Machine Learning Course

Vahid Reza Khazaie

Linear Algebra, Calculus, Probability

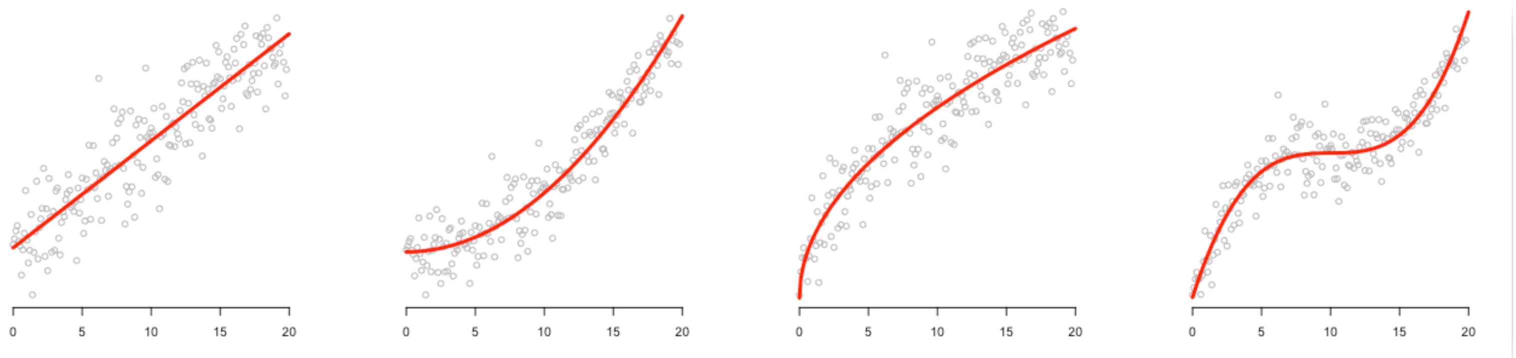
- Linear Algebra
  - Vector, Matrix
  - Addition, Multiplication, …
  - Inverse Matrix
- Calculus
  - Derivative
  - Gradient
- Probability
  - Basic Concepts
  - Conditional Probability

Supervised Learning

- Given the “right answer” for each example in the data.

Regression Problem

- Predict real-valued output

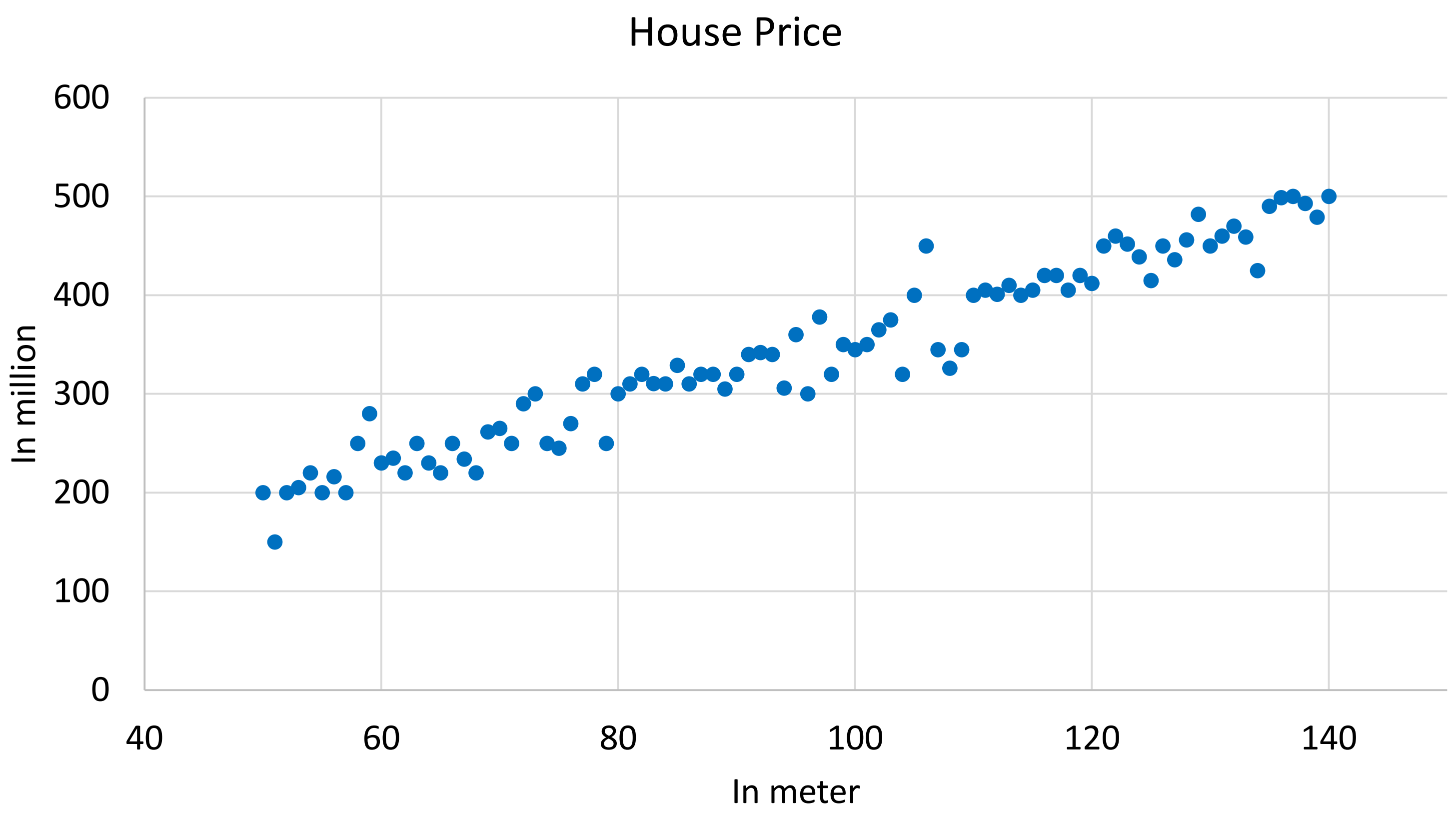


Linear Regression

- Linear Regression:

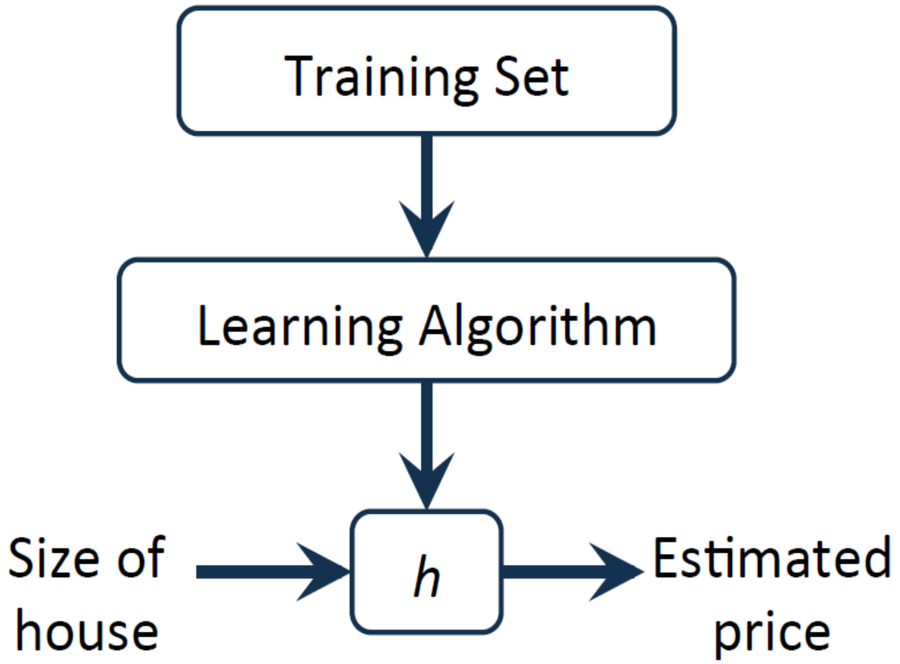
  In statistics, linear regression is a linear approach to modeling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regression. Wikipedia

Linear Regression



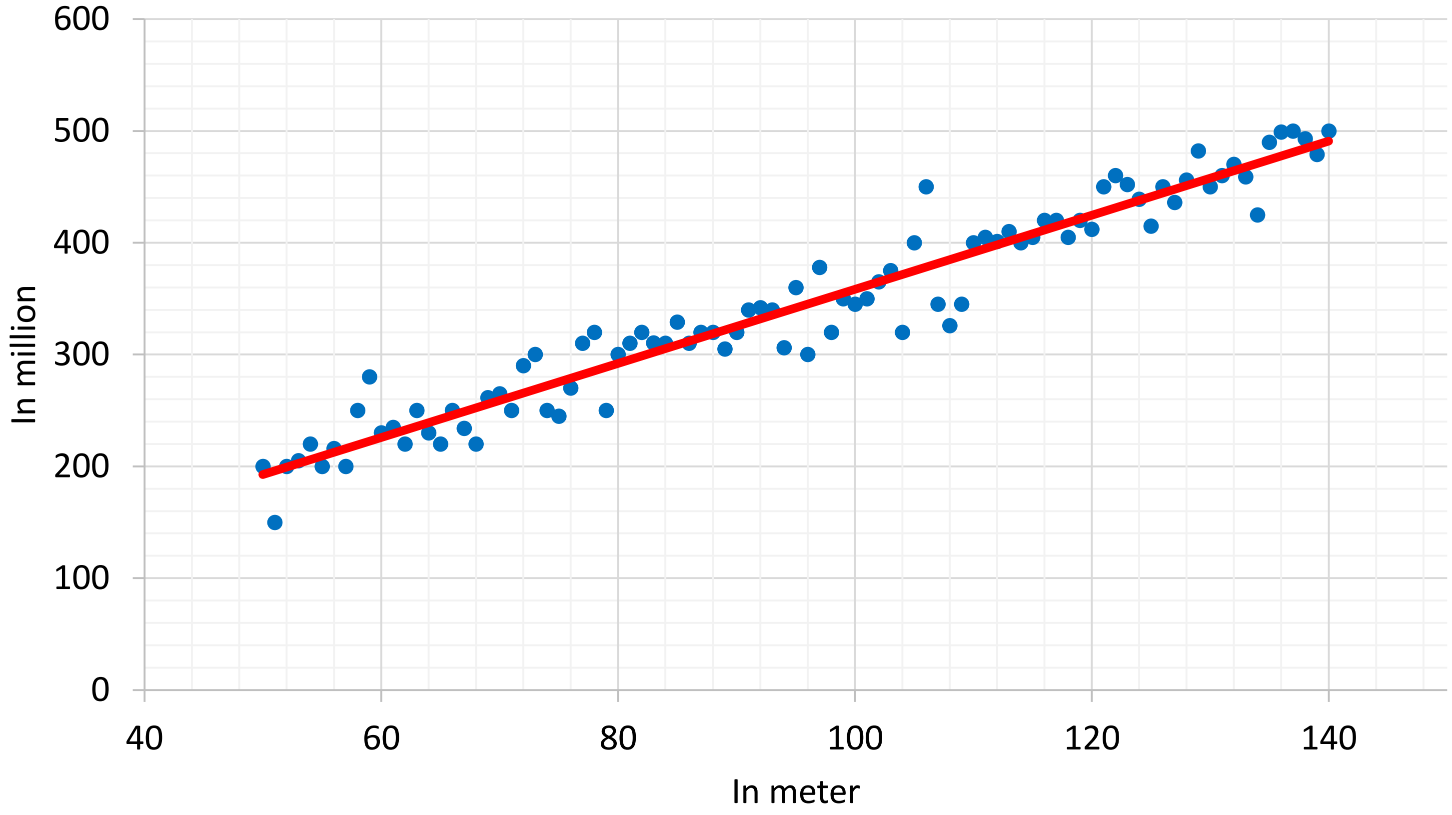
Linear Regression

Size of house (x)	House Price (y)
50	200
51	150
52	200
53	205
54	220
55	200
…	…



$$x \xrightarrow{h_{(x)}} y \quad h_{(x)} = ?$$

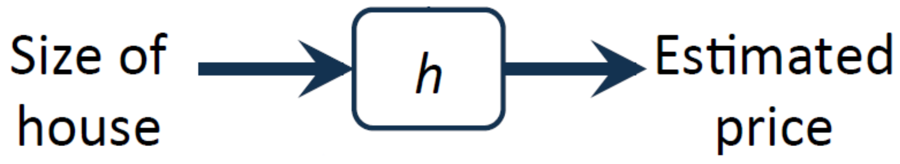
Linear Regression



Linear Regression

- How do we represent h(x)?

$$\mathbf{h_{(x)} = \theta_1 x + \theta_0}$$



Linear regression

- **with one variable x (Univariate linear regression)**
- **with multiple variables x_0, x_1, x_2 ,... (Multivariate linear regression)**

Linear Regression

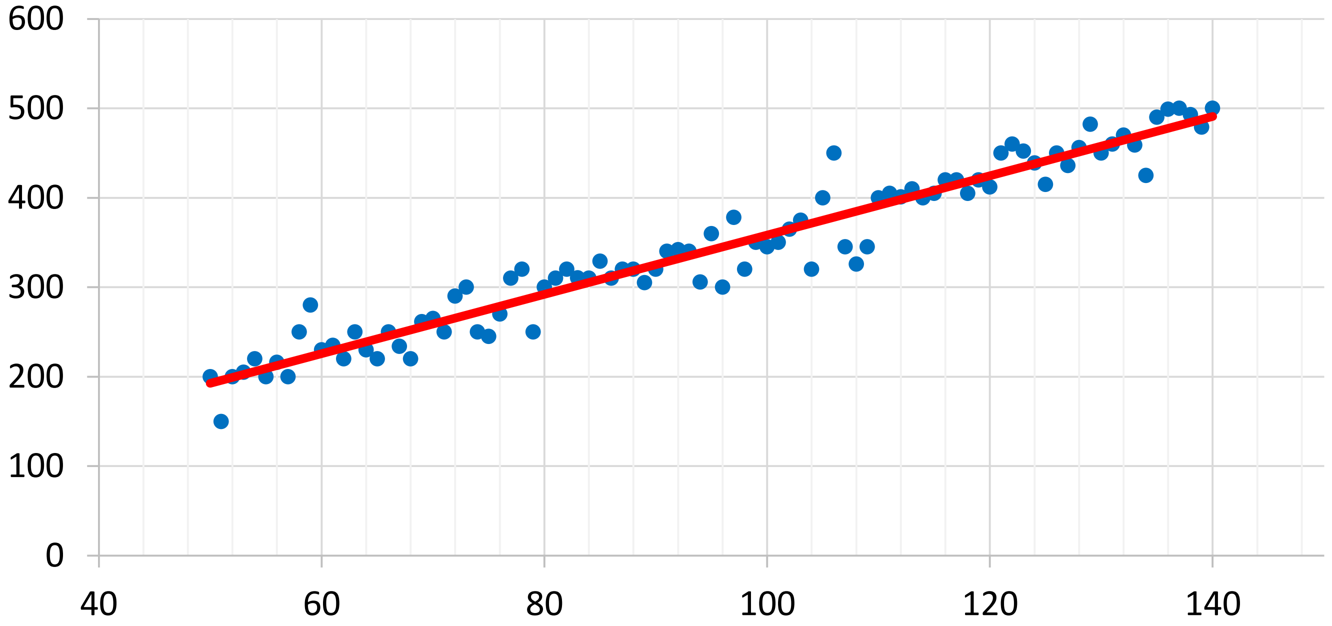
- How do we represent h(x)?

$$h_{(x)} = \theta_1 x + \theta_0$$

How do we choose parameters θ_1 and θ_0?

Linear Regression

How do we choose parameters θ_1 and θ_0?



Linear Regression

How do we choose parameters θ_1 and θ_0?

- Choose $\boldsymbol{\theta_1}$ and $\boldsymbol{\theta_0}$ in way that $\mathbf{h_{(x)}}$ is close to y for all training samples.

$$x \longrightarrow y$$

$$x \longrightarrow h_{(x)}$$

Linear Regression

How do we realize that h(x) is close to y for all training samples?

- We a need a cost function:
- For this problem, we use:

Mean square error (**MSE**): MSE is the average squared loss per example over the whole dataset. To calculate MSE, sum up all the squared losses for individual examples and then divide by the number of examples:

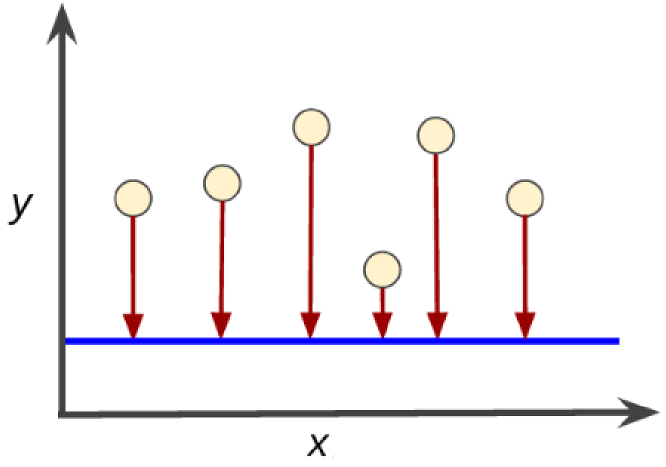
$$MSE = \frac{1}{N} \sum_{(x,y)\in D} (y - \underbrace{prediction(x)}_{h(x)})^2$$

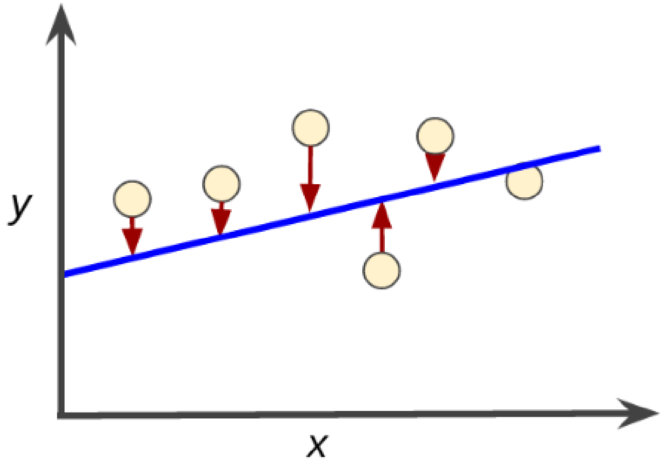
Linear Regression

How do we realize that h(x) is close to y for all training samples?

- For this problem, we use:

$$MSE = \frac{1}{N} \sum_{(x,y)\in D} (y - prediction(x))^2$$



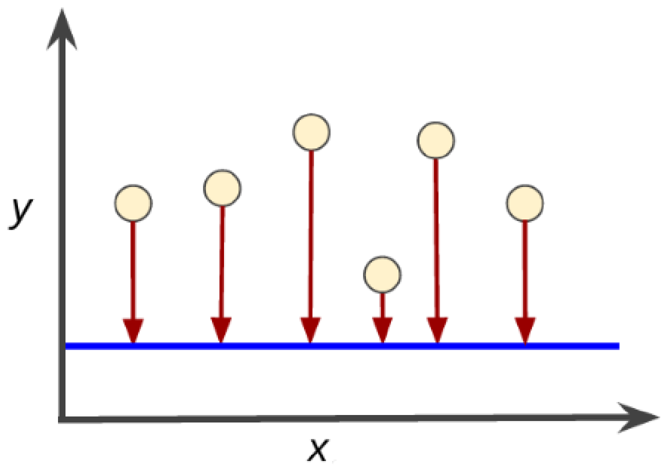


Linear Regression

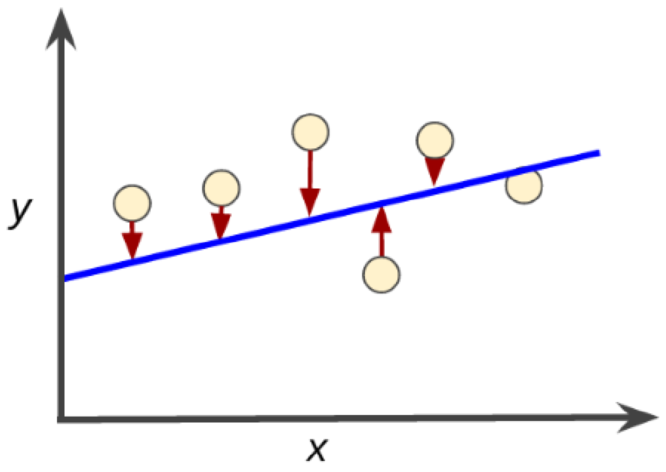
How do we realize that h(x) is close to y for all training samples?

- For this problem, we use:

$$MSE = \frac{1}{N} \sum_{(x,y)\in D} (y - prediction(x))^2$$



High Loss



Low Loss

Linear Regression

- **Hypothesis:**

  $h_{(x)} = \theta_1 x + \theta_0$

- **Parameters:**

  θ_1 , θ_0

- **Cost Function:**

  $J(\theta_1 , \theta_0) = \frac{1}{2m} \sum_1^m (h(x^{(i)}) - y^{(i)})^2$

- **Goal:**

  minimize $J(\theta_1 , \theta_0)$

Linear Regression

Suppose that we have only θ_1 parameter:

- **Hypothesis:**

  $h_{(x)} = \theta_1 x$

- **Parameters:**

  θ_1

- **Cost Function:**

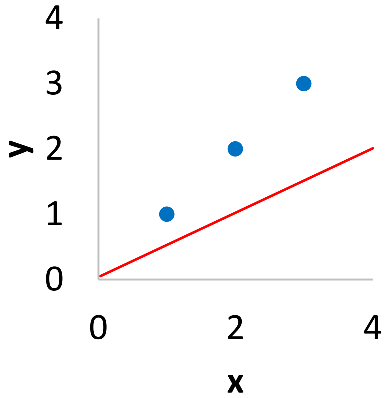
  $J(\theta_1) = \frac{1}{2m} \sum_{1}^{m} (h(x^{(i)}) - y^{(i)})^2$

- **Goal:**

  minimize $J(\theta_1)$

Linear Regression

Suppose that we have only θ_1 parameter: $h_{(x)} = \theta_1 x$

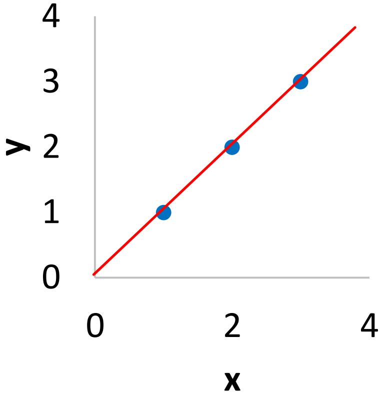


θ_1 = 0.5 -> y = 0.5x

$J(\theta_1) = \frac{1}{2m} \sum_1^m (h(x^{(i)}) - y^{(i)})^2$

$J(1) = \frac{1}{2*3} \sum_1^3 (h(x^{(i)}) - y^{(i)})^2$

$J(1) = \frac{1}{2*3} * \left((0.5 - 1)^2 + (1 - 2)^2 + (1.5 - 3)^2\right) = \frac{3.5}{6}$ = 0.58



θ_1 = 1 -> y = x

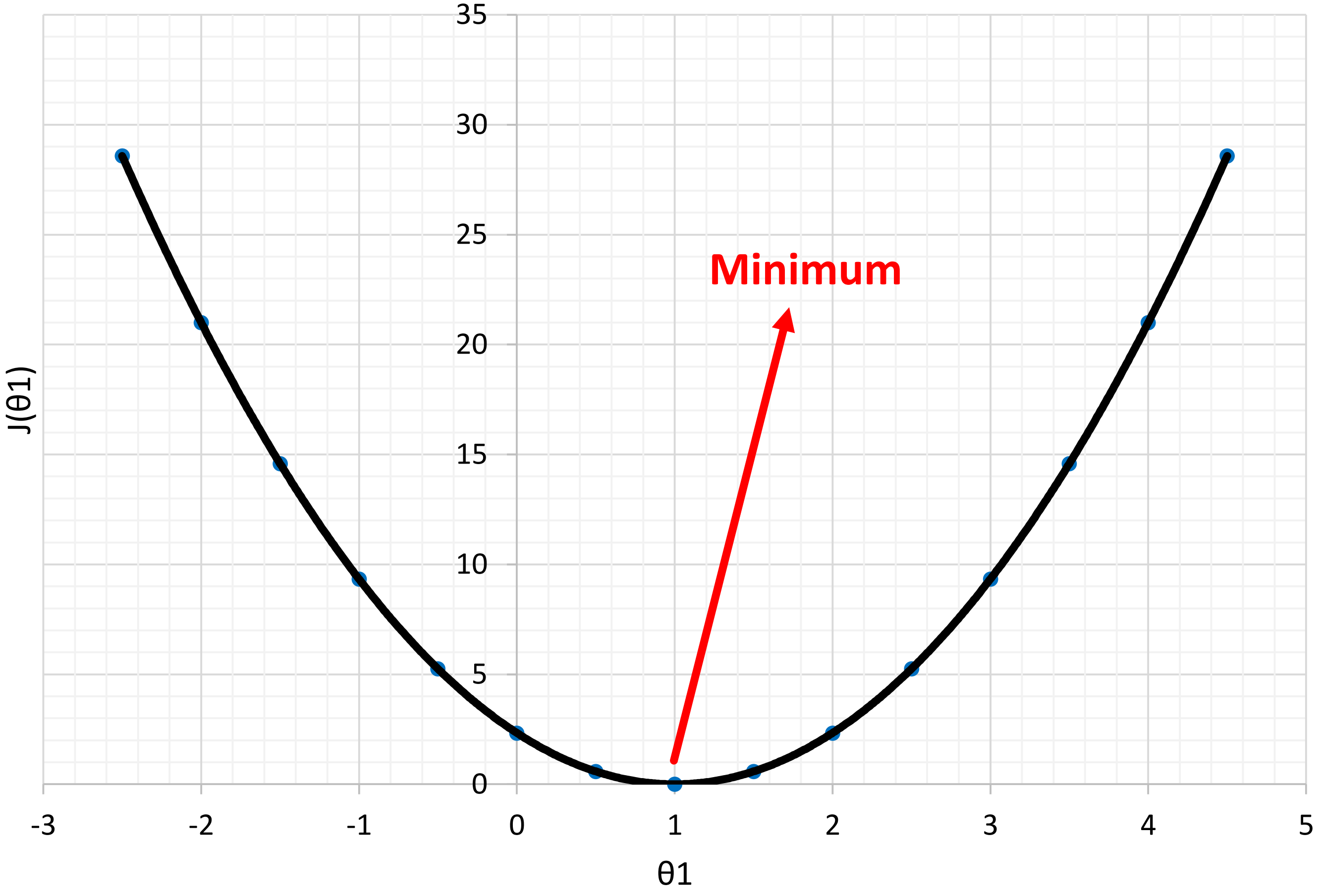
$J(\theta_1) = \frac{1}{2m} \sum_1^m (h(x^{(i)}) - y^{(i)})^2$

$J(1) = \frac{1}{2*3} \sum_1^3 (h(x^{(i)}) - y^{(i)})^2$

$J(1) = \frac{1}{2*3} * \left((1 - 1)^2 + (2 - 2)^2 + (3 - 3)^2\right) = 0$

Linear Regression

Suppose that we have only θ_1 parameter: $h_{(x)} = \theta_1 x$

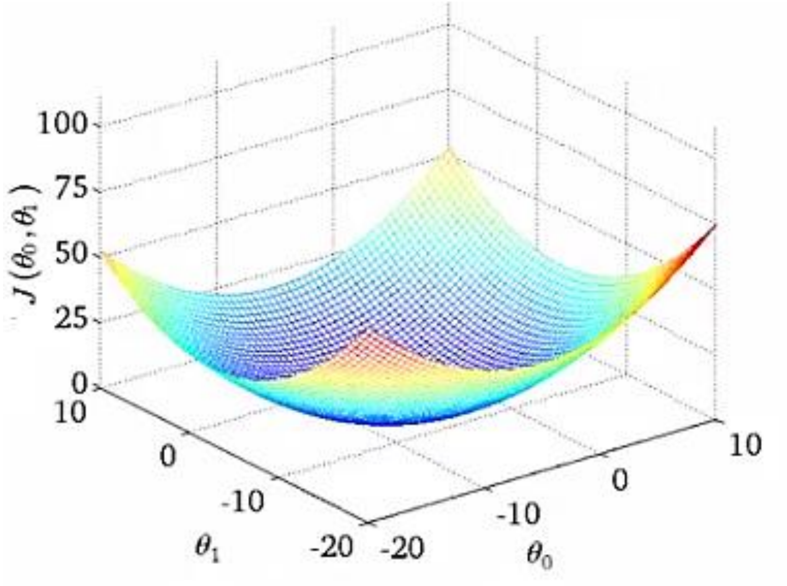


Linear Regression

Suppose that we have only θ_1 parameter: $h_{(x)} = \theta_1 x$

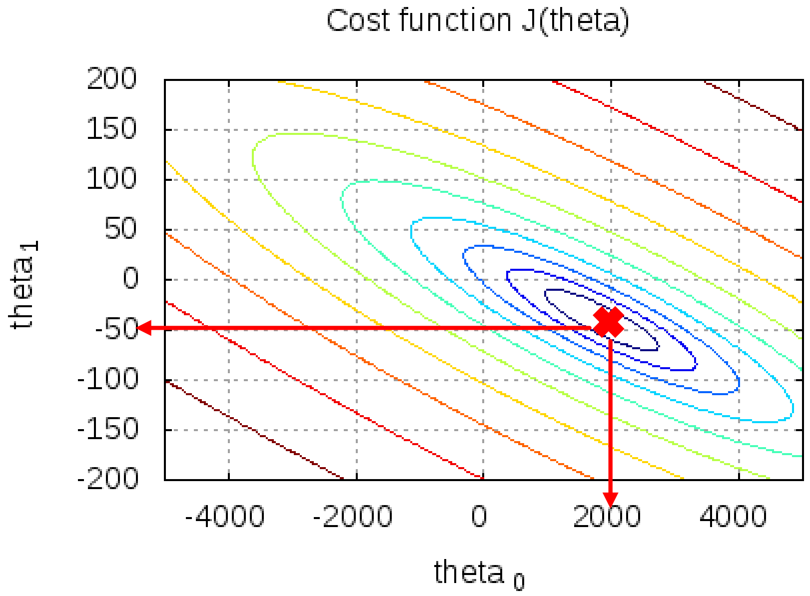
Linear Regression

Now suppose that we have θ_0 and θ_1 parameters: $h_{(x)} = \theta_1 x + \theta_0$



Linear Regression

Now suppose that we have θ_0 and θ_1 parameters: $h_{(x)} = \theta_1 x + \theta_0$



Linear Regression

- **Cost Function:**

  $J(\theta_1 , \theta_0) = \frac{1}{2m} \sum_1^m (\mathrm{h}(\mathrm{x}^{(i)}) - y^{(i)})^2$

- **Goal:**

  minimize $J(\theta_1 , \theta_0)$

- **Procedure:**

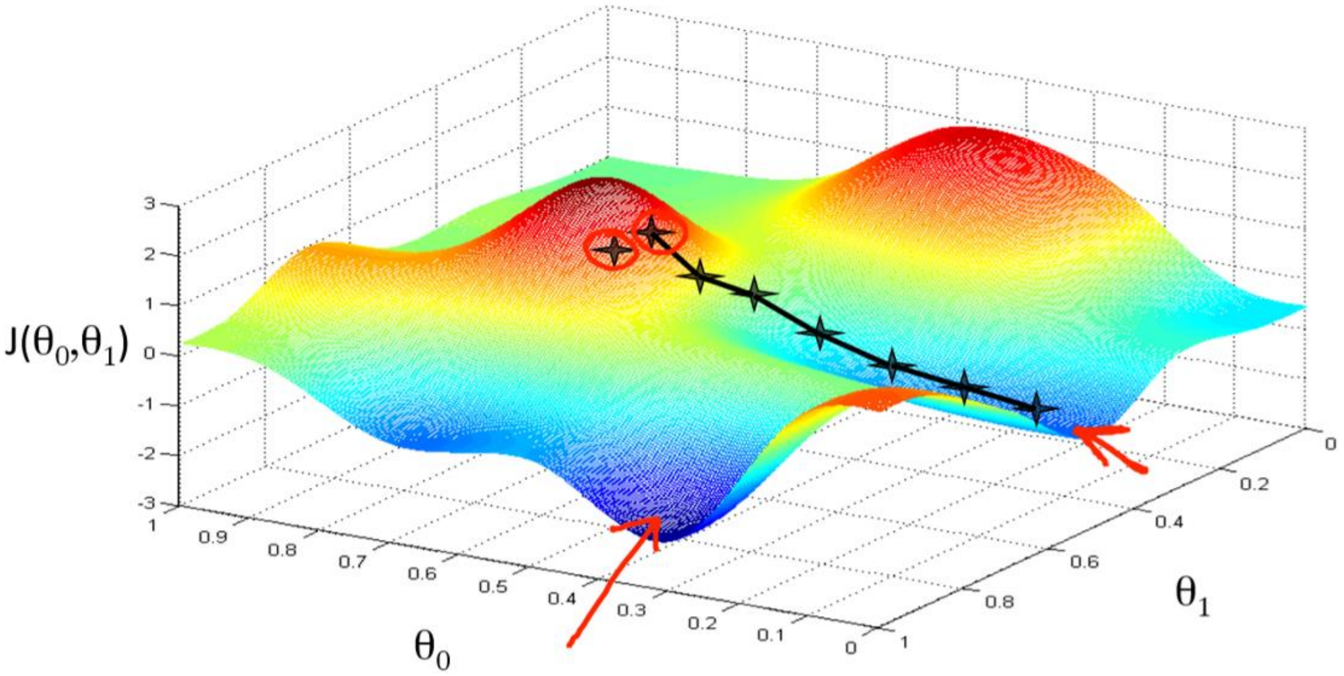
  choose θ_1 , θ_0 randomly

  keep changing θ_1 , θ_0 to reduce $J(\theta_1 , \theta_0)$

  until reach a minimum point

Linear Regression

Now suppose that we have θ_0 and θ_1 parameters: $h_{(x)} = \theta_1 x + \theta_0$



Linear Regression

- **Gradient Descent algorithm**

Repeat until convergence {

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

}

Linear Regression

- **Gradient Descent algorithm**

$$
\begin{aligned}
&\text{repeat until convergence } \{ \\
&\quad \theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right) \\
&\quad \theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right) \cdot x^{(i)} \\
&\}
\end{aligned}
$$

Linear Regression

- **Gradient Descent algorithm**

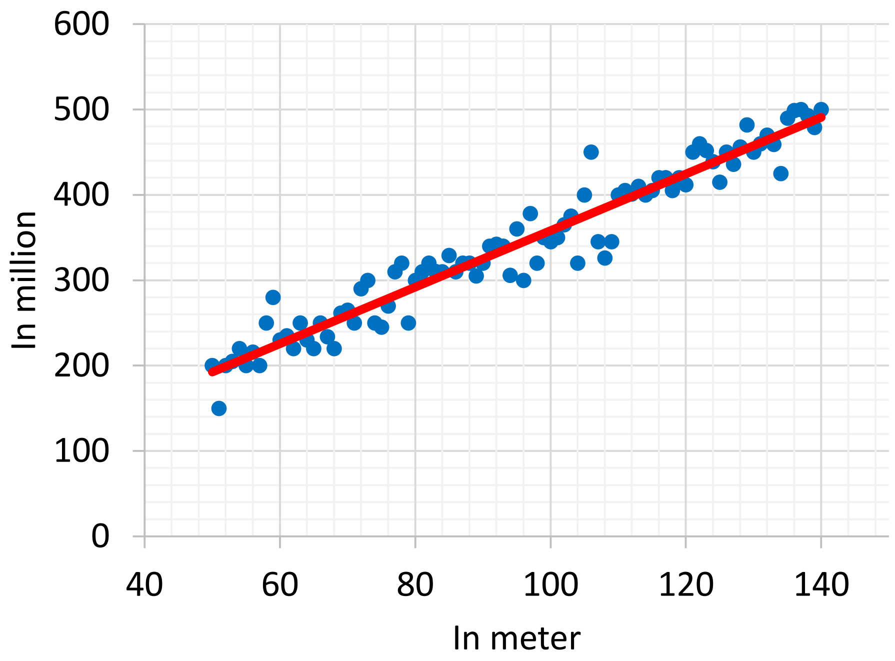
$$\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) = \frac{\partial}{\partial \theta_j} \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

$$= \frac{\partial}{\partial \theta_j} \frac{1}{2m} \sum_{i=1}^{m} \left(\theta_0 + \theta_1 x^{(i)} - y^{(i)}\right)^2$$

$$j = 0 : \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)$$

$$j = 1 : \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right) \cdot x^{(i)}$$

Linear Regression



05_linear_regression_python

Supervised Learning

- Given the “right answer” for each example in the data.

Classification Problem

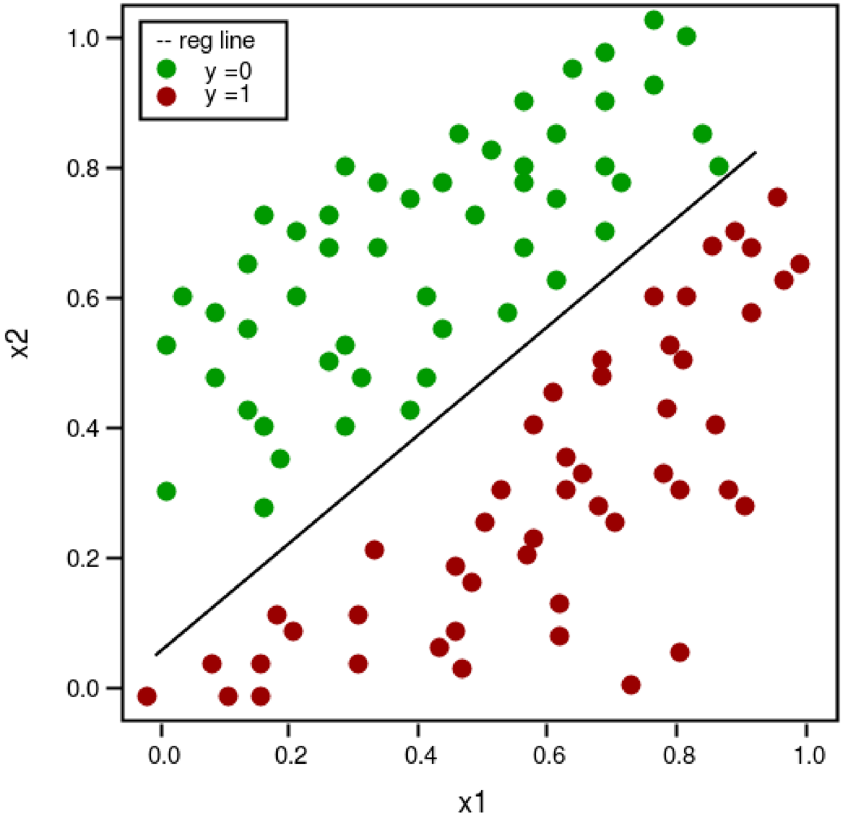
- Predict discrete value
  - Email: spam / not spam
  - Image: cat / not cat (dog)
  - Medical data: patient / healthy

Logistic Regression

- Logistic Regression:

  In statistics, the **logistic model** (or **logit model**) is used to model the probability of a certain class or event existing such as pass/fail, win/lose, alive/dead or healthy/sick. This can be extended to model several classes of events such as determining whether an image contains a cat, dog, lion, etc. Each object being detected in the image would be assigned a probability between 0 and 1 and the sum adding to one. [Wikipedia]

Logistic Regression



Logistic Regression

- How do we represent h(x)?

$$h_{(x)} = \theta_1 x + \theta_0$$

Logistic regression

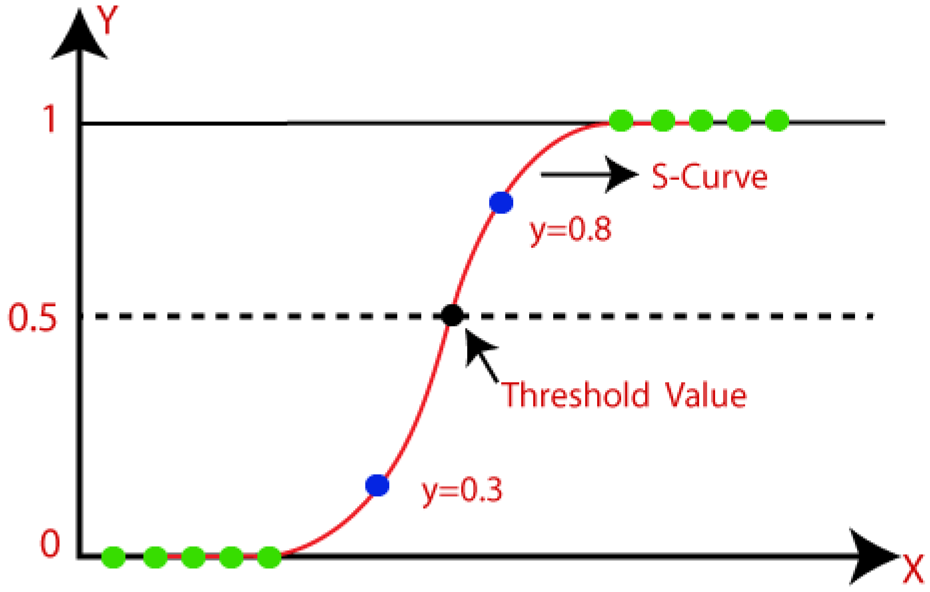
- **$y = \{0, 1\}$**
- **Threshold: 0.5**
- **If $h_{(x)} >= 0.5$ -> $y = 1$**
- **If $h_{(x)} < 0.5$ -> $y = 0$**

Logistic Regression

- How do we represent h(x)?

$$h_{(x)} = \theta_1 x + \theta_0$$

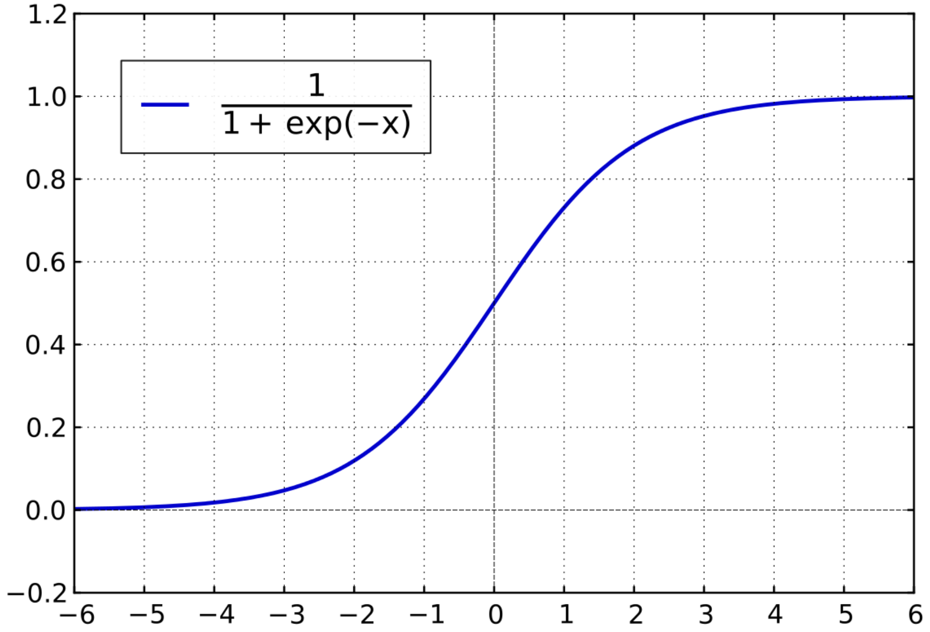
How do we apply threshold?



Logistic Regression

- How do we represent h(x)?

$$h_{(x)} = \theta_1 x + \theta_0$$



How do we apply threshold?

- **Sigmoid Function**
- **0 <= Output <= 1**
- **Output >= 0.5: 1 else 0**

Logistic Regression

- How do we represent h(x)?

$$h_{(x)} = \theta_1 x + \theta_0$$

$$z = h_{(x)} = \theta_1 x + \theta_0$$

$$g(z) = \frac{1}{1+ e^{-z}}$$

How do we choose parameters θ_1 and θ_0?

Logistic Regression

How do we choose parameters θ_1 and θ_0?

- Choose $\boldsymbol{\theta_1}$ and $\boldsymbol{\theta_0}$ in way that $\mathbf{h_{(x)}}$ is close to y for all training samples.

$$x \longrightarrow y\ \{0\ ,\ 1\}$$

$$x \longrightarrow h_{(x)}\ \{0,\ 1\}$$

Logistic Regression

How do we realize that h(x) is close to y for all training samples?

- We a need a cost function:
- For this problem, we use:

Binary Cross-Entropy:

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

Binary Cross-Entropy / Log Loss

Logistic Regression

How do we realize that h(x) is close to y for all training samples?

- We a need a cost function:
- For this problem, we use:

  Binary Cross Entropy:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

$$\mathrm{Cost}(h_\theta(x), y) = -\log(h_\theta(x)) \qquad \text{if } y = 1$$

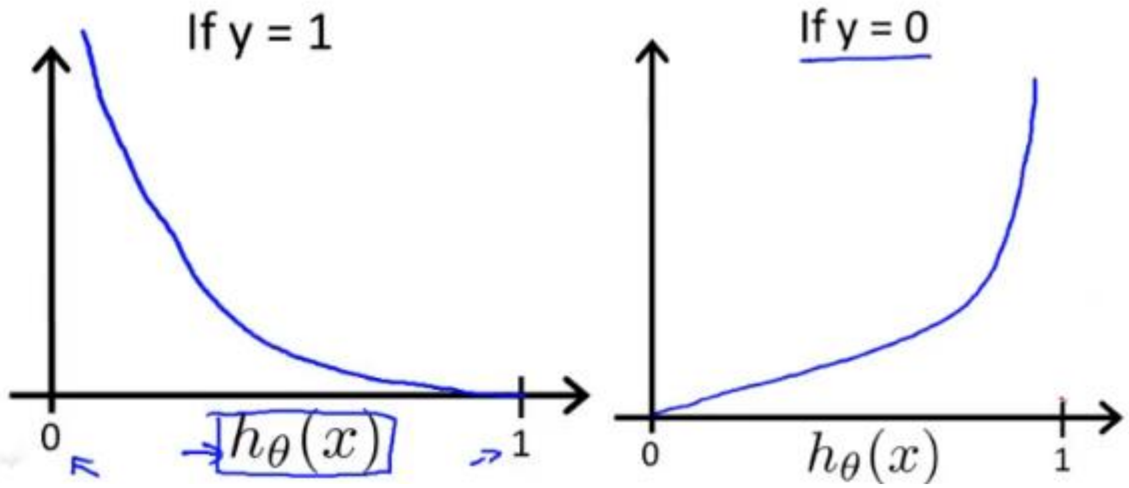
$$\mathrm{Cost}(h_\theta(x), y) = -\log(1 - h_\theta(x)) \qquad \text{if } y = 0$$

Logistic Regression

How do we realize that h(x) is close to y for all training samples?

- We a need a cost function:
- For this problem, we use:

Binary Cross Entropy:



Logistic Regression

- **Hypothesis:**

  $h_{(x)} = \theta_1 x + \theta_0$

- **Parameters:**

  θ_1 , θ_0

- **Cost Function:**

  $J(\theta_1 , \theta_0)$

- **Goal:**

  minimize $J(\theta_1 , \theta_0)$

Logistic Regression

- **Gradient Descent algorithm**

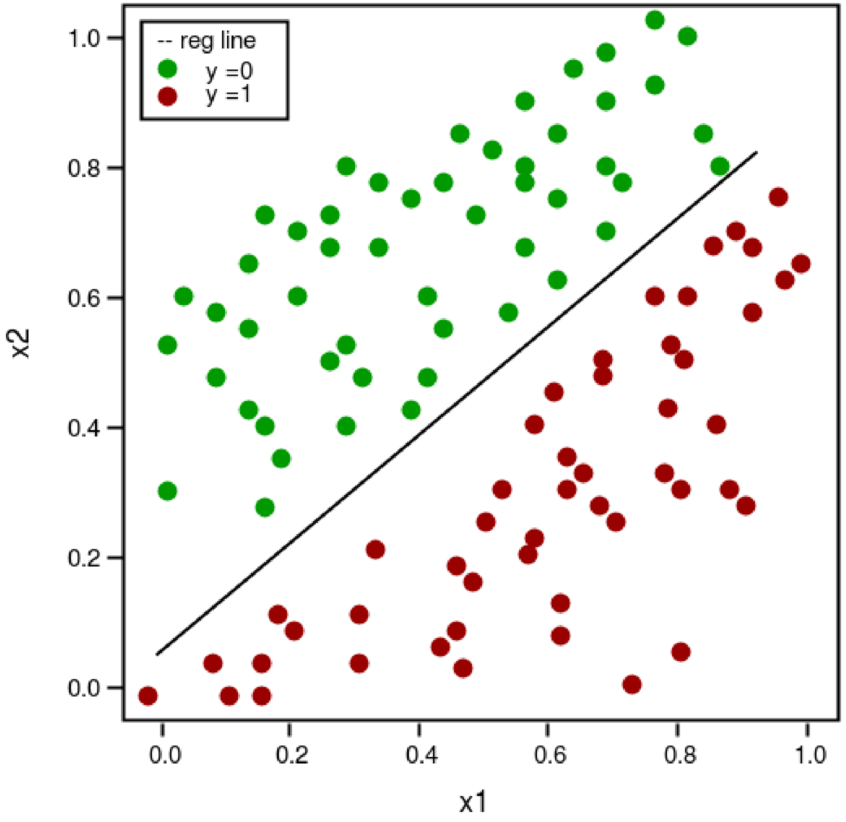
repeat until convergence {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right) \cdot x^{(i)}$$

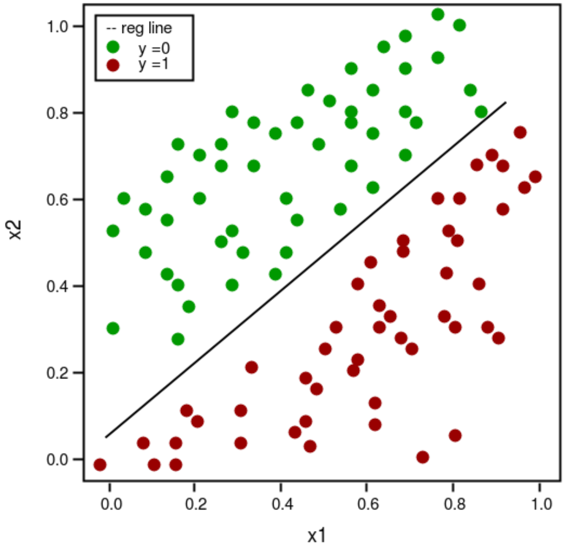
}

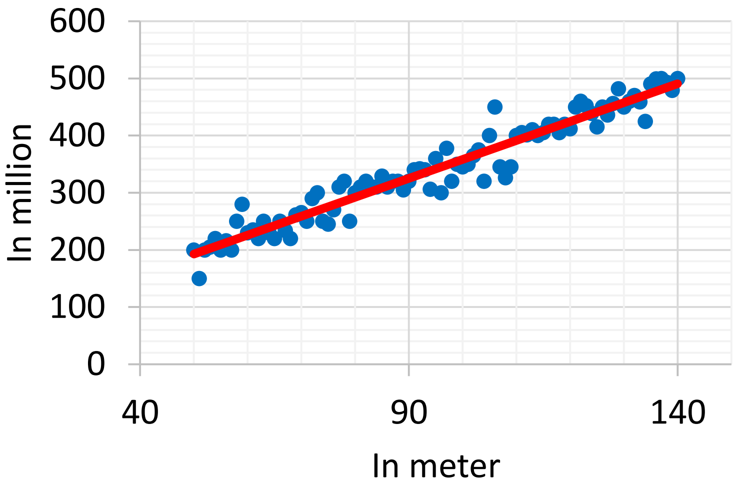
Logistic Regression



[06_logistic_regression_python]()

Logistic Regression and Linear Regression





07_linear_logistic_regression_sklearn

Good resources!

- https://ml-cheatsheet.readthedocs.io/en/latest/linear_algebra.html
- https://ml-cheatsheet.readthedocs.io/en/latest/calculus.html#introduction
- https://ml-cheatsheet.readthedocs.io/en/latest/probability.html
- https://www.youtube.com/watch?v=WnqQrPNYz5Q
- https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a

References

- https://www.coursera.org/learn/machine-learning