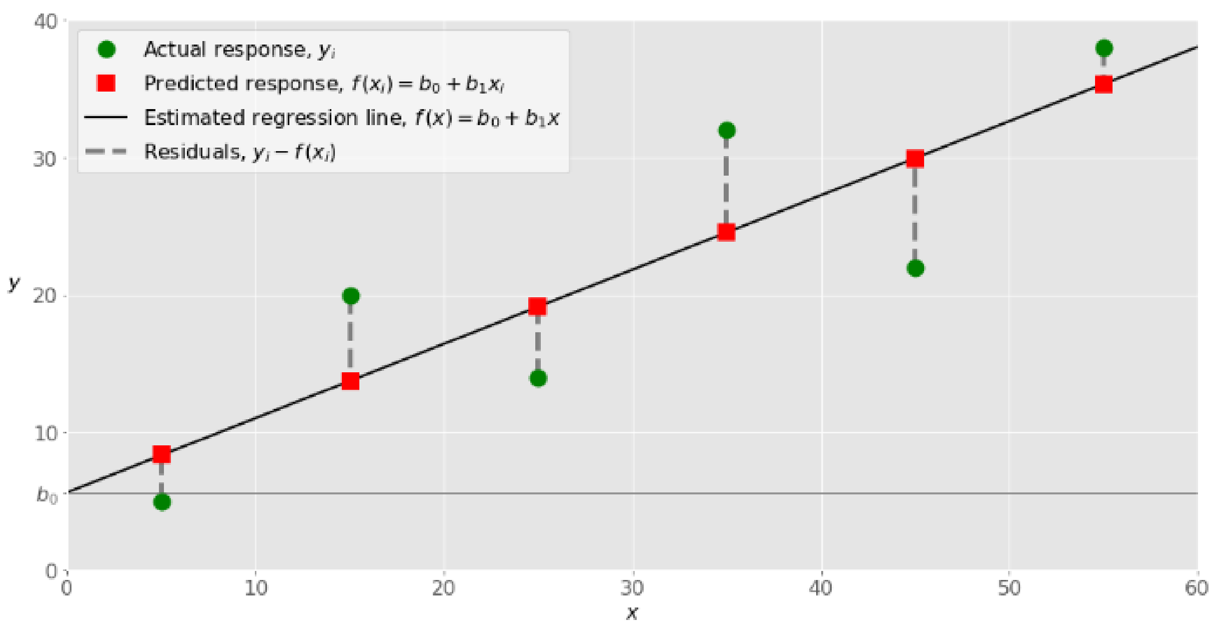
AI Lab 12
Section A

Introduction

Linear Regression is a machine learning algorithm based on **supervised learning**. It performs a **regression task**. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables, they are considering and the number of independent variables being used. Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output). Hence, the name is Linear Regression.

Simple Linear Regression

Simple or single-variate linear regression is the simplest case of linear regression with a single independent variable, $\mathbf{x} = x$.
The following figure illustrates simple linear regression:



The value of b_0, also called the **intercept**, shows the point where the estimated regression line crosses the y axis. It is the value of the estimated response (x) for $x = 0$. The value of b_1 determines the **slope** of the estimated regression line.

Multiple Linear Regression:

Multiple or multivariate linear regression is a case of linear regression with two or more independent variables. If there are just two independent variables, the estimated regression function is $(x_1, x_2) = b_0 + b_1x_1 + b_2x_2$. It represents a regression plane in a three-dimensional space.

Linear Regression Model

1) **Prediction model (Based on input and some model parameter)**

$$f(x) = \beta_0 + \sum_{j=1}^{d} \beta_j x_j$$

$\beta's$ Are called parameter of coefficients or weights.

Learning the linear model Learning the $\beta's$

For one feature (d=1)

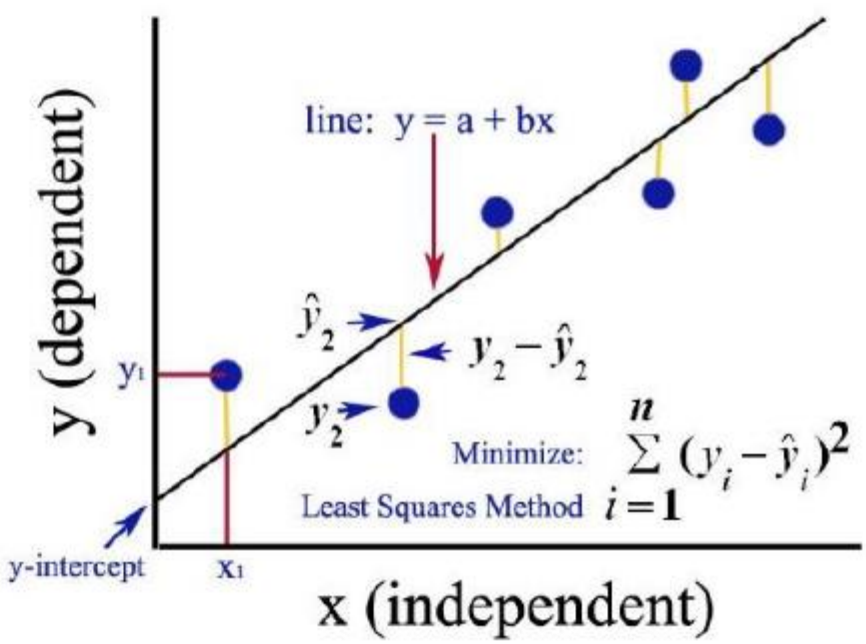
$f(x) = \beta_0 + \beta_1 x$

2) **Cost /Risk function**

Use least square loss: $loss(y_i, f(x_i)) = (y_i, f(x_i))^2$

We want to minimize the loss over all examples, that is minimize the cost or risk function.

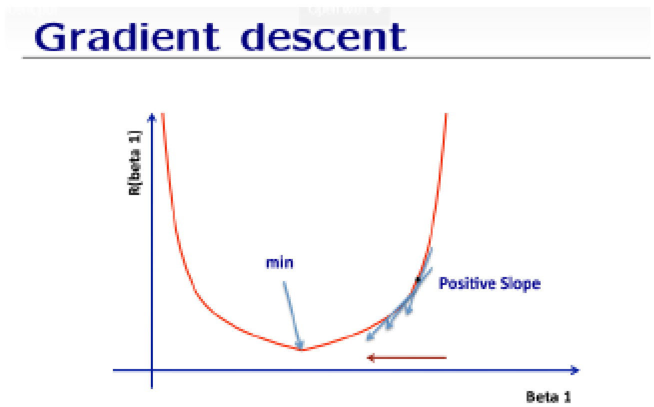
$$R = \frac{1}{2n} \sum_{i=1}^{n} (y_i - f(x_i))^2 \qquad R(\beta) = \frac{1}{2n} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$$



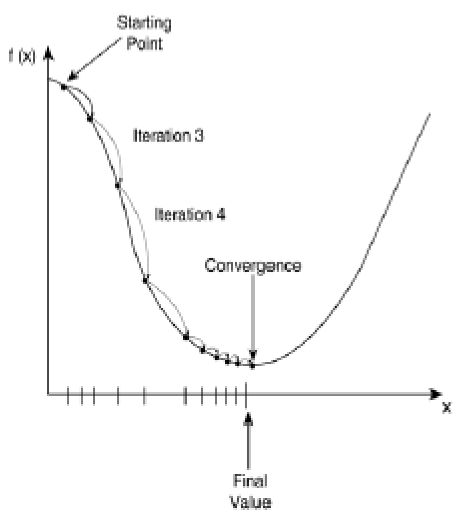
Cost/Risk Function

3) Calculate model parameter while decreasing the cost/risk

We will find the β_0 and β_1 to minimize the risk. And it can be done by using gradient descent which is an optimized method.



Gradiant Descent



Show Convergence

Repeat until convergence.

Update simultaneously all β_j's for (j=0 and j=1)

$$\beta_0 = \beta_0 - \propto \frac{\partial}{\partial \beta_0} R(\beta_0, \beta_1)$$

$$\beta_1 = \beta_1 - \propto \frac{\partial}{\partial \beta_1} R(\beta_0, \beta_1)$$

In the linear case:

$$\frac{\partial R}{\partial \beta_0} = \frac{1}{n}\sum_{i=1}^{n} \quad (y_i - \beta_0 - \beta_1 x_i) \times (-1)$$

$$\frac{\partial R}{\partial \beta_1} = \frac{1}{n}\sum_{i=1}^{n} \quad (y_i - \beta_0 - \beta_1 x_i) \times (-x_i)$$

Let's generalize it!

$$\beta_0 = \beta_0 - \propto \frac{1}{n}\sum_{i=1}^{n} \quad (y_i - \beta_0 - \beta_1 x_i) \times (-1)$$

$$\beta_1 = \beta_1 - \propto \frac{1}{n}\sum_{i=1}^{n} \quad (y_i - \beta_0 - \beta_1 x_i) \times (-x_i)$$

4) **Identification of hyper parameters**

" $\propto$ "is a learning rate and we can call it hyper parameter. We can also learn this by using algorithms for example: ADAM but in our lab task we can assume any value of it from 0 to 1.

There are five basic steps when you are implementing linear regression:

1. Import the packages and classes you need.
2. Provide data to work with and eventually do appropriate transformations.
3. Create a regression model and fit it with existing data.
4. Check the results of model fitting to know whether the model is satisfactory.
5. Apply the model for predictions.

These steps are more or less general for most of the regression approaches and implementations.

Your task is to classify the following dataset using linear regression. You have to implement linear regression model function step by step and cannot use any libraries like sklearn for model.

The link of dataset is given below: https://archive.ics.uci.edu/dataset/53/iris

- Do pre-processing on the dataset (remove null and dupicate values)
- Show scatter plot of the dataset.
- The splitting of train-test data will be 75% - 25%.
- Fit the model on training dataset
- Perform binary classification on the test dataset
- Find mean square error.