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

Steps to be followed:

1. Create some random data

having two features, in a .csv file named “random_linear.csv”
This data will be used for training of model for linear regression.

2. Import the data

The data present in random.csv files has total of 97 samples divided into input ‘x’ and output ‘y’ on the basis of the parameters stored in 1st and 2nd column of the .csv file.

Code:

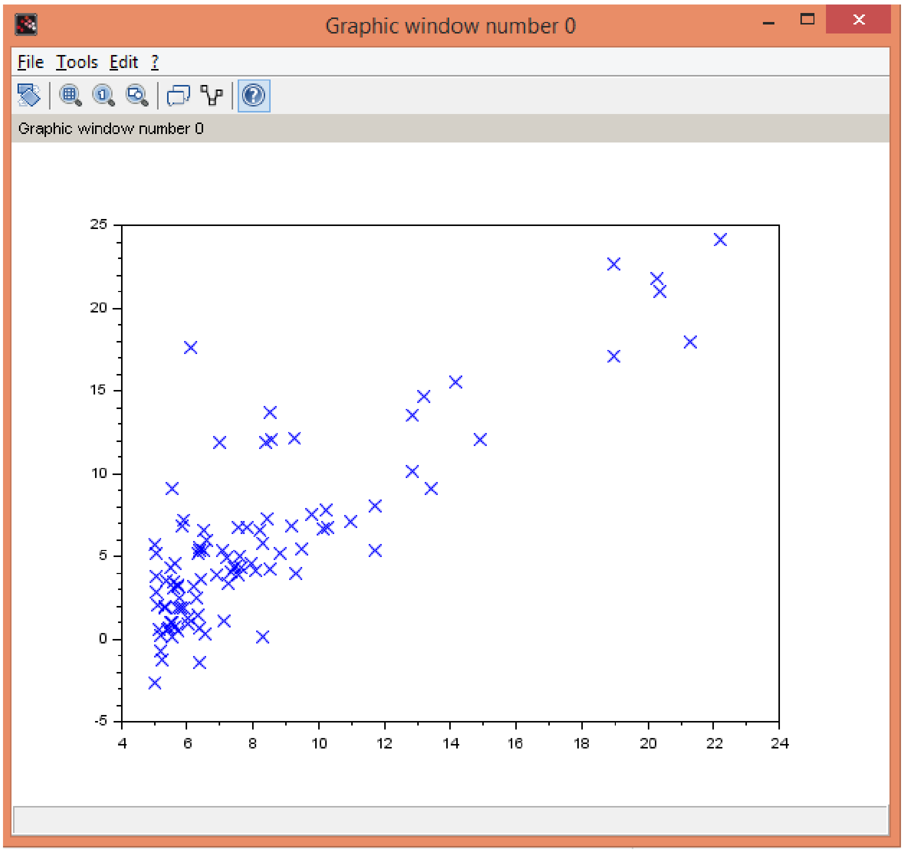
```
t = csvRead(“random_linear.csv”);
```

3. Split the data into t1 and t2 for plots

```
t1 = t(:,1);
t2 = t(:,2);
```

4. Plot or represent the random data generated

```
clf(0); scf(0);
plot(t1,t2,'bx');
```



5. Building a classification model

Our model should figure out how to fit the best straight line to our data

Separate data into features and results

```
x = t1; y = t2;
[m, n] = size(x);
```

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

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

Hypothesis function for logistic regression is defined as

$$h\theta(x) = \theta T \, x = \theta 0 + \theta 1x1$$

It’s magnitude is the probability that the data with the features x lies on the line $h\theta(x)$

The Cost Function in logistic regression is

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Gradient Descent

Update rule for θ using gradient descent algorithm is

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \qquad \text{(simultaneously update } \theta_j \text{ for all } j\text{).}$$

Code:

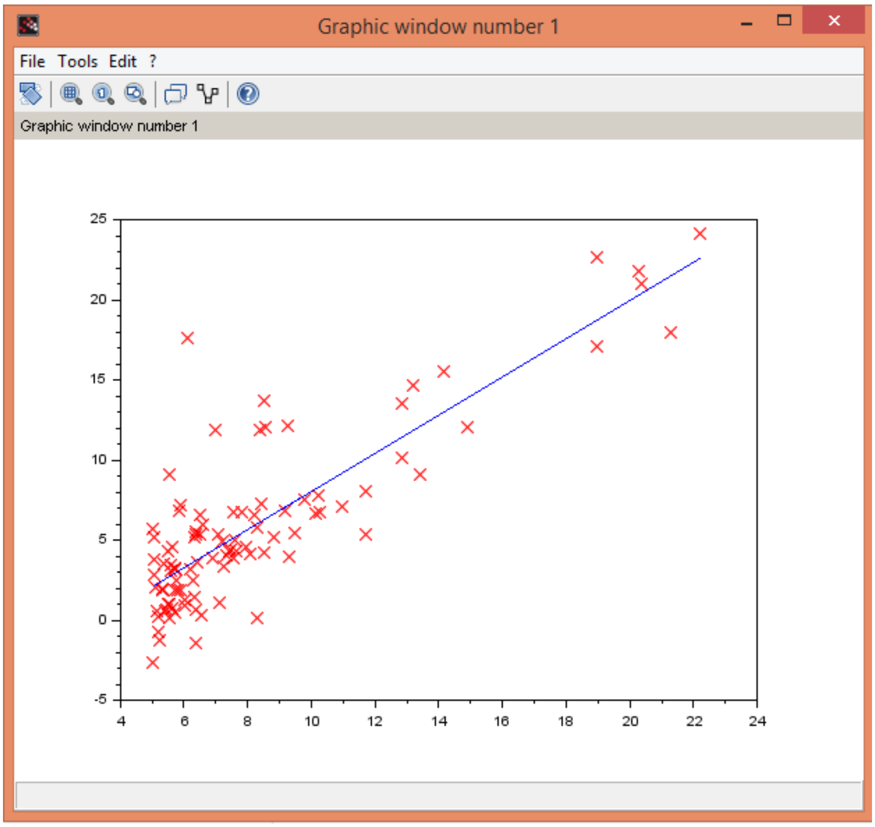
```
//initialize fitting parameters
theta = zeros(n + 1, 1);

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

for iter = 1:n_iter do
    theta = theta - a * (x' *(x*theta-y)) / m;
    J(iter) = 1/(2*m) * sum((x*theta -y).^2)
end
```

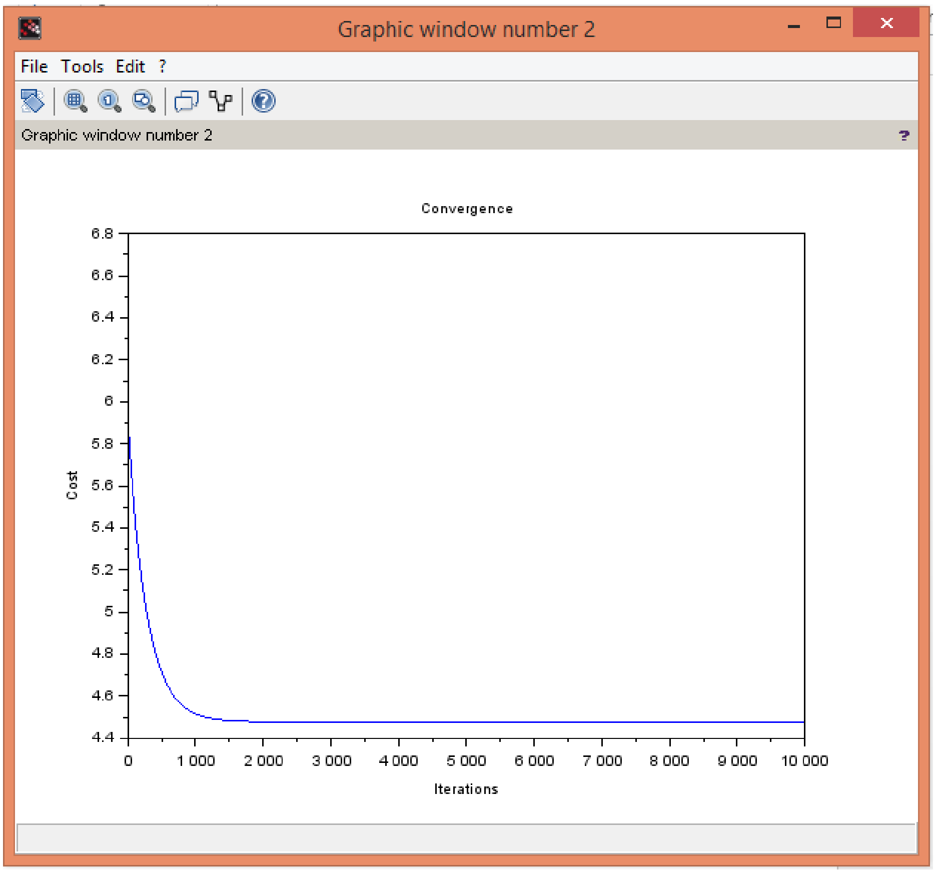
6. Visualize the output

```
// Display the result
disp(theta)
clf(1);scf(1);
plot(t1,t2,'rx');
plot(x(:,2), x*theta, '-')
```



7. Visualize the cost function for convergence of the model

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



References:

http://www.holehouse.org/mlclass/04_Linear_Regression_with_multiple_variables.html