Lecture 2
Supervised Learning and Linear Regression

Lê Anh Cường

Faculty of Information Technology

TDTU, 2019

Recap lecture 1

- What is Machine Learning?
- General architecture of ML based system.
- What is a ML Model?
- ML types?
- An example of ML: Naïve Bayesian classification
- Exercise: NB classification for text spam classification, …

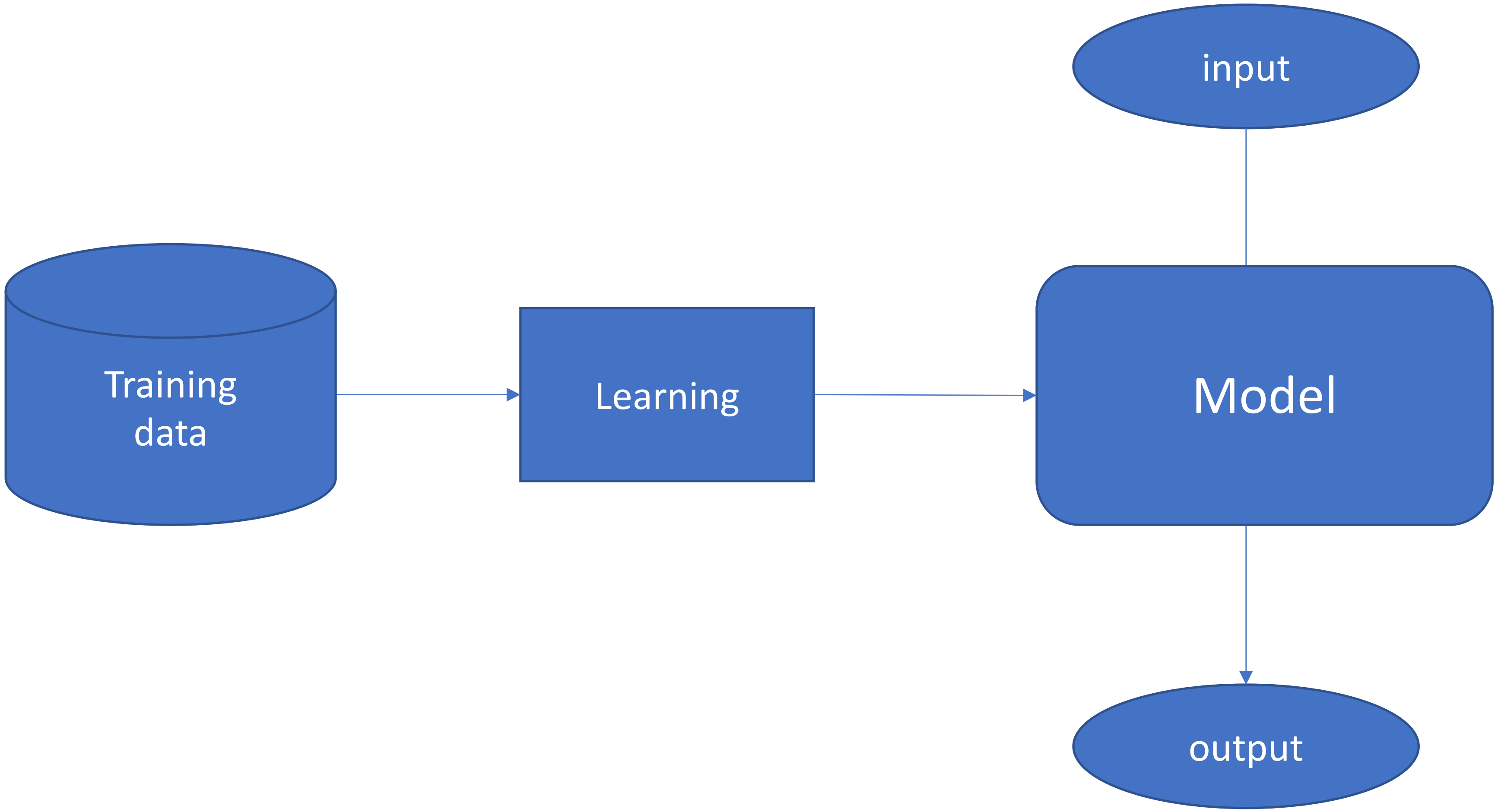
Outline

- Supervised learning problems?
- Linear regression problem and formulation?
- Model and learning model?
- Practice and Exercises

Supervised Learning problems?

- Classification
  - Single label
  - Multi-labels
- Generation
  - Labels
  - Real Values
- Sequence generation
  - Sequence of real values
  - Sequence of labels
- Structure Generation

Supervised Learning: general model?



General Mechanism of Supervised Learning

- Choose a model
    - Determine parameters of model
- Learning parameters of the model
- Inference on the model

Repeat: Naïve Bayes classification

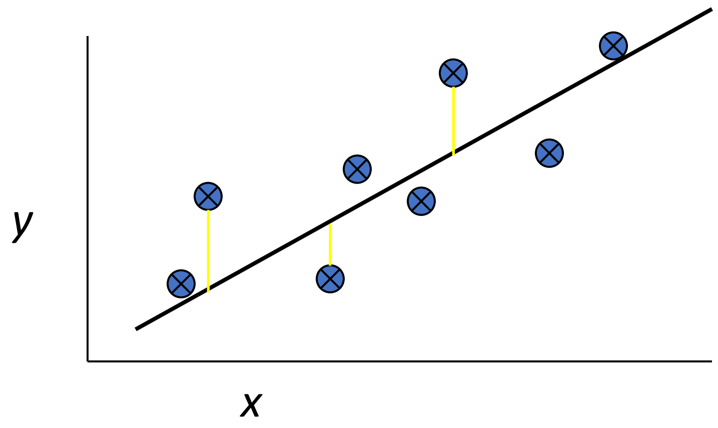
- Problem
- Model?
- Parameters of the model?

Regression vs Classification

- Distinguish between Regression and Classification
- Give examples?

Regression

- For classification the output(s) is nominal
- In regression the output is continuous
  - Function Approximation
- Many models could be used – Simplest is linear regression
  - Fit data with the best hyper-plane which "goes through" the points
  - For each point the differences between the predicted point and the actual observation is the *residue*



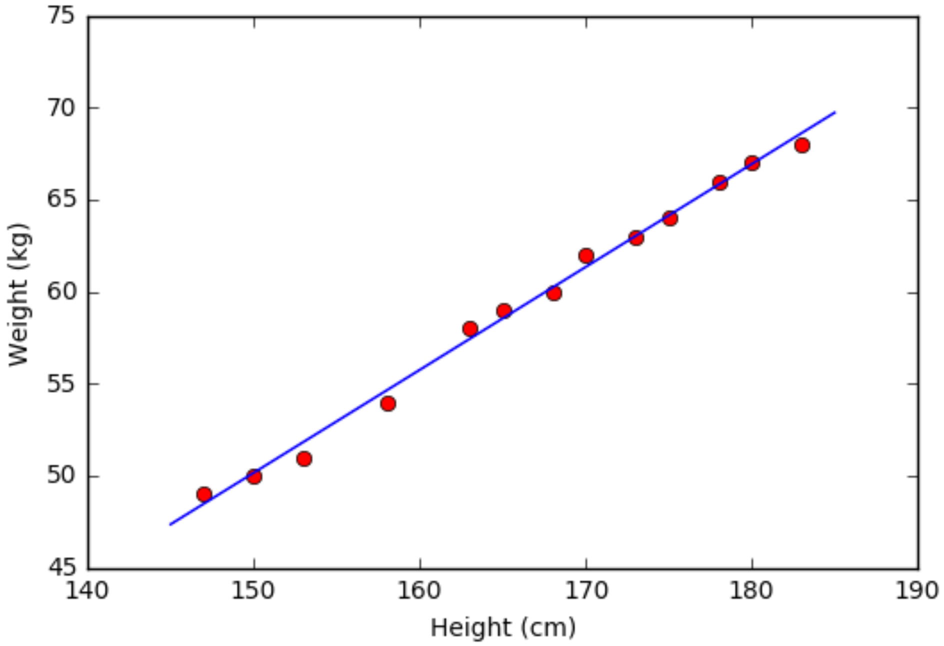
Regression (Hồi qui)

- From Height, predict Weight?

Height(cm)	Weight(kg)	Height(cm)	Weight(kg)
147	49	168	60
150	50	170	72
153	51	173	63
155	52	175	64
158	54	178	66
160	56	180	67
163	58	183	68
165	59		

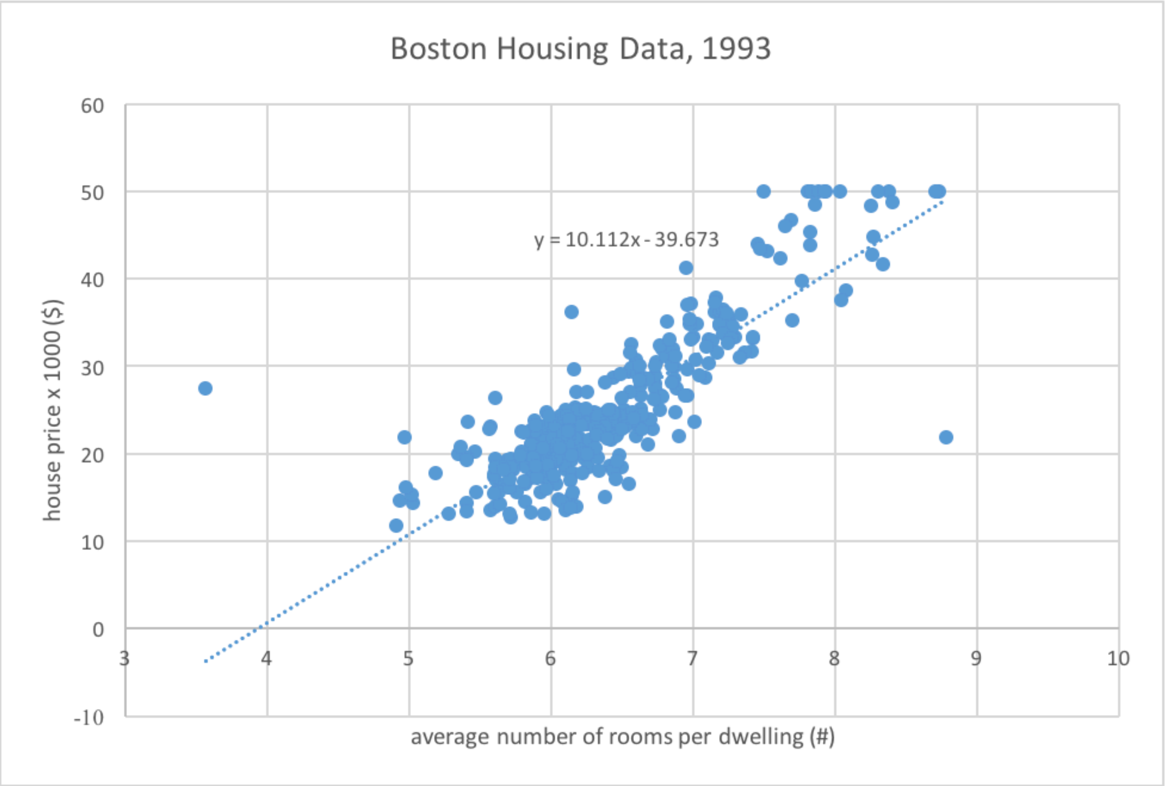
Linear Regression

- How is the relationship between Height and Weight?
- Suppose that Weight linearly depends on Height



Linear Regression

- Example of house pricing



Learning model

- Choose model form?
- Learning model’s parameters
- Loss function
- Learning: minimize the loss function

Linear Regression: Formulation

Given a data set $\{y_i,\, x_{i1},\ldots,x_{ip}\}_{i=1}^{n}$ of *n* statistical units, a linear regression model assumes that the relationship between the dependent variable *y* and the *p*-vector of regressors **x** is linear. This relationship is modeled through a *disturbance term* or *error variable ε* — an unobserved random variable that adds "noise" to the linear relationship between the dependent variable and regressors. Thus the model takes the form

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^\mathsf{T}\boldsymbol\beta + \varepsilon_i, \qquad i = 1,\ldots,n,$$

where $^\mathsf{T}$ denotes the transpose, so that $\mathbf{x}_i^\mathsf{T}\boldsymbol\beta$ is the inner product between vectors $\mathbf{x}_i$ and $\boldsymbol\beta$.

Often these *n* equations are stacked together and written in matrix notation as

$$\mathbf{y} = X\boldsymbol\beta + \boldsymbol\varepsilon,$$

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

$$X = \begin{pmatrix} \mathbf{x}_1^\mathsf{T} \\ \mathbf{x}_2^\mathsf{T} \\ \vdots \\ \mathbf{x}_n^\mathsf{T} \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix},$$

$$\boldsymbol\beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix}, \quad \boldsymbol\varepsilon = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}.$$

Simple Linear Regression

$$y = f(\mathrm{x}) = \ w_0 + \ w_1 x$$

$$\mathrm{Loss} = L = \frac{1}{2N}\sum_{i=1}^{N}(f(x_i) - y_i)^2 = \frac{1}{2N}\sum_{i=1}^{N}((w_0 + \ w_1 x) - y_i)^2 =$$

Simple Linear Regression

$$y = f(\mathrm{x}) = \ w_0 + \ w_1 x$$

$$\mathrm{Loss} = L = \frac{1}{2N}\sum_{i=1}^{N}(f(x_i) - y_i)^2 = \frac{1}{2N}\sum_{i=1}^{N}((w_0 + \ w_1 x) - y_i)^2 =$$

Parameters: w_0, w_1

Goal: minimize Loss function

$$\frac{\partial L}{\partial w_0} = 0, \qquad \frac{\partial L}{\partial w_1} = 0$$

Linear Regression: Learning

Goal: minimize Loss function

$$\frac{\partial L}{\partial w_0} = 0, \qquad \frac{\partial L}{\partial w_1} = 0$$

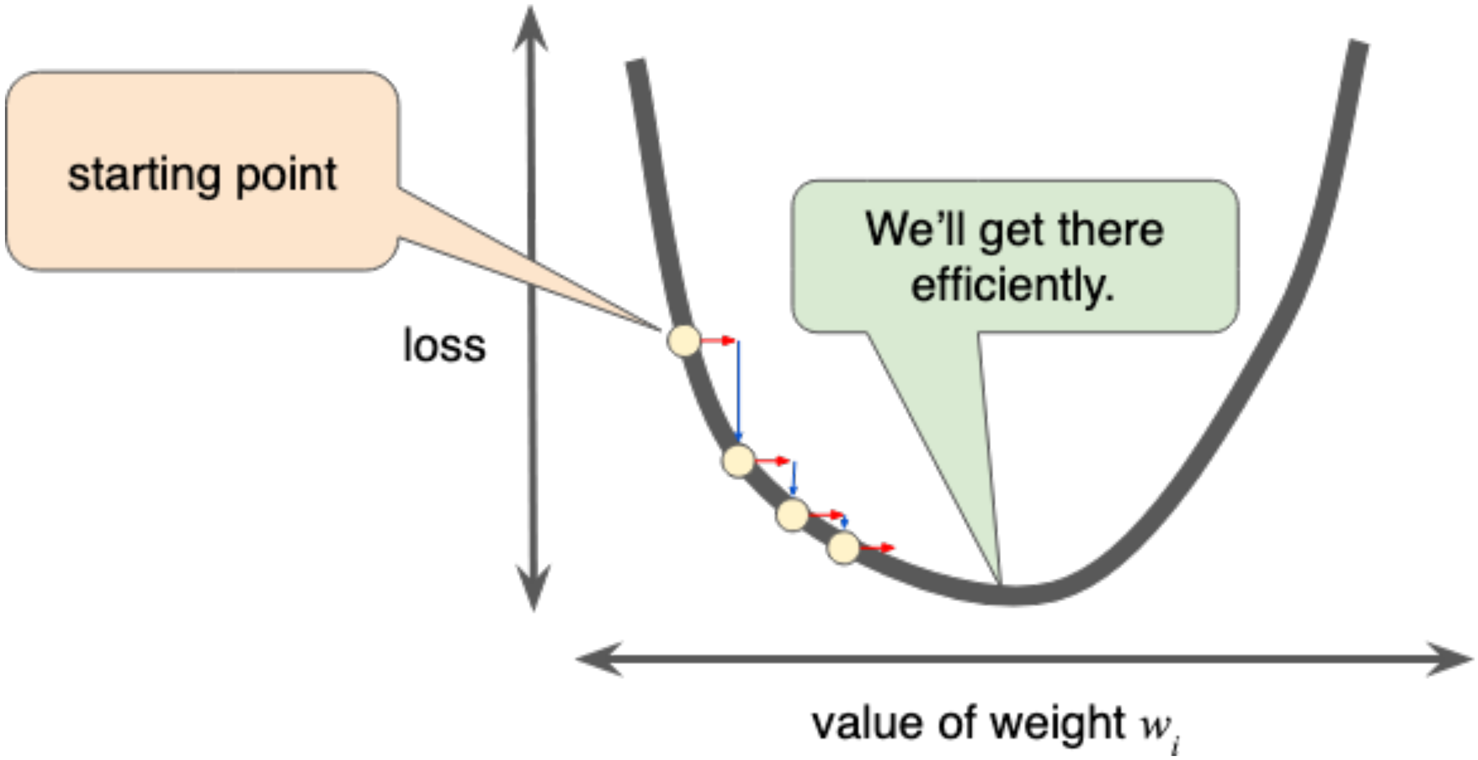
$$L = \frac{1}{2N}\sum_{i=1}^{N}(f(x_i) - y_i)^2 =$$

$$= \frac{1}{2N}\sum_{i=1}^{N}((w_0 + w_1 x) - y_i)^2$$

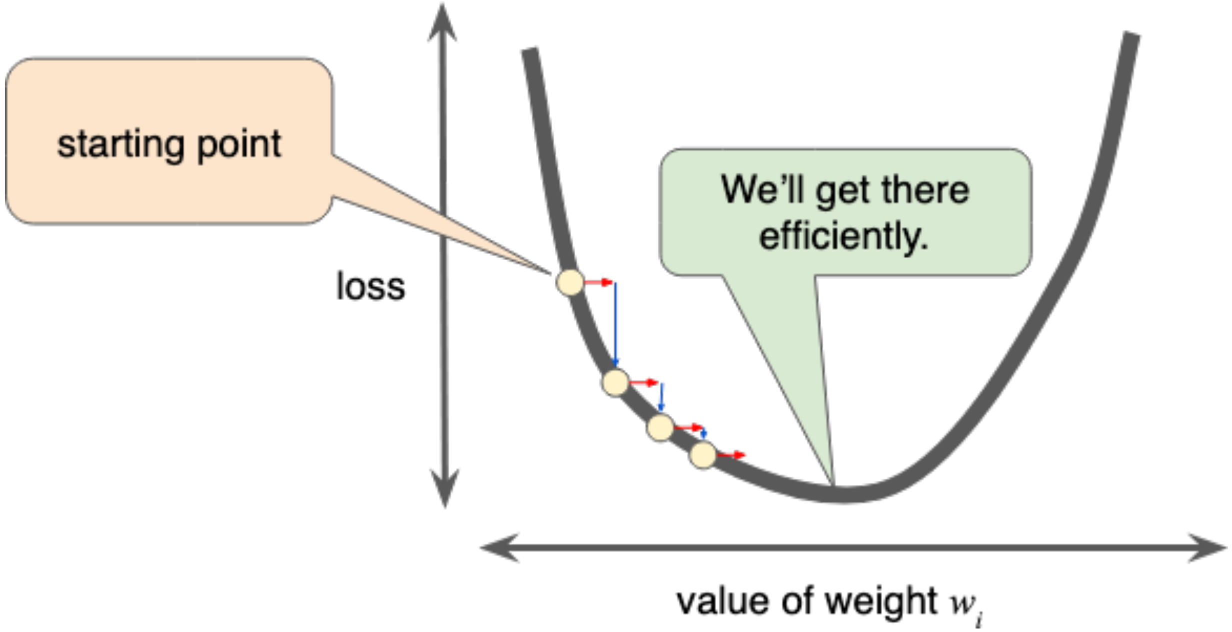
$$\frac{\partial L}{\partial w_0} = \frac{1}{N}\sum_{i=1}^{N}(f(x_i) - y_i)$$

$$\frac{\partial L}{\partial w_1} = \frac{1}{N}\sum_{i=1}^{N}(f(x_i) - y_i)\, x_i$$

Linear Regression: Learning by Gradient Descent



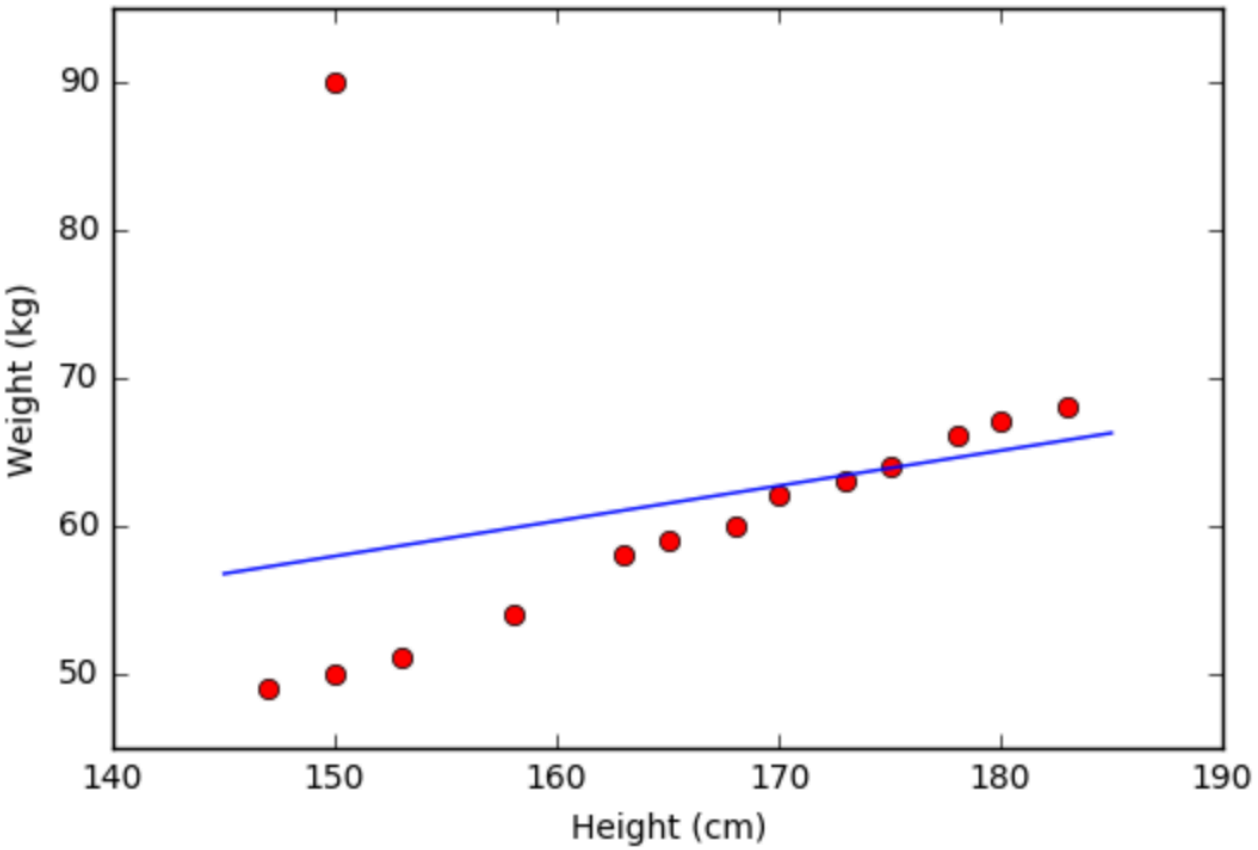
Linear Regression: Learning by Gradient Descent



Learning rate

$$w_i = w_i - \mu \frac{\partial L}{\partial w_i}$$

Linear Regression is sensitive with outliers



Summary

- What is Regression learning?
- When use Linear Regression?
- Limitation of Linear Regression?

Exercise

- Implement Linear Regression for the problem of house pricing with multiple variables:
  - Firstly doing in theory aspect
  - Secondly, python implementation