

Machine learning handbook

Classifier

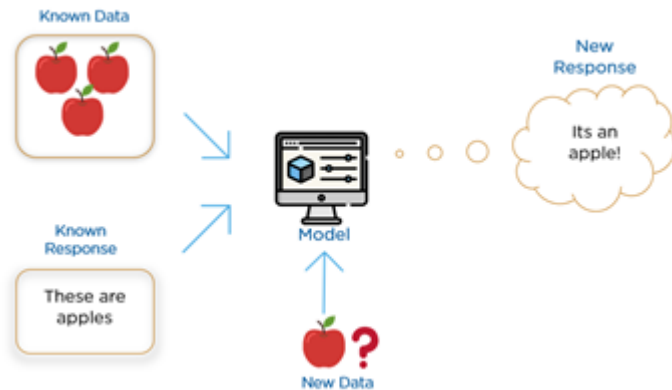
Taeyang Yang

Update: July 13, 2018

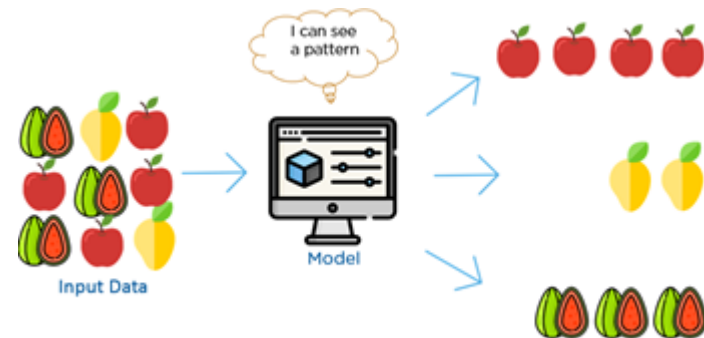
BCILAB, UNIST

Supervised & unsupervised learning

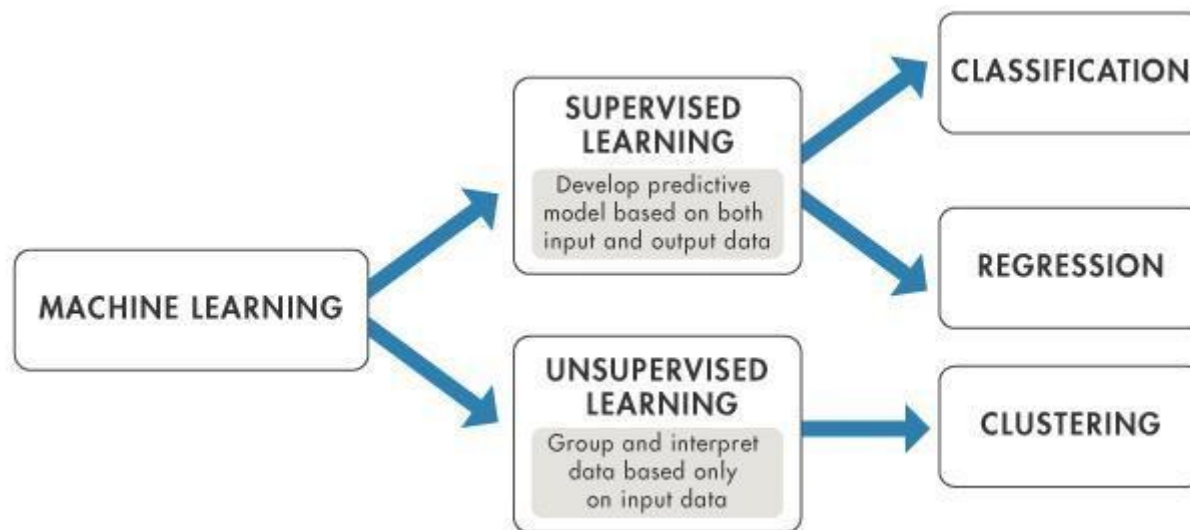
Supervised learning



Unsupervised learning



Classification, Regression & Clustering

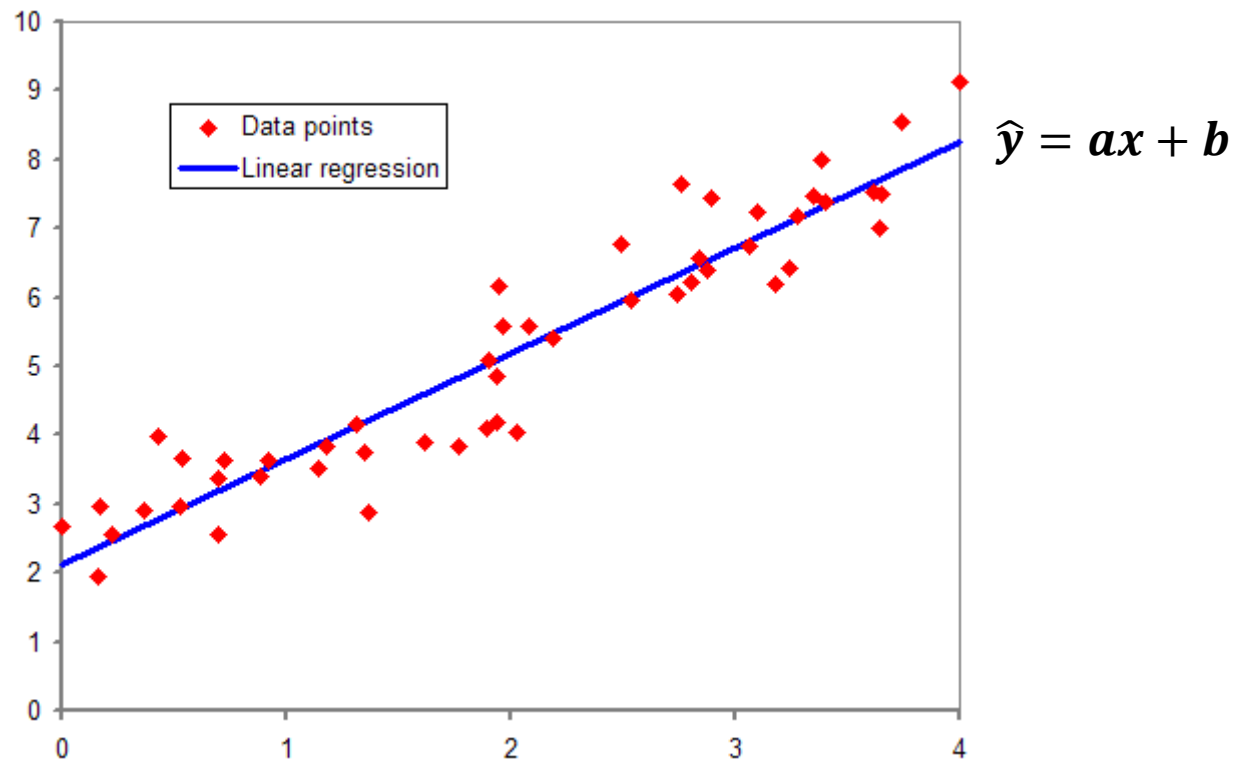


1

Supervised learning

- Linear Regression
- Naïve Bayes (NB) classifier
- Linear Discriminant Analysis (LDA)
- Support Vector Machine (SVM)

Linear Regression (overview)



Linear Regression

- Same expression !

$$\mathbf{y} = \mathbf{ax} + \mathbf{b}$$

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

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix} = \begin{bmatrix} x_{11} & \cdots & x_{1p} \\ x_{21} & \cdots & x_{2p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix}, \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

- Cost (Loss) function

$$\begin{aligned} J(\boldsymbol{\beta}) = MSE &= \frac{1}{2m} \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)})^2 \\ &= \frac{1}{2m} \sum_{i=1}^m (X^{(i)}\boldsymbol{\beta} + \varepsilon - y^{(i)})^2 \end{aligned}$$

Linear Regression

- Cost (Loss) function

$$J(\beta) = MSE = \frac{1}{2m} \sum_{i=1}^m (X^{(i)}\beta + \varepsilon - y^{(i)})^2$$

- Derivate

$$\begin{aligned}\frac{\partial J}{\partial \varepsilon} &= \frac{1}{m} \sum_{i=1}^m (X^{(i)}\beta + \varepsilon - y^{(i)}) \\ \frac{\partial J}{\partial \beta} &= \frac{1}{m} \sum_{i=1}^m (X^{(i)}\beta + \varepsilon - y^{(i)})X^{(i)}\end{aligned}$$

Linear Regression

- Solution 1: Normal equation, **derivate=0**

$$\begin{aligned}\varepsilon m + \beta \sum_{i=1}^m X^{(i)} &= \sum_{i=1}^m y^{(i)} \\ \varepsilon \sum_{i=1}^m X^{(i)} + \beta \sum_{i=1}^m X^{(i)2} &= \sum_{i=1}^m y^{(i)} X^{(i)}\end{aligned}$$

- Matrix expression

$$\begin{bmatrix} \varepsilon \\ \beta \end{bmatrix}^T \begin{bmatrix} m & \sum_{i=1}^m X^{(i)} \\ \sum_{i=1}^m X^{(i)} & \sum_{i=1}^m X^{(i)2} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^m y^{(i)} \\ \sum_{i=1}^m y^{(i)} X^{(i)} \end{bmatrix}$$

Linear Regression

- Trick

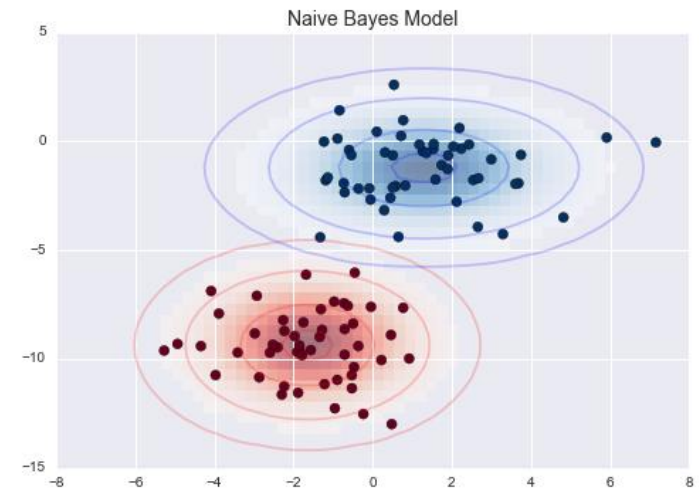
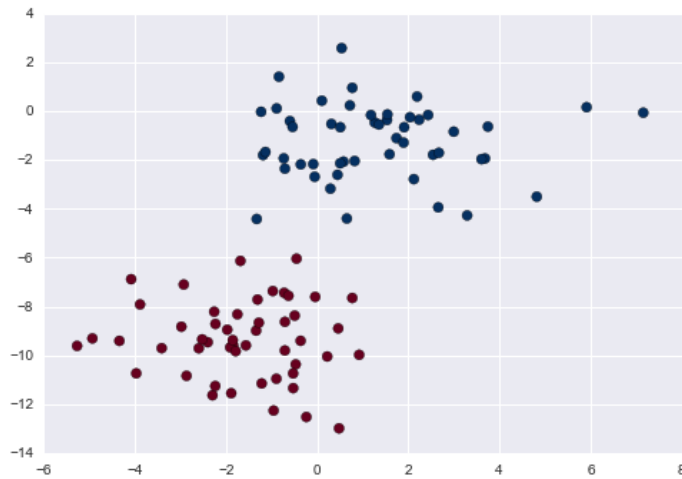
$$\mathbf{w} = \begin{bmatrix} \varepsilon \\ \beta \end{bmatrix}, \mathbf{X} = \begin{bmatrix} 1 & \mathbf{x}_1^T \\ 1 & \mathbf{x}_2^T \\ \vdots & \vdots \\ 1 & \mathbf{x}_n^T \end{bmatrix}, \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

$$\mathbf{X}^T \mathbf{X} = \begin{bmatrix} m & \sum_{i=1}^m x_i \\ \sum_{i=1}^m x_i & \sum_{i=1}^m x_i^2 \end{bmatrix}, \mathbf{X}^T \mathbf{y} = \begin{bmatrix} \sum_{i=1}^m y_i \\ \sum_{i=1}^m y_i x_i \end{bmatrix}$$

- Estimated weight

$$\mathbf{w} \mathbf{X}^T \mathbf{X} = \mathbf{X}^T \mathbf{y}$$
$$\mathbf{w} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

Naïve Bayes classifier (overview)



Naïve Bayes classifier

- Bayes theorem-based model
 - Bayes' theorem

$$\underset{\text{Posterior probability}}{p(C_k|x)} = \frac{\overset{\text{Likelihood}}{p(x|C_k)}\overset{\text{Class prior probability}}{p(C_k)}}{\underset{\text{Predictor prior probability}}{p(x)}}$$

Naïve Bayes classifier

$$p(C_k|x) = \frac{p(x|C_k)p(C_k)}{p(x)}$$

- **Joint probability:** chain rule of conditional probability

$$\begin{aligned} p(C_k, x_1, \dots, x_n) &= p(x_1, \dots, x_n, C_k) \\ &= p(x_1|x_2, \dots, x_n, C_k)p(x_2, \dots, x_n, C_k) \\ &= p(x_1|x_2, \dots, x_n, C_k)p(x_2|x_3, \dots, x_n, C_k)p(x_3, \dots, x_n, C_k) \\ &= \dots \\ &= p(x_1|x_2, \dots, x_n, C_k)p(x_2|x_3, \dots, x_n, C_k) \dots p(x_n|C_k)p(C_k) \end{aligned}$$

- **Assumption of “naïve conditional independence” :**
 - Each x_i is **conditionally independent** of every other features x_j for $j \neq i$ given the category C_k

$$p(x_i|x_{i+1}, \dots, x_n, C_k) = p(x_i|C_k)$$

- **Posterior**

$$\begin{aligned} p(C_k|x_1, \dots, x_n) &\propto p(C_k, x_1, \dots, x_n) = p(C_k)p(x_1|C_k)p(x_2|C_k) \dots p(x_n|C_k) \\ &= p(C_k) \prod_{i=1}^n p(x_i|C_k) \end{aligned}$$

Naïve Bayes classifier

- Maximum a posterior (MAP)

$$\hat{y} = \operatorname{argmax}_{k \in \{1, \dots, K\}} p(C_k) \prod_{i=1}^n p(x_i | C_k)$$

- What is $p(x_i | C_k)$?

- parameter estimation **based on assumption of likelihood**

- Gaussian naïve Bayes $\rightarrow \mu, \sigma$

$$p(x = v | C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(v-\mu_k)^2}{2\sigma_k^2}}$$

- Multinomial naïve Bayes $\rightarrow p_{ki}$

$$p(X | C_k) = \frac{(\sum_i x_i)!}{\prod_i x_i!} \prod_i p_{ki}^{x_i}$$

Linear Discriminant Analysis (LDA)

Shrinkage Linear discriminant analysis

Support Vector Machine (SVM)

2

Unsupervised learning

- Random forest
- K-nearest neighborhood
- K-means clustering

Random forest

K-nearest neighborhood

K-means clustering