Machine learning Demo for **Logistic Regression** using **Scilab**

Steps to be followed:

1. Create some random data

in form of “high =1 “ and “low=0, in a .csv file named “random.csv”
This data will be used for training of model for logistic regression.

Code:

```
b0 = 5;
t = b0 * rand(100,2);
t = [t 0.5+0.5*sign(t(:,2)+t(:,1)-b0)];

b = 1;
flip = find(abs(t(:,2)+t(:,1)-b0)<b);
t(flip,$)=grand(length(t(flip,$)),1,"uin",0,1);
```

2. Import the data

The data present in random.csv files has total of 100 samples divided into two classes 1s and 0s on the basis of the parameters stored in 1st and 2nd column of the .csv file.

Code:

```
t = csvRead(“random.csv”);
```

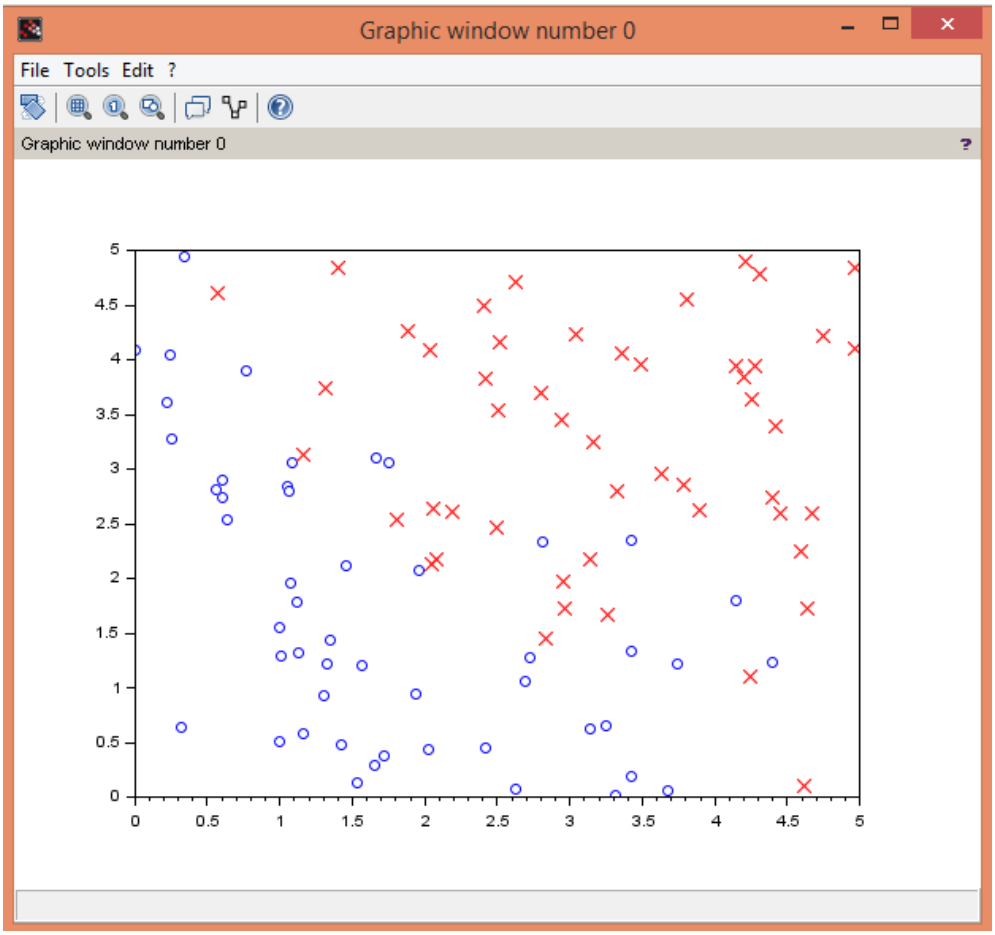
3. Split the data into two classes

```
c0 = t(find(t(:,$)==0),:);
c1 = t(find(t(:,$)==1),:);
```

4. Plot or represent the random data generated

```
clf(0); scf(0);
plot(c1(:,1),c1(:,2),'rx')
plot(c0(:,1),c0(:,2),'bo')
```


Mandar Deshpande mandar061095@gmail.com



5. Building a classification model

Our model should predict the particular input data that belongs to class 1.

Separate data into features and results

```
x = t(:, 1:$-1); y = t(:, $);
[m, n] = size(x);
```

Add and extra column to feature vector x to accommodate the intercept term

```
x = [ones(m, 1) x]
```

Hypothesis function for logistic regression is defined as

$$h(\theta, x) = 1 / (1 + exp(-\theta^T x))$$

It's magnitude is the probability that the data with the features x belong to the class 1.

The Cost Function in logistic regression is

$$J = [-y^T log(h) - (1-y)^T log(1-h)]/m$$

Where log is the "element-wise" logarithm, not a matrix logarithm.

Mandar Deshpande mandar061095@gmail.com

6. Gradient Descent

Update rule for θ using gradient descent algorithm is

$$\theta \rightarrow \theta - \alpha \nabla J = \theta - \alpha x^T (h - y) / m$$

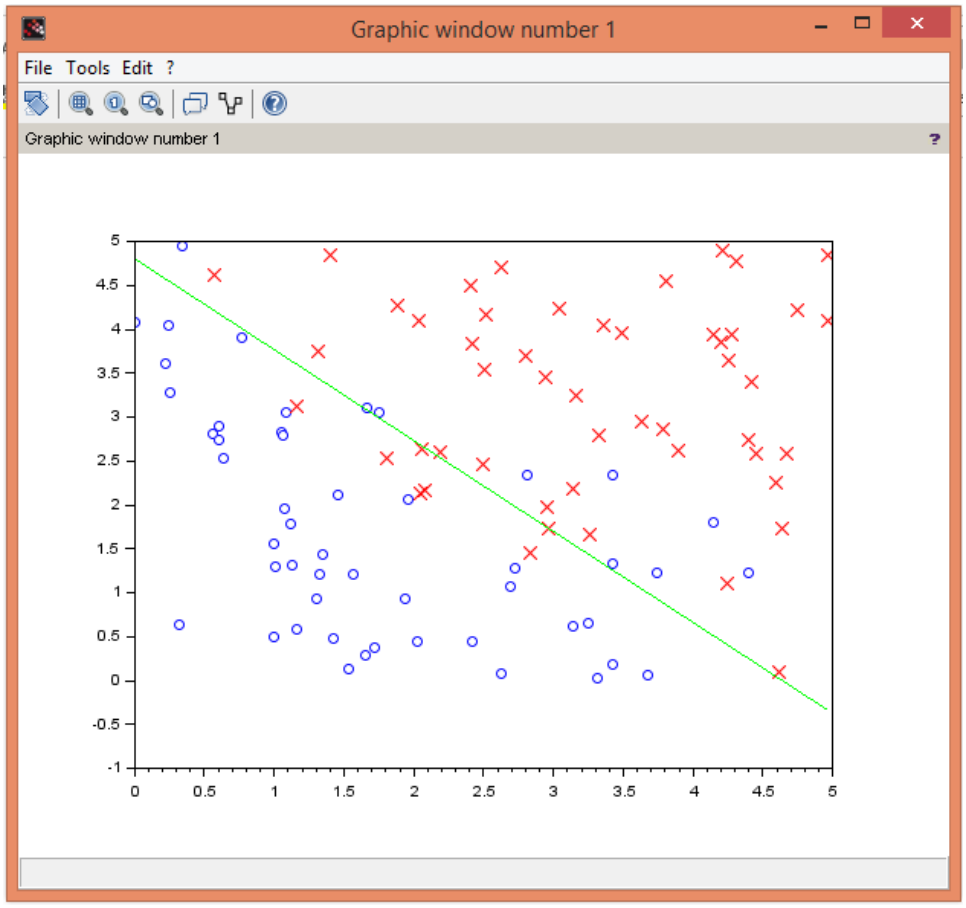
Code:

```
//initialize fitting parameters
theta = zeros(n + 1, 1);

// Learning rate ‘a’ and number of iterations ‘n_iter’
a = 0.01;
n_iter = 10000;

for iter = 1:n_iter do
    z = x * theta;
    h = ones(z) ./ (1+exp(-z));
    theta = theta - a * x' *(h-y) / m;
    J(iter) = (-y' * log(h) - (1-y)' * log(1-h))/m;
end
```

7. Visualize the output

```
// Display the result
disp(theta)
u = linspace(min(x(:,2)),max(x(:,2)));

clf(1);scf(1);
plot(c0(:,1),c0(:,2),'bo')
plot(c1(:,1),c1(:,2),'rx')
plot(u,-(theta(1)+theta(2)*u)/theta(3),'-g')
```

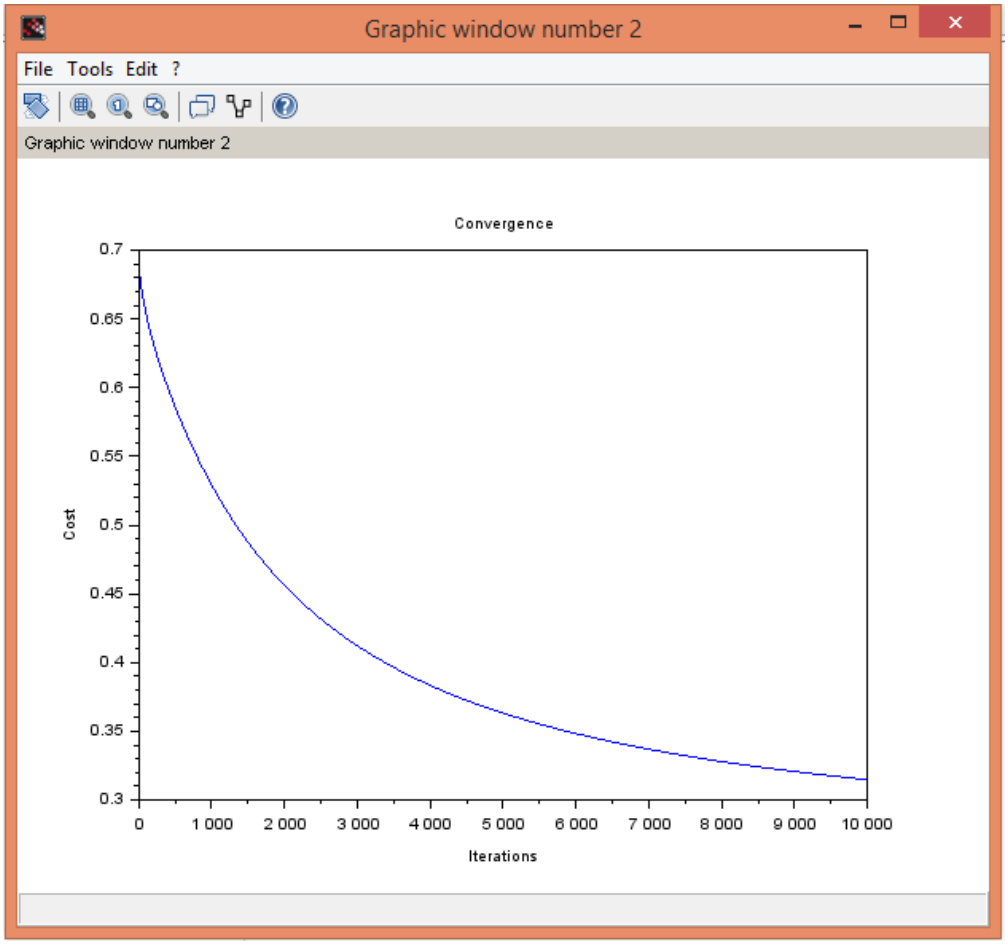

Mandar Deshpande mandar061095@gmail.com



8. Visualize the cost function for convergence of the model

```
clf(2);scf(2);
plot(1:n_iter, J');
xtitle('Convergence','Iterations','Cost')
```


Mandar Deshpande mandar061095@gmail.com



References:

http://scilab.io/machine-learning-logistic-regression-tutorial/

http://www.holehouse.org/mlclass/06_Logistic_Regression.html