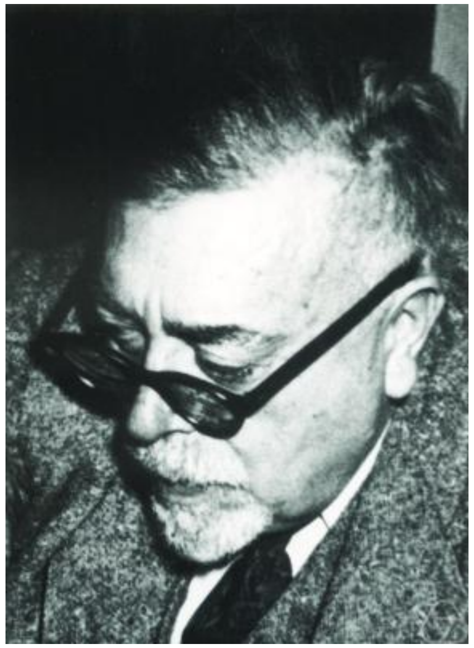
Biomedical Data Analysis in Python3

Introduction to Scientific Computing, Visualization, Machine Learning and Debugging

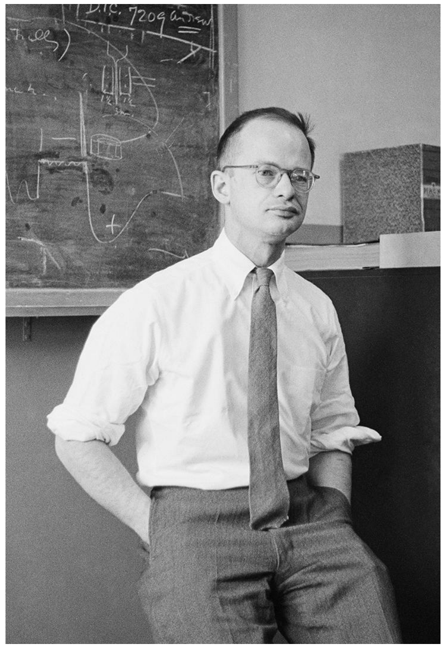
https://www.webofstories.com/play/marvin.minsky/100

Introduction



Norbert Wiener
(1894 - 1964)

Warren S. McCulloch
(1898 - 1969)



Walter Pitts
(1923 - 1969)

- Introduction
- Regression
- Dimensional Reduction
- Clustering
- Classification

Regression

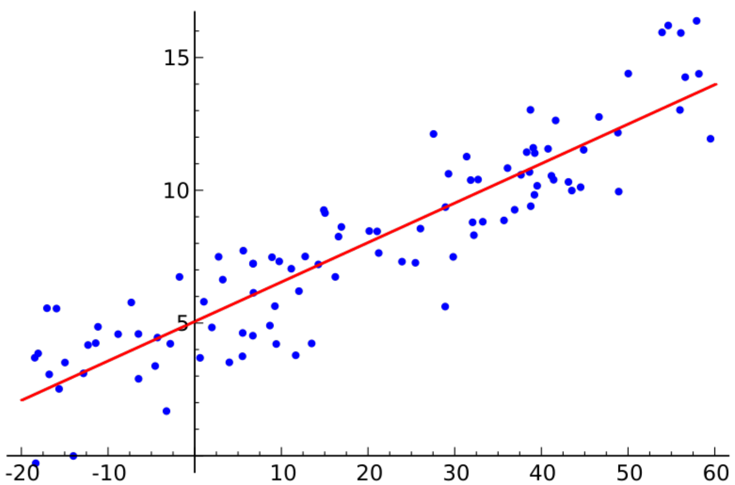
- Linear Regression
- Linear Correlation
- Logistic Regression

Linear Regression

$$y_i = w_1 x_{i1} + w_2 x_{i2} + \cdots + w_n x_{in} + b$$
$$= \mathbf{x}_i \mathbf{w}^T + b_i$$

$$\mathbf{Y}^{m\times 1} = \mathbf{X}^{m\times n} \mathbf{w}^{n\times 1} + \mathbf{b}^{m\times 1}$$

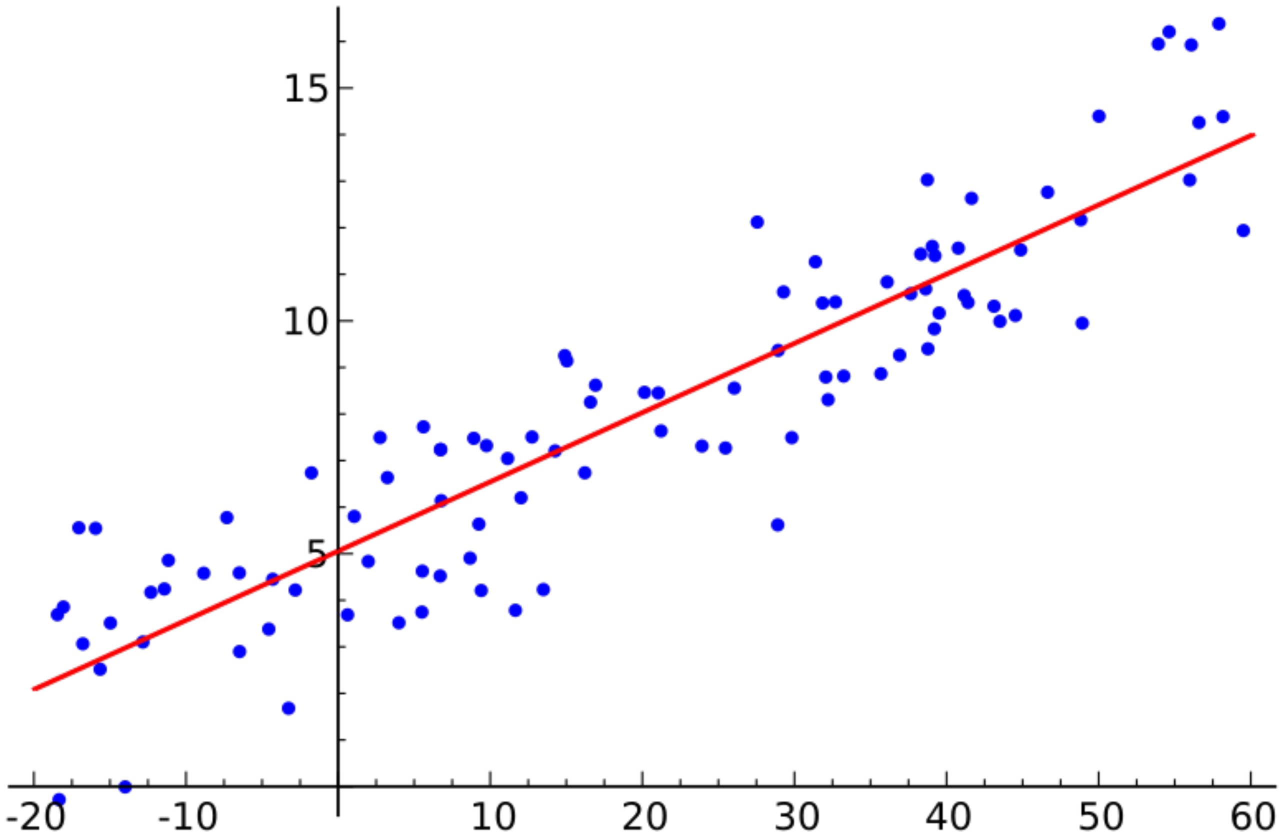
Linear
Independent
Normal
Equal variance



When n = 1

$$Y \sim \mathcal{N}(\mu_{Y|X}, \sigma^2) \qquad \mu_{Y|X} = \alpha + \beta X$$

$$Y = \mu_{Y|X} + \epsilon = \alpha + \beta X + \epsilon$$



$$\hat{Y} = a + bX$$

$$Y - \hat{Y} = \hat{\epsilon}$$

When n = 1

$$Y - \bar{Y} = (\hat{Y} - \bar{Y}) + (Y - \hat{Y})$$

$$\sum (Y - \bar{Y})^2 = \sum (\hat{Y} - \bar{Y})^2 + \sum (Y - \hat{Y})^2$$

$$SST = SSR + SSE$$

$$\nu_T = n - 1,\ \nu_R = 1,\ \nu_E = n - 2$$

$$R^2 = \frac{SSR}{SST} = r^2$$

When n = 1

$$F = \frac{SSR/\nu_R}{SSE/\nu_E} \Rightarrow \; p = ???$$

$$R^2 = ???$$

Linear Correlation (aka. Pearson Correlation)

$$
\begin{aligned}
r &= \frac{\sum(X-\bar{X})(Y-\bar{Y})}{\sqrt{\sum(X-\bar{X})^2\sum(Y-\bar{Y})^2}} \\
&= \frac{\mathrm{cov}(X,Y)}{\sigma_X\sigma_Y}
\end{aligned}
$$

$$
t_r = \frac{r-0}{S_r},\ S_r = \sqrt{\frac{1-r^2}{n-1}},\ \nu = n-2
$$

Dimensional Reduction (feature extraction)

- Principle Component Analysis, PCA

Principle Component Analysis

Principal component analysis (PCA) is a statistical procedure that uses an *orthogonal transformation* to convert a set of observations of possibly correlated variables into a set of values of *linearly uncorrelated variables* called principal components.

In linear algebra, an **orthogonal transformation** is a linear transformation $T : V \to V$ on a real *inner product space* V, that preserves the inner product.

$$\langle \mathbf{u}, \mathbf{v} \rangle = \langle T\mathbf{u}, T\mathbf{v} \rangle$$

Linear uncorrelated means variables have no linear correlation, that is the correlation coefficient is 0; that is the covariance of the variables are 0.

Principle Component Analysis

This transformation (i.e. PCA) is defined in such a way that the first principal component has the *largest possible variance* (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components.

Principle Component Analysis

$$\mathbf{X}^{n\times p} = (\mathbf{x}_1, \mathbf{x}_1, \ldots, \mathbf{x}_n)^T$$

$$\mathbf{w}_{(k)} = (w_1, w_2, \ldots, w_p)_{(k)}$$

$$\mathbf{t}_{(i)} = (t_1, t_2, \ldots, t_l)_{(i)}$$

$$t_{k(i)} = \mathbf{x}_{(i)} \cdot \mathbf{w}_{(k)} \text{ for } i = 1, \ldots, n \; k = 1, \ldots, l$$

$$\mathbf{w}_{(1)} = \underset{\|\mathbf{w}\|=1}{\arg\max} \left\{ \sum_i (t_1)^2_{(i)} \right\}$$

Clustering

- Centroid-based clustering, K-Means Clustering
- Connectivity-based clustering, hierarchical clustering
- Density-based clustering
- Distribution-based clustering, etc.

Centroid-Based

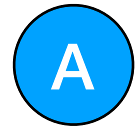
- **K-Means**

- Given a set of observations ($\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_n$), where each observation is a *d*-dimensional real vector,

- k-means clustering aims to partition the n observations into k ($\leq n$) sets $\mathbf{S} = \{S_1, S_2, \ldots, S_k\}$

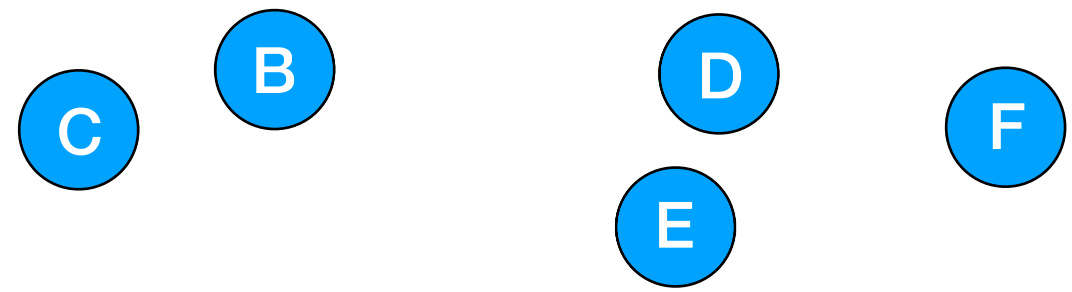
- so as to minimize the within-cluster sum of squares (WCSS) (i.e. variance).

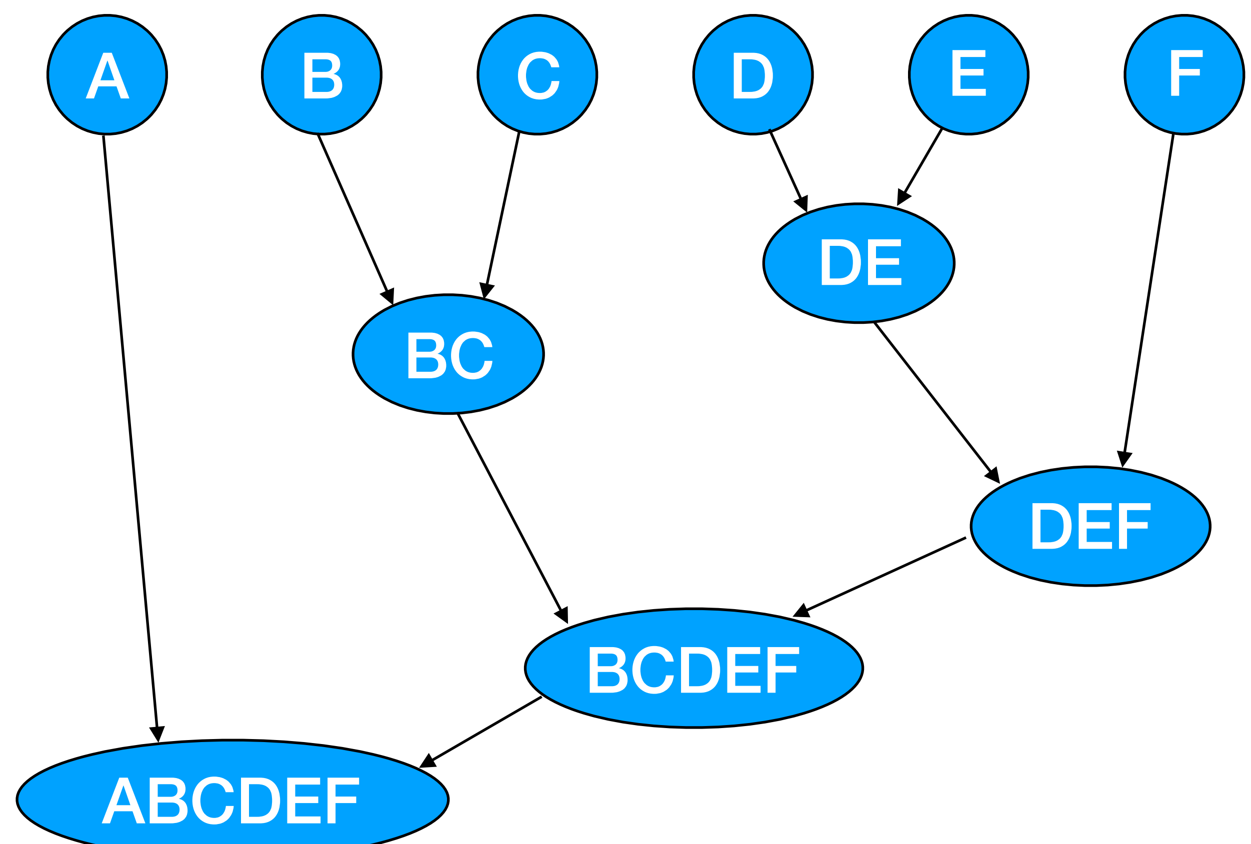
$$\underset{\mathbf{S}}{\arg\min} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \left\| \mathbf{x} - \mu_i \right\|^2$$



Connectivity-Based

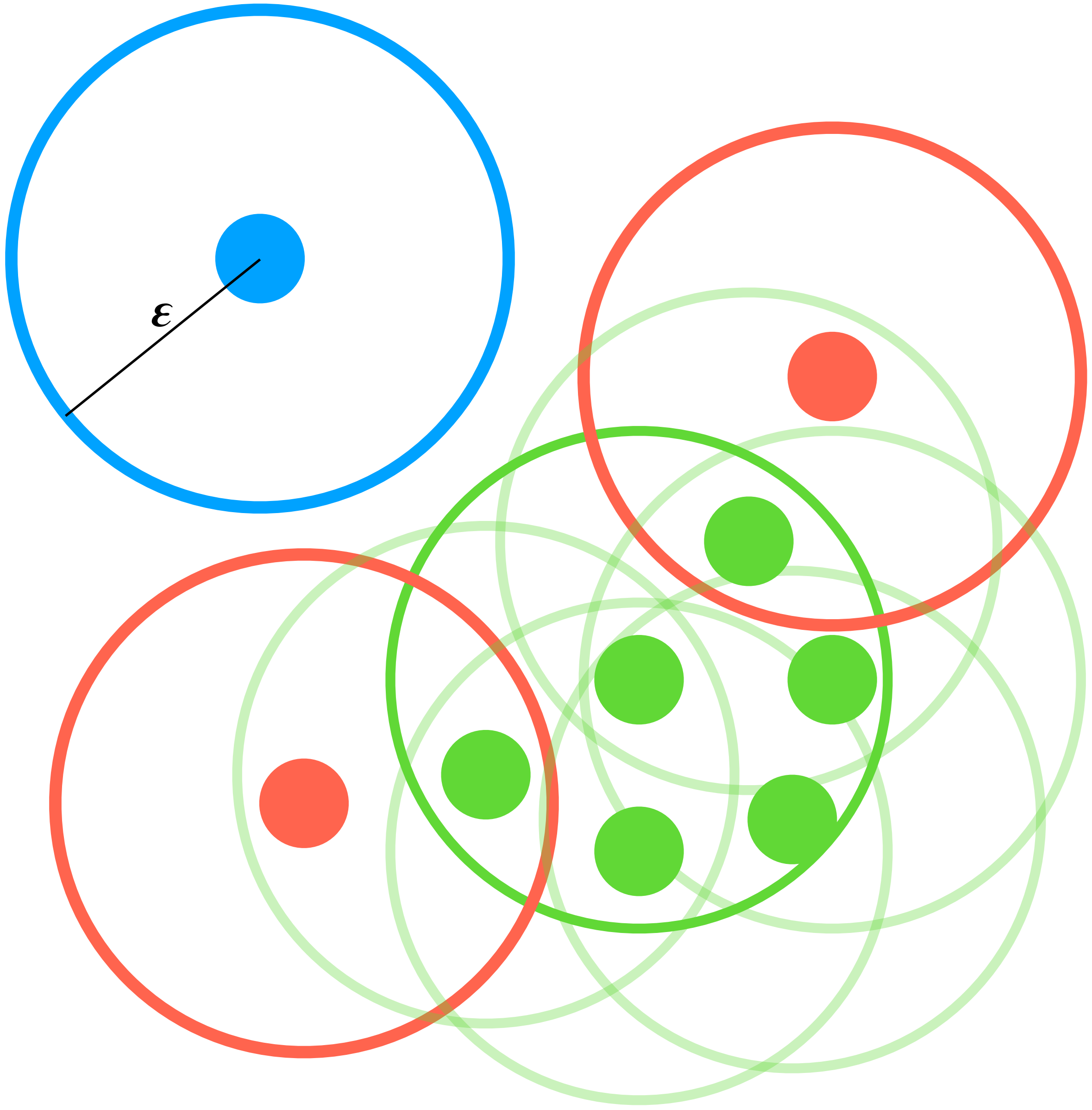
- Hierarchical clustering
  Agglomerative
  Divisive
- Metric
- Linkage criteria





Density-Based

- DBSCAN
  - Core point
  - Directly reachable
  - Reachable
  - Outlier



Classification

- Binary Classification
  - Linear Model, from Logistic to Softmax
  - Support Vector Machine
- Multi-Class Classification
  - One-against-all
  - One-against-one

Linear Model

Discriminant function

$$f(\mathbf{x}, \mathbf{w}) = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b$$
$$= \mathbf{w}^T \mathbf{x} + b$$

Decision function

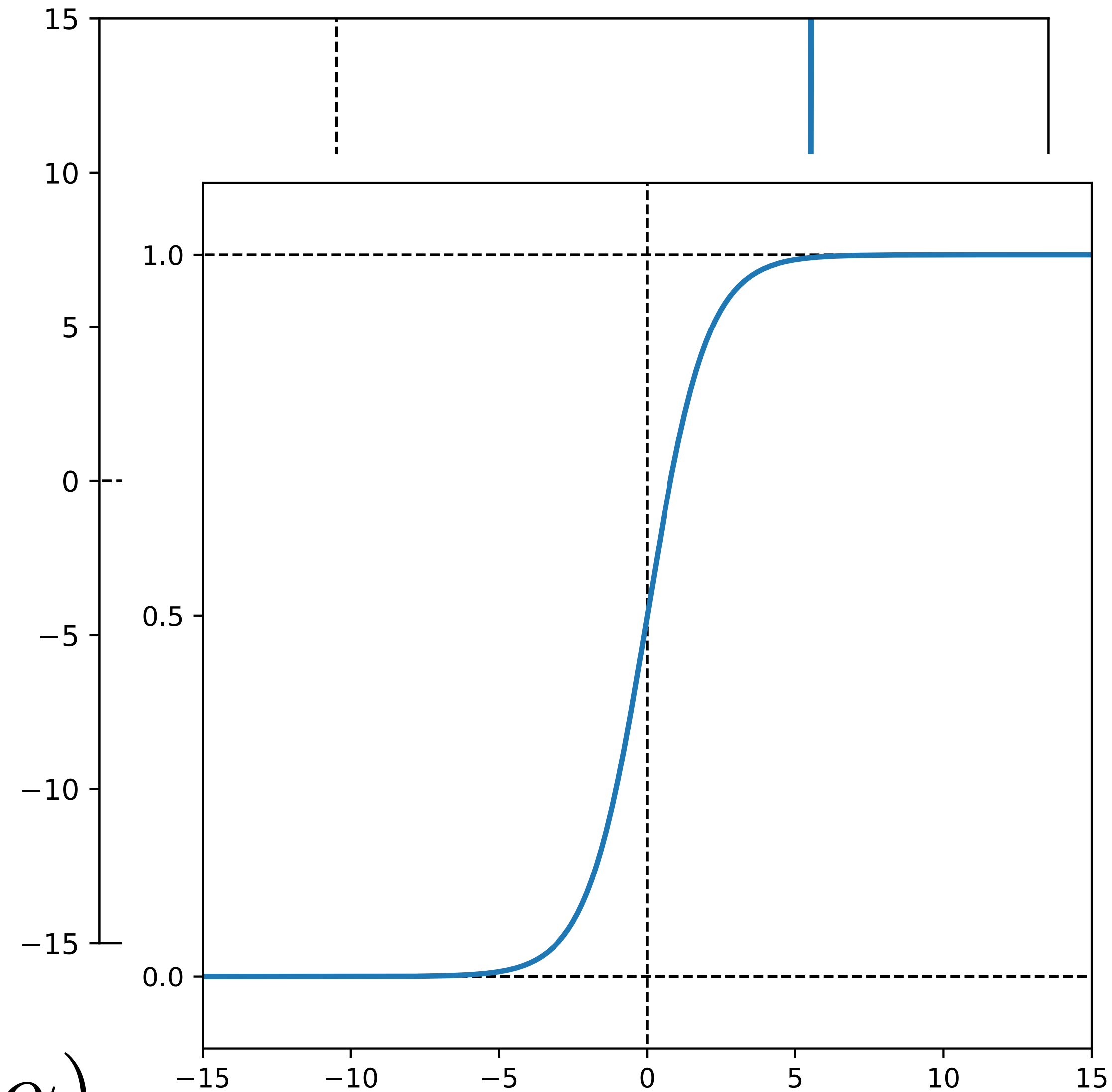
$$g(f(\mathbf{x}, \mathbf{w})) = \begin{cases} 0 & \text{when} \ldots \\ 1 & \text{when} \ldots \end{cases}$$

Deep Learning

$$g_n(\ldots g_2(g_1(f(\mathbf{x}, \mathbf{w}))))$$

Logistic Regression

- Odds $$\text{odds}(p) = \frac{p}{1-p}$$
- Odds Ratio, OR



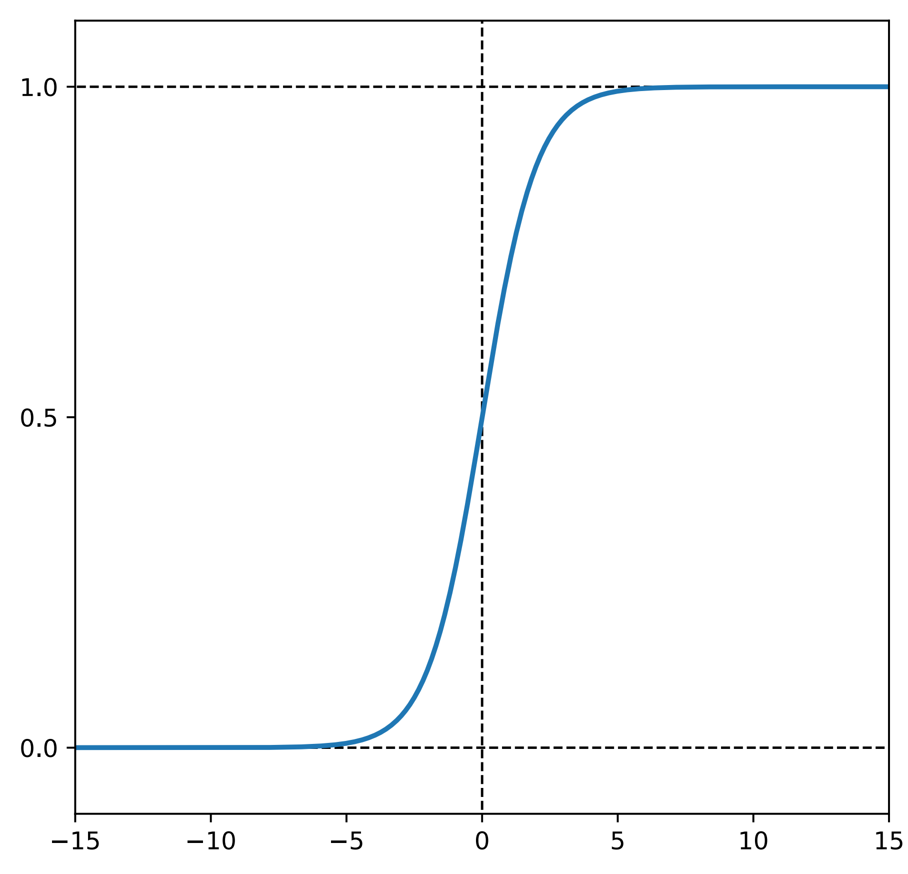
$$\text{logit}(p) = \ln(\frac{p}{1-p})$$

$$\text{logit}^{-1}(\alpha) = \text{logistic}(\alpha) = \frac{\exp(\alpha)}{\exp(\alpha)+1}$$

Logistic Regression

$$\ell = \text{logit} = \beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n$$

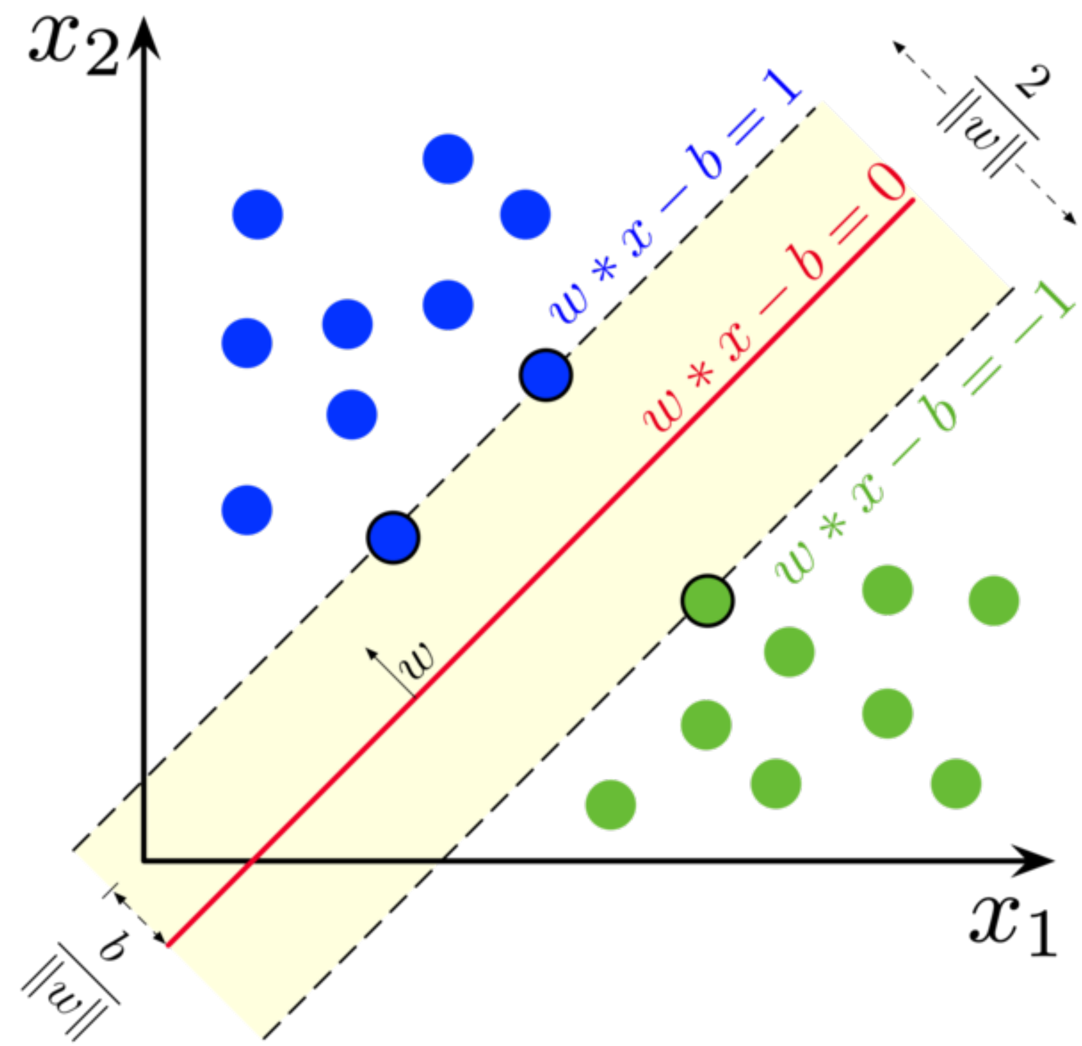
$$p = \text{logistic} = \frac{\exp(\ell)}{\exp(\ell) + 1}$$



Support Vector Machine

$$\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}, \text{where } y_n \in \{1, -1\}$$

$$y^{(n)}(\mathbf{w}^T\mathbf{x}^{(n)} + b) > 0$$



Multi-Class Classification

- One-against-all
- One-against-one

A B C D	A B C D
A B C D	A B C D
A B C D	A B C D
A B C D	A B C D
	A B C D
	A B C D

REVIEW

- Python3 Tour
- Numeric Data and Numpy
- Tabular Data and Pandas
- Visualization and Matplotlib
- Machine Learning
    - Regression
    - Dimensional Reduction
    - Clustering
    - Classification

“There are no answers, only cross-references.”

– Norbert Wiener