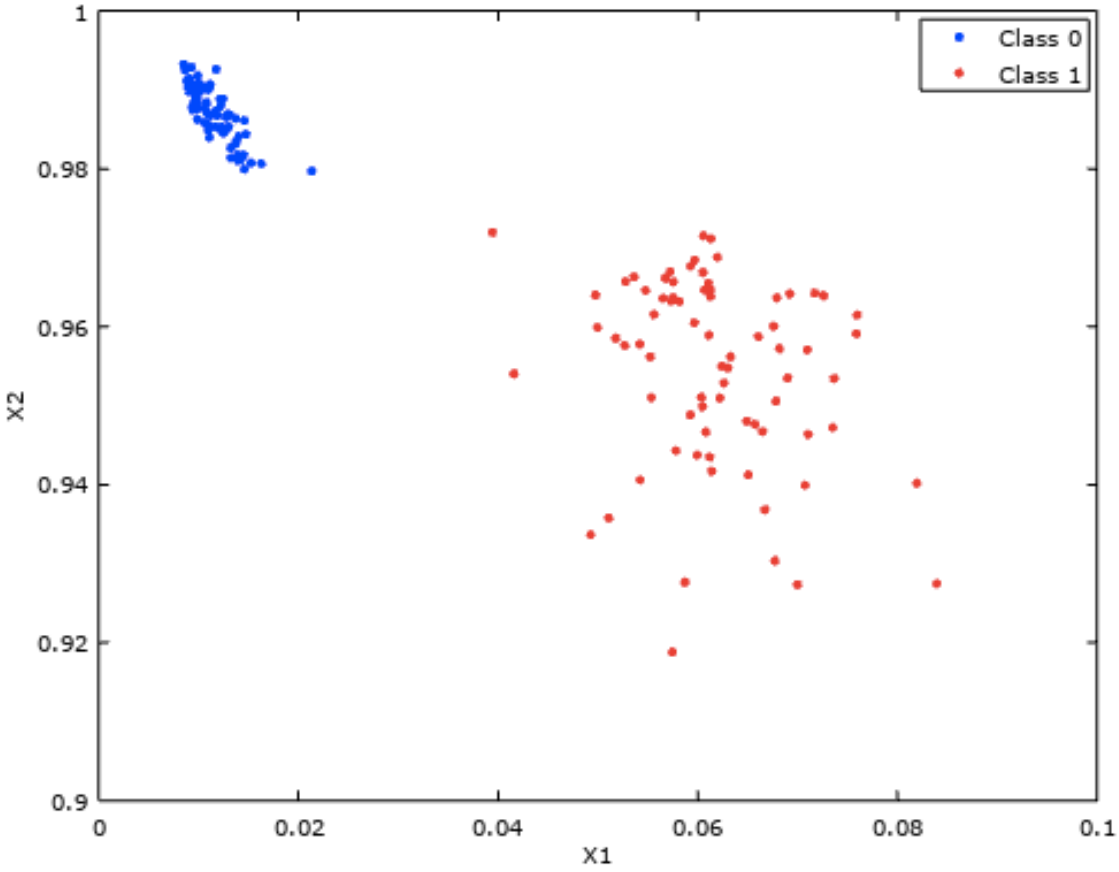
# BLG 454E LEARNING FROM DATA

HOMEWORK 2

TEAM: BEE MEISTER
EMRE ÖZDİL – 150120138
MERVE ECEVİT – 150140115

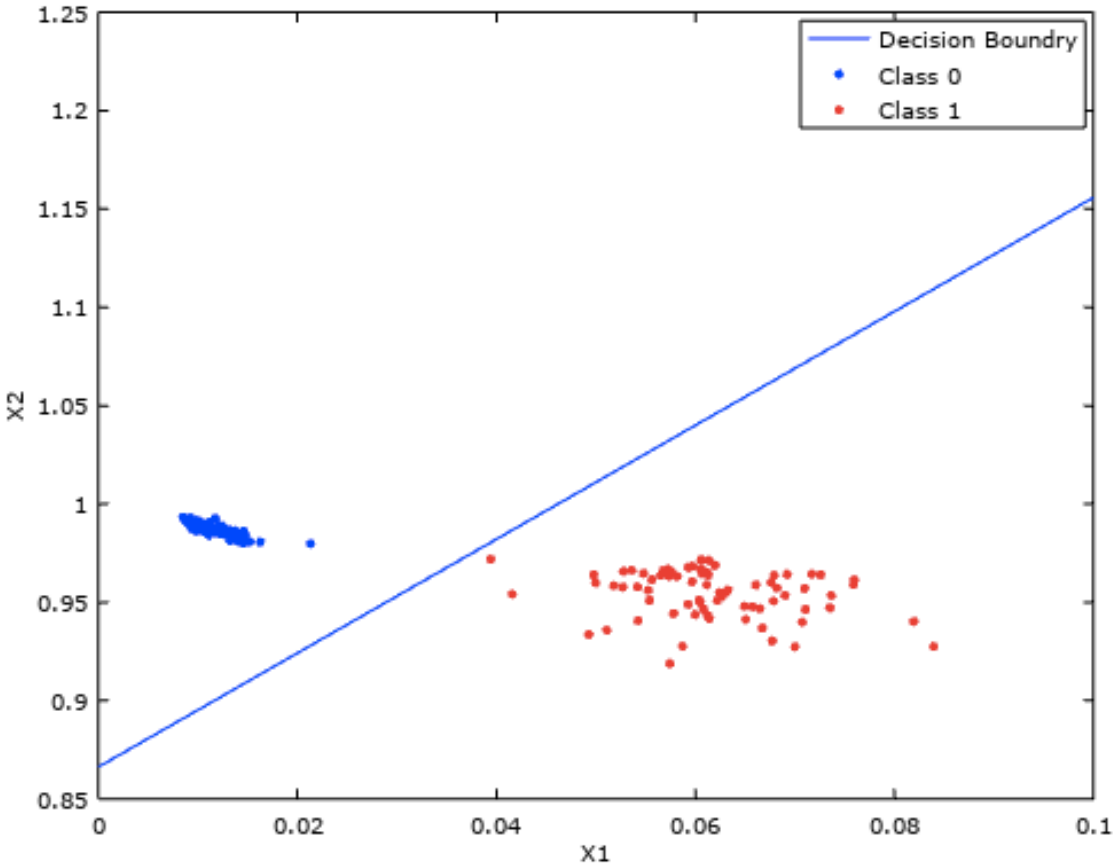
**1)** In order to classify the dataset, we divided data into two groups which are Training and Test Set. Training Set has 150 elements while Test Set has 30 elements. Using Training Set, Theta matrix is found. In order to determine the class of Test Set, Theta matrix is used. The result of the classification can be seen in below.

## Class 0 and Class 1 of Training Set without Decision Boundary



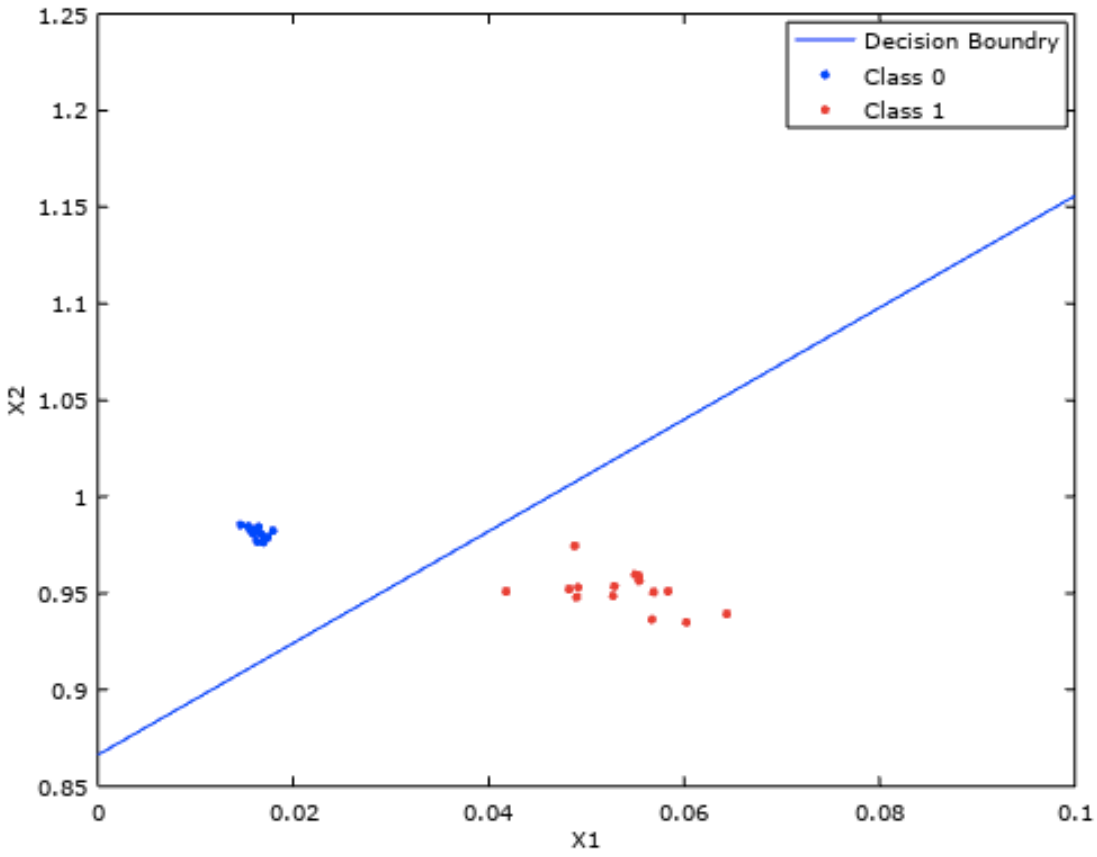

Training Set elements of Class 0 ⇒ blue points
Training Set elements of Class 1 ⇒ red points

## Class 0 and Class 1 of Training Set with Decision Boundary



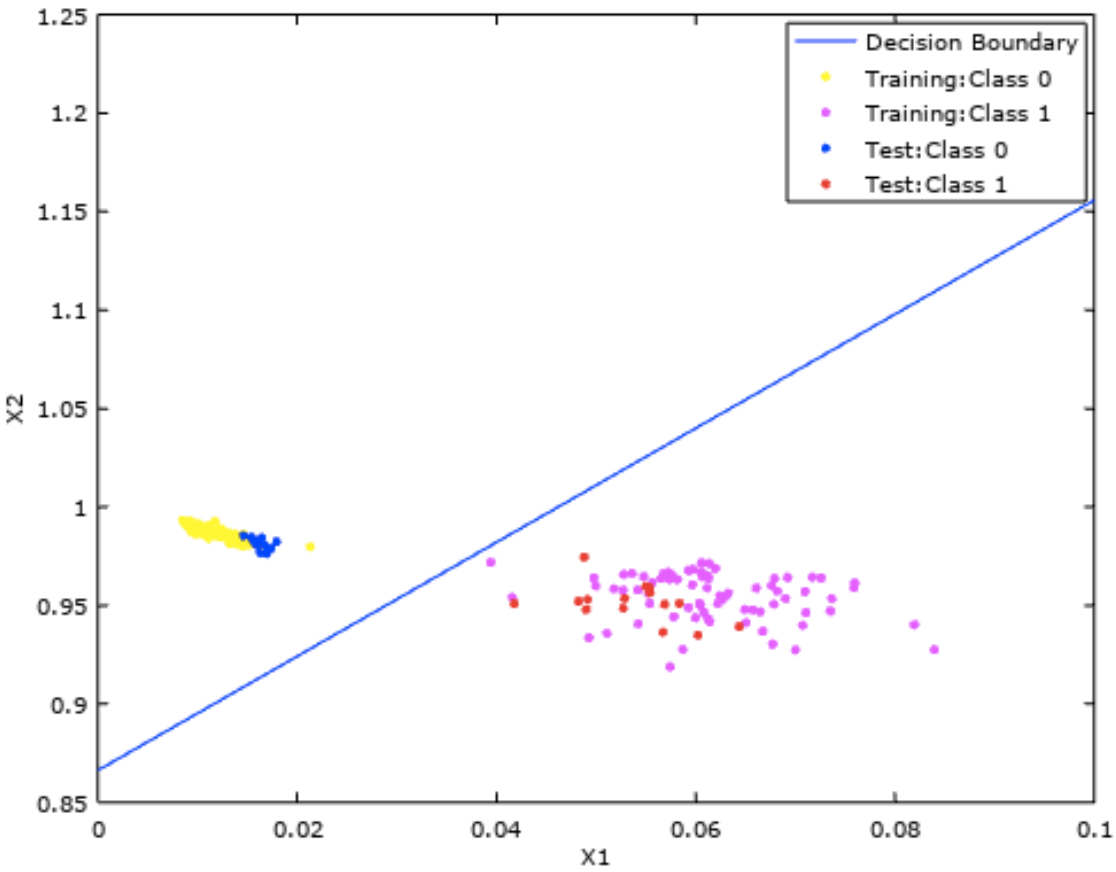

Training Set elements of Class 0 ⇒ blue points
Training Set elements of Class 1 ⇒ red points
Decision Boundary ⇒ blue line

## Class 0 and Class1 of Test Set with Decision Boundary




Test Set elements of Class 0 ⇒ blue points
Test Set elements of Class 1 ⇒ red points
Decision Boundary ⇒ blue line

## Class 0 and Class 1 of Training and Test Set with Decision Boundary




Training Set elements of Class 0 ⇒ yellow points
Training Set elements of Class 1 ⇒ magenta points
Test Set elements of Class 0 ⇒ blue points
Test Set elements of Class 1 ⇒ red points
Decision Boundary ⇒ blue line

## Accuracy of Logistic Regression

The classes of selected Test Set and the classes which are found using logistic regression is compared. Classes of all of the Test Set elements found properly. Therefore, accuracy of logistic regression is equals to %100. From the MATLAB code, method of finding the accuracy can be seen.

```
>> hw2

Accuracy of Logistic Regression = 100.000000
```

2) Gradient Descent

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \qquad \alpha \text{ is learning rate}$$

if $\alpha$ is low : not enough learning, error rate is too high and does not descend rapidly enough, we would have to wait too many times for good result.

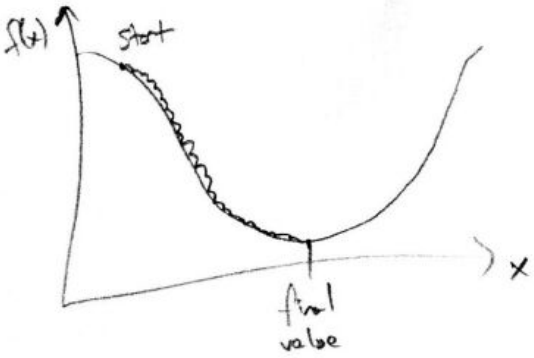
if $\alpha$ is high : performance will diverge. In particular, we will see the less number jump higher and higher until it becomes not a number

In brief

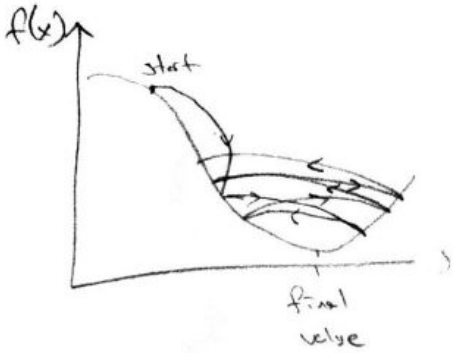
too low: slow convergence

to high: $J(\theta)$ may not decrease on every iteration, may not converge

too low




too high




Too choose $\alpha$

0,0001 , 0,0003 , 0,001 , 0,005 , 0,01 , 0,05 , 0,1

3x ≈3x 3x ≈3x 3x ≈3x

3) If the number of classes are grater than 3.

$C_K$ as the reference class and assume that

$$\log \frac{P(x/C_i)}{P(x/C_k)} = w_i^T x + w_i^0 \, 0$$

Then we have

$$\frac{P(C_i|x)}{P(C_k|x)} = \exp[w_i^T x + w_{i0}]$$

with $w_{i0} = w_{i0}^0 + \log P(C_i)/P(C_k)$.

We see that

$$\sum_{i=1}^{k-1} \frac{P(C_i|x)}{P(C_k|x)} = \frac{1 - P(C_k|x)}{P(C_k|x)} = \sum_{i=1}^{k-1} \exp[w_i^T x + w_{i0}]$$

$$\Rightarrow P(C_k|x) = \frac{1}{1 + \sum_{i=1}^{k-1} \exp[w_i^T x + w_{i0}]}$$

and also that

$$\frac{P(C_i|x)}{P(C_k|x)} = \exp[w_i^T x + w_{i0}]$$

$$\Rightarrow P(C_i|x) = \frac{\exp[w_i^T x + w_{i0}]}{1 + \sum_{j=1}^{k-1} \exp[w_j^T x + w_{j0}]}$$

$$i = 1, \dots, k-1$$

To treat all classes uniformly, we can write

$$y_i = \hat{P}(C_i|x) = \frac{\exp[w_i^T x + w_{i0}]}{\sum_{j=1}^{k} \exp[w_j^T x + w_{j0}]}, \quad i = 1, \dots, K$$

$$l(\{w_i, w_{i0}\}_i | x) = \prod_t \prod_i (y_i^t)^{r_i^t}$$

and the error function is again cross-entropy:

$$E(\{w_i, w_{i0}\}_i | X) = -\sum_t \sum_i r_i^t \log y_i^t$$

We again use gradient descent. If $y_i = \exp(a_i) / \sum_j \exp(a_j)$, we have

$$\frac{\partial y_i}{\partial a_j} = y_i (\delta_{ij} - y_j)$$

where $\delta_{ij}$ is the Kronecker delta, which is 1 if $i=j$ and 0 if $i \neq j$ (exercise 3). Given that $\sum_i r_i^t = 1$, we have the following update equations for $j = 1, \ldots, K$

$$\Delta w_j = \eta \sum_t \sum_i \frac{r_i^t}{y_i^t} y_i^t (\delta_{ij} - y_j^t) x^t$$

$$= \eta \sum_t \sum_i r_i^t (\delta_{ij} - y_j^t) x^t$$

$$= \eta \sum_t \left[ \sum_i r_i^t \delta_{ij} - y_j^t \sum_i r_i^t \right] x^t$$

$$= \eta \sum_t (r_j^t - y_j^t) x^t$$

$$\Delta w_{j0} = \eta \sum_t (r_j^t - y_j^t)$$