Machine Learning and Transportation - Introduction

Peter Chen

Shanghai University of Engineering Science

December 7-15, 2019

机器学习是什么？

- 机器学习是人工智能的一个分支
- 人工智能的研究历史有着一条从以“**推理**”为重点，到以“**知识**”为重点，再到以“**学习**”为重点的自然清晰的脉络
- 机器学习是实现人工智能的一个途径，即以机器学习为手段解决人工智能中的问题
- 机器学习在近30多年已发展为一门**多领域交叉学科**，涉及**概率论、统计学、逼近论、凸分析、计算复杂性理论等多门学科**。机器学习理论主要是设计和分析一些让计算机可以自动“学习”的算法
- 机器学习算法是一类从数据中自动分析获得规律，并利用规律对未知数据进行预测的算法
- 因为学习算法中涉及了大量的**统计学理论**，机器学习与推断统计学联系尤为密切，也被称为**统计学习理论**

What is Machine Learning?

- **Tom Mitchell** (professor of Carnegie Mellon University) provided a modern definition: "A computer program is said to learn from ***experience E (经验)*** with respect to some class of ***tasks T (任务)*** and ***performance measure P (性能)***, if its performance at tasks in T, as measured by P, improves with experience E"
- Example: playing checkers.

  E = the experience of playing many games of checkers

  T = the task of playing checkers.

  P = the probability that the program will win the next game
- In general, any machine learning problem can be assigned to one of two broad classifications: **Supervised learning (监督学习)** and **Unsupervised learning (无监督学习)**

Difference between Data Science, Machine Learning, and Artificial Intelligence

- Data Science produces **Insights**;
- Machine Learning produces **Predictions**;
- Artificial Intelligence produces **Actions**.

- David Robinson

- 数据科学产生洞见
- 机器学习产生预测
- 人工智能产生行动

Data Science - Insights

- Statistical Inference 统计推断
- Data Visualization 数据可视化
- Experiment Design 实验设计
- Domain Knowledge 领域知识
- Communication 沟通

Data scientists may use simple tools or very complex method to analyze trillions of data. **The goal is to gain a better understanding of their data.**

Machine Learning - Predictions

- Given instance X with particular features, predict Y about it.
- Overlap between data science and machine learning
  - Logistic regression can be used to draw insights about relationship (the richer a user is the more likely they will buy our products, so we should change our marketing strategy.)

Artificial Intelligence - Actions

- Game-playing algorithms (Deep Blue, AlphaGo)
- Robotics and control theory (motion planning, walking a bipedal robot)
- Optimization (Google Maps choosing a route)
- Natural language processing
- Reinforcement learning

- **游戏算法 (深蓝，AlphaGo)**
- **机器人和控制理论 (运动规划，行走双足机器人)**
- **优化算法 (Google 地图选择路线)**
- **自然语言处理**
- **强化学习**

How would the three be used together?

Example: Self-Driving Car

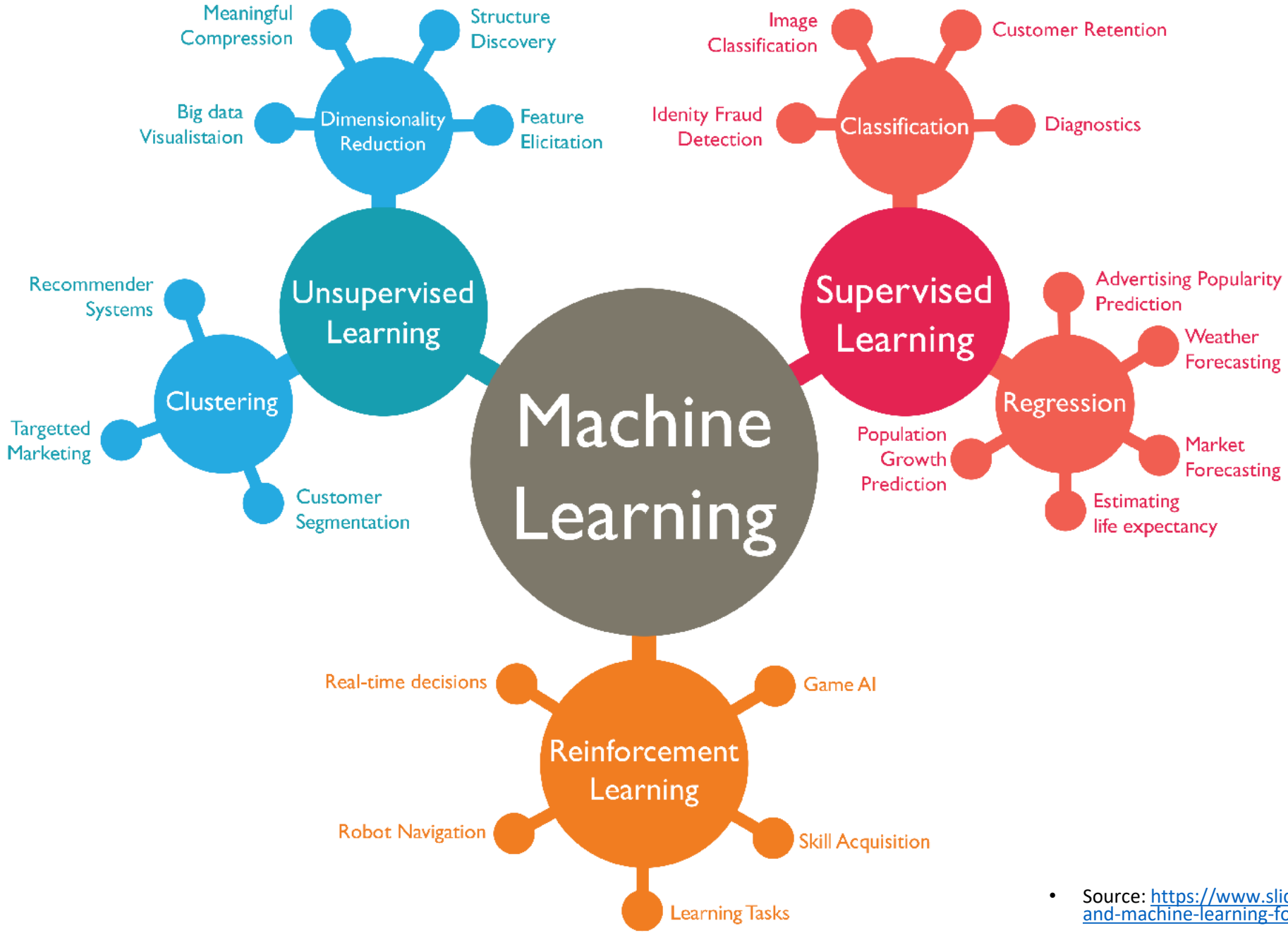
- **Machine learning**: The car has to recognize a stop sign using its cameras. We construct a dataset of millions of photos of streetside objects, and train an algorithm to predict which have stop signs.
- **Artificial intelligence**: Once our car can recognize stop signs, it needs to decide when to take the action of applying the brakes. It's dangerous to apply them too early or too late, and we need it to handle varying road conditions (for example, to recognize on a slippery road that it's not slowing down quickly enough), which is a problem of **control theory**.

How would the three be used together?

- **Data science**: In street tests we find that the car's performance isn't good enough, with some **false negatives** in which it drives right by a stop sign. After analyzing the street test data, we gain the **insight** that the rate of false negatives depends on the time of day: it's more likely to miss a stop sign before sunrise or after sunset. We realize that most of our training data included only objects in full daylight, so we construct a better dataset including nighttime images and go back to the machine learning step.

Machine Learning Problems

	Supervised Learning	Unsupervised Learning
Discrete	**Classification** Nearest Neighbor, Naive Bayes, Decision Trees, Classification Rule Learners Artificial Neural Network, Support Vector Machine	**Clustering** k-means clustering
Continuous	**Numeric Prediction** Linear Regression, Regression Trees, Model Trees	**Dimensionality Reduction**



- Source: https://www.slideshare.net/awahid/big-data-and-machine-learning-for-businesses

Classification vs. Clustering

- **Classification**: have a set of predefined classes (training set) and want to know which class a new object belongs to
- **Clustering**: try to group a set of objects and find whether there is some relationship between the objects

Supervised Classification	**Unsupervised Clustering**
• Known number of classes • Based on a training set • Used to classify future observations • Algorithms: Nearest Neighbor, Decision Tree	• Unknown number of classes • No prior knowledge • Used to understand (explore) data • Algorithms: K-means, Expectation Maximization

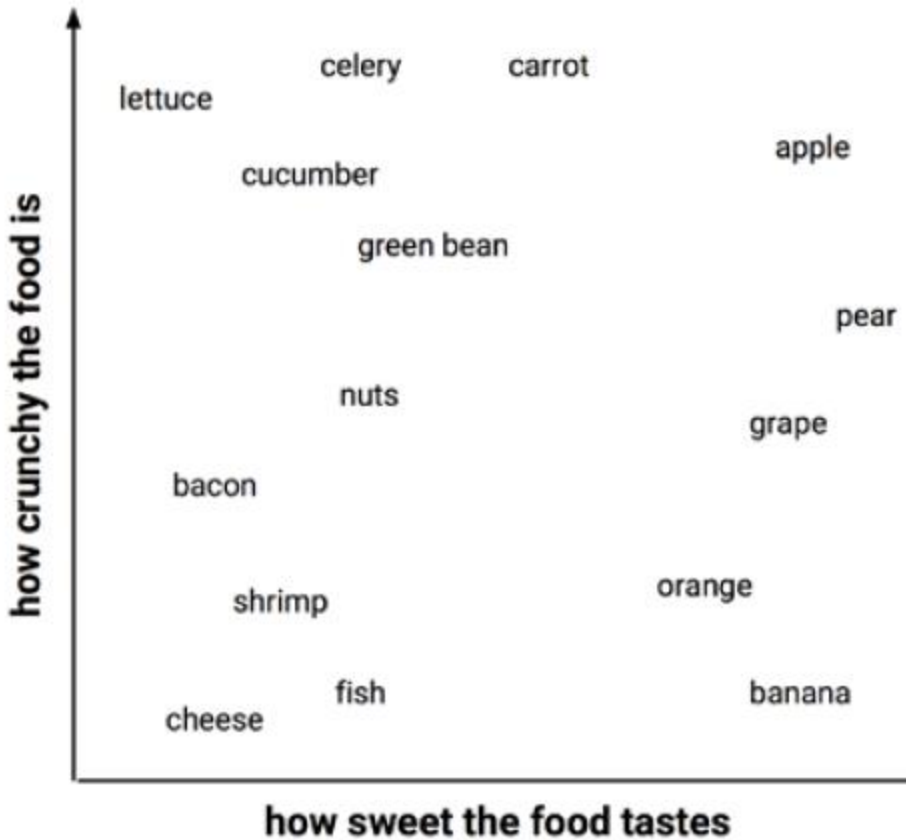
	Supervised Learning	Unsupervised Learning
Discrete	**Classification** **Nearest Neighbor**, Naive Bayes, Decision Trees, Classification Rule Learners Artificial Neural Network, Support Vector Machine	**Clustering** k-means clustering
Continuous	**Numeric Prediction** Linear Regression, Regression Trees, Model Trees	**Dimensionality Reduction**

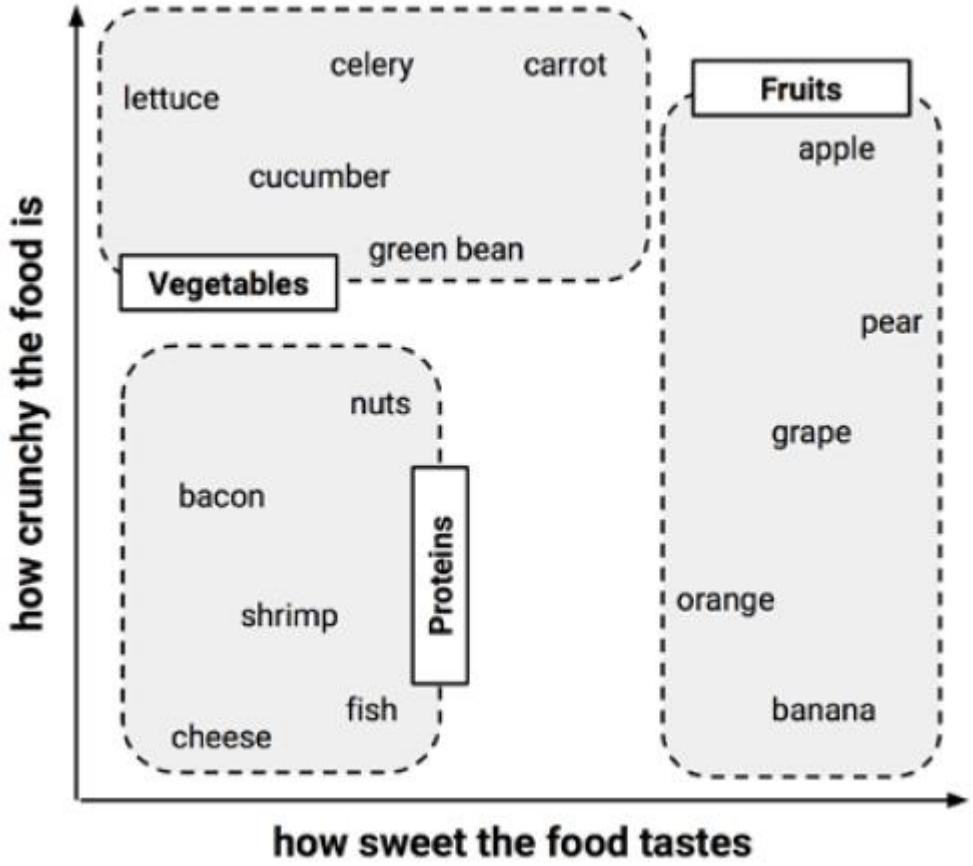
k-NN Algorithm k-近邻算法

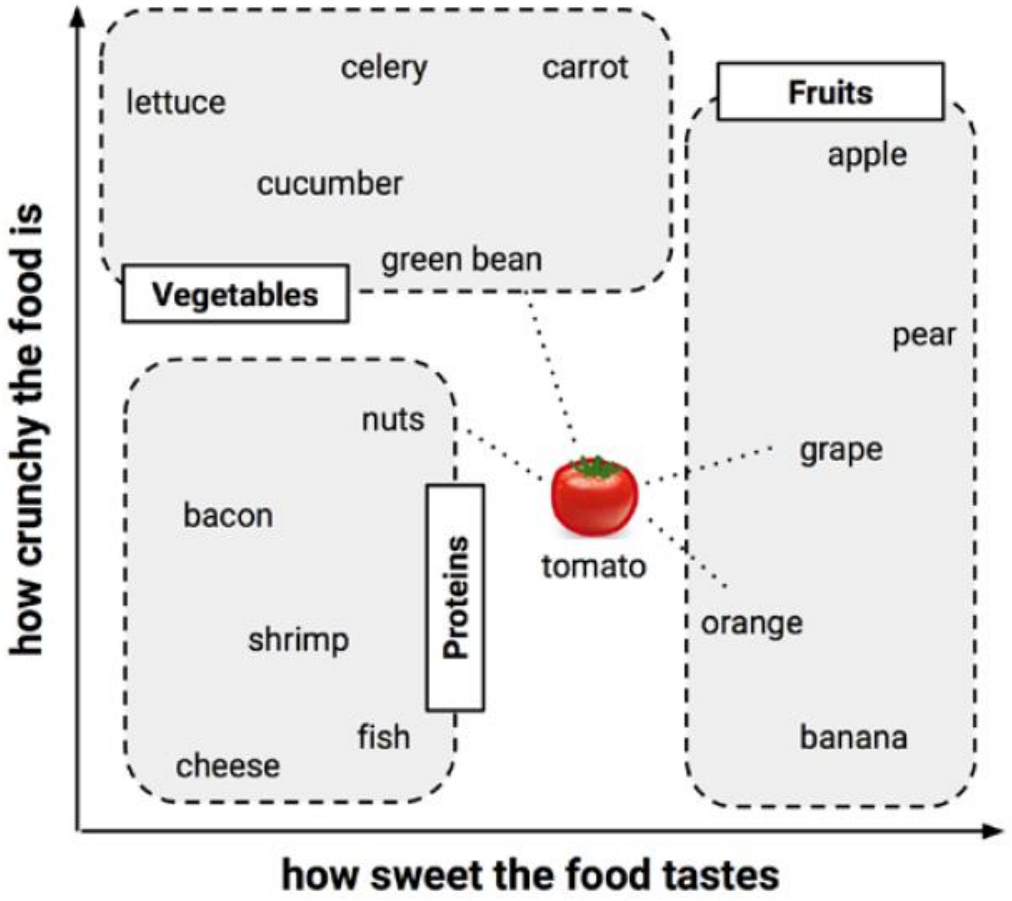
- The nearest neighbors approach to **classification** is exemplified by the **k-nearest neighbors** algorithm (k-NN)
- **k** is a variable term implying that **any number of nearest neighbors** could be used
- The algorithm requires a training dataset made up of examples that have been classified into several categories, as labeled by a nominal variable
- Measuring similarity with distance - **Euclidean Distance**
- Transformation – **normalizing** numeric data

k-NN Example

Ingredient	Sweetness	Crunchiness	Food type
apple	10	9	fruit
bacon	1	4	protein
banana	10	1	fruit
carrot	7	10	vegetable
celery	3	10	vegetable
cheese	1	1	protein


how crunchy the food is
how sweet the food tastes
lettuce
celery
carrot
cucumber
apple
green bean
pear
nuts
grape
bacon
shrimp
orange
fish
banana
cheese


how crunchy the food is
how sweet the food tastes
lettuce
celery
carrot
cucumber
green bean
Vegetables
Fruits
apple
pear
grape
orange
banana
nuts
bacon
shrimp
fish
cheese
Proteins



Tomorrow: Sweetness = 6, Crunchiness = 4

k=1, orange -> fruit

k=3, orange, grape, nuts -> fruit

Ingredient	Sweetness	Crunchiness	Food type	Distance to the tomato
grape	8	5	fruit	*sqrt((6 - 8)^2 + (4 - 5)^2) = 2.2*
green bean	3	7	vegetable	*sqrt((6 - 3)^2 + (4 - 7)^2) = 4.2*
nuts	3	6	protein	*sqrt((6 - 3)^2 + (4 - 6)^2) = 3.6*
orange	7	3	fruit	*sqrt((6 - 7)^2 + (4 - 3)^2) = 1.4*

Normalizing Numeric Data

- Add an additional feature for food's spiciness (0 – over 1 million)
- Impact distance function more than other two factors
- Use min-max normalization:

$$X_{new} = \frac{X - \min(X)}{\max(X) - \min(X)}$$

- Or z-score standardization:

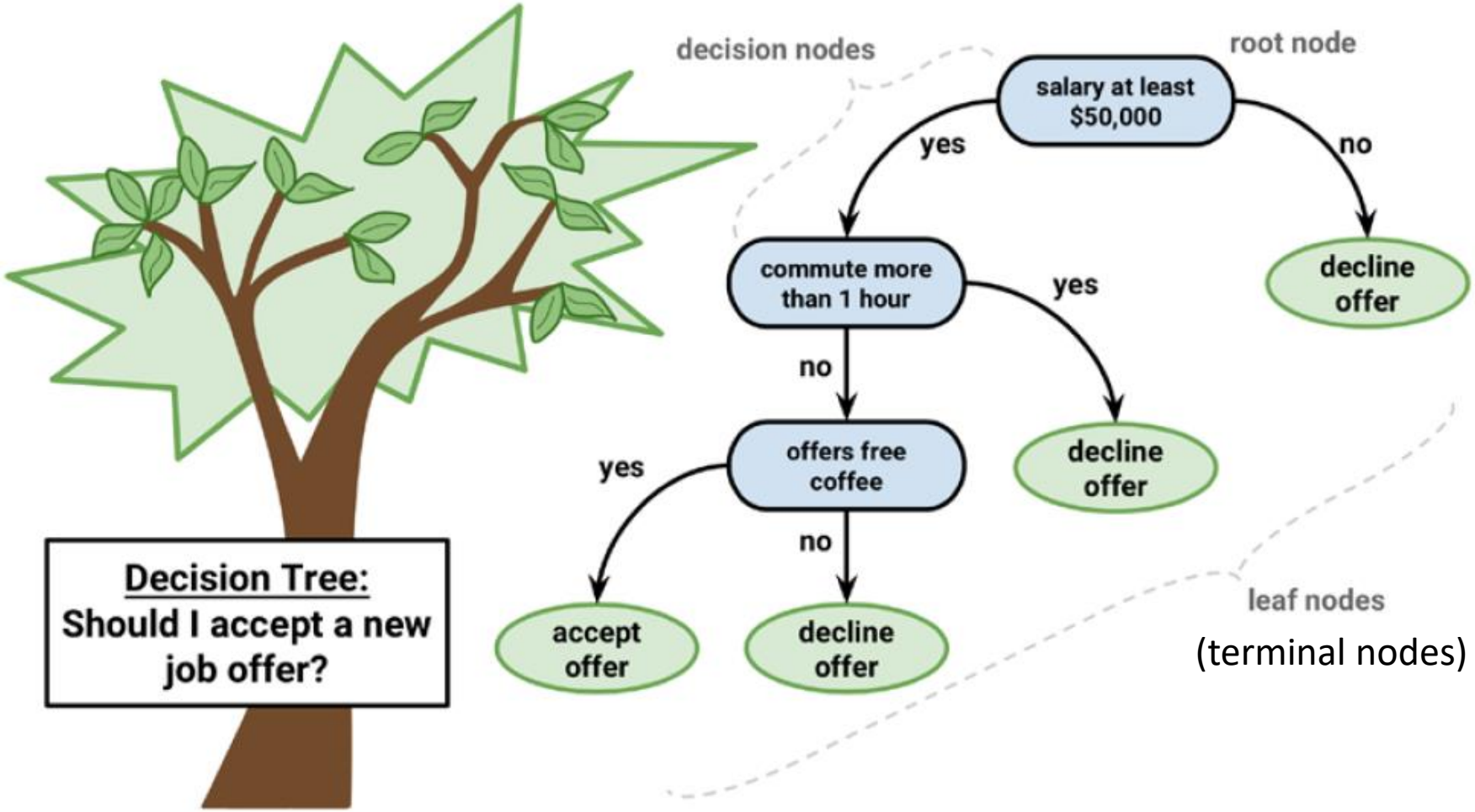
$$X_{new} = \frac{X - \mu}{\sigma} = \frac{X - \text{Mean}(X)}{\text{StdDev}(X)}$$

Application of k-NN in Transportation

- A Short-Term Traffic Flow Forecasting Method Based on a Three-Layer K-Nearest Neighbor Non-Parametric Regression Algorithm (Wang, et al., 2016)
- Efficient K-Nearest Neighbor Search in Time-Dependent Spatial Network (Demiryurek et al., 2011)
- Short-Term Traffic Volume forecasting: A k-Nearest Neighbor Approach Enhanced by Constrained Linearly Sewing Principle Component Algorithm (Zheng, et al., 2014)
- Transportation Modes Classification Using Sensors on Smartphones – using KNN, DT, and SVM (Fang, et al., 2016)

	Supervised Learning	Unsupervised Learning
Discrete	**Classification** Nearest Neighbor, **Decision Trees**, Naive Bayes, Classification Rule Learners Artificial Neural Network, Support Vector Machine	**Clustering** k-means clustering
Continuous	**Numeric Prediction** Linear Regression, Regression Trees, Model Trees	**Dimensionality Reduction**

Decision Trees 决策树



Decision Trees Algorithm

- Decision trees are built using a heuristic called **recursive partitioning (递归划分)**. This approach is also commonly known as **divide and conquer** because it splits the data into subsets, which are then split repeatedly into even smaller subsets, and so on and so forth until the process stops when the algorithm determines the data within the subsets are sufficiently **homogenous (一致性)**

Decision Trees Algorithm

- **C5.0** algorithm - industry standard to produce decision trees: developed by J. Ross Quinlan as an improved version of his prior algorithm, **C4.5**, which itself is an improvement over his **Iterative Dichotomiser 3** (**ID3**) algorithm

C5.0 Algorithm

- The first challenge of a decision tree is to identify **which feature** to split upon
- Goal: split the data such that the resulting partitions contained examples primarily of **a single class** => measure **Purity** (**纯度**)
- Measurement of purity in C5.0: **Entropy**

$$\text{Entropy}(S) = \sum_{i=1}^{c} -p_i \, log_2(p_i)$$

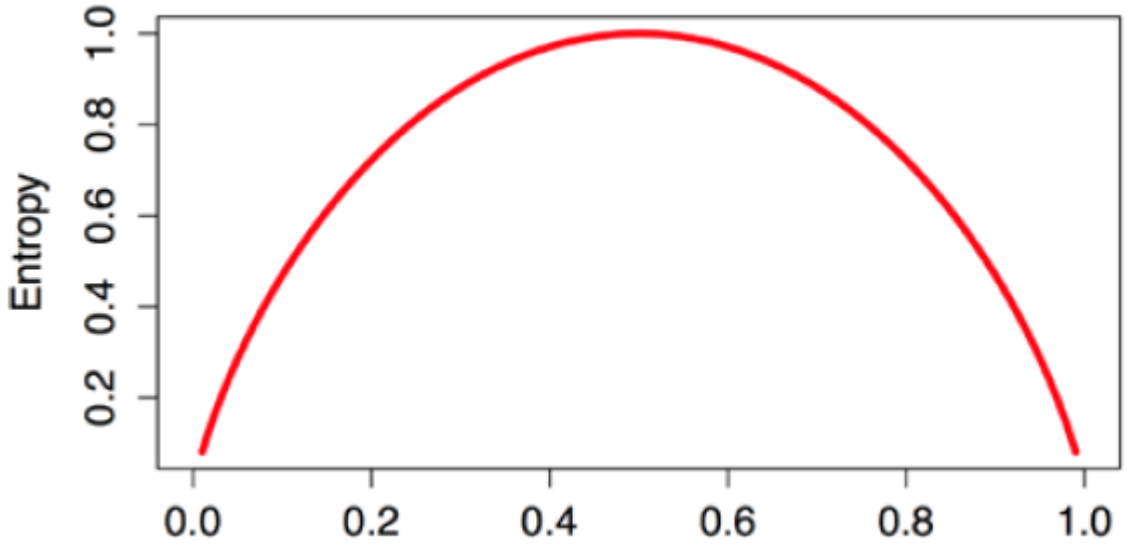
* A concept from information theory that quantifies the **randomness, or disorder**, within a set of class values

given segment of data (*S*), *c* refers to the number of class levels and p_i refers to the proportion of values falling into class level *i*

- Decision tree hopes to find splits that **reduce entropy**, ultimately increasing **homogeneity** within the groups.

C5.0 Algorithm

Two-Class Example




```
curve(-x * log2(x) - (1 - x) * log2(1 - x),
      col = "red", xlab = "x", ylab = "Entropy", lwd = 4)
```

- **Information Gain** (信息增益)

$$\text{InfoGain}(F) = \text{Entropy}(S_1) - \text{Entropy}(S_2)$$

before – after

$$\text{Entropy}(S) = \sum_{i=1}^{n} w_i \, \text{Entropy}(P_i)$$

C5.0 Algorithm

- Information gain is not the only splitting criterion that can be used to build decision trees.
- Other common criteria are **Gini index** (基尼系数), **Chi-Squared statistic** (卡方统计量), and **gain ratio** (增益比)

C5.0 Algorithm

Strength

- An all-purpose classifier that does well on most problems
- Highly automatic learning process, which can handle numeric or nominal features, as well as missing data
- Excludes unimportant features
- Can be used on both small and large datasets
- Results in a model that can be interpreted without a mathematical background (for relatively small trees)
- More efficient than other complex models

Weakness

- Decision tree models are often biased toward splits on features having a large number of levels
- It is easy to overfit or underfit the model
- Can have trouble modeling some relationships due to reliance on axis-parallel splits
- Small changes in the training data can result in large changes to decision logic
- Large trees can be difficult to interpret and the decisions they make may seem counterintuitive

Application of Decision Tree in Transportation

- Predicting Short-Term Subway Ridership and Prioritizing Its Influential Factors Using Gradient Boosting Decision Trees (Ding, et al., 2016)
- Traffic Accident Analysis Using Decision Trees and Neural Networks (Chong, et al., 2014)
- Using Decision Tree Induction Systems for Modeling Space-Time Behavior (Arentze, et al., 2010)
- Analyzing Transit Service Quality Evolution Using Decision Trees and Gender Segmentation (Ona, et al., 2013)

	Supervised Learning	Unsupervised Learning
Discrete	**Classification or Categorization** Nearest Neighbor, Naive Bayes, Decision Trees, Classification Rule Learners	**Clustering** k-means clustering
Continuous	**Numeric Prediction** **Linear Regression**, Regression Trees, Model Trees	**Dimensionality Reduction**

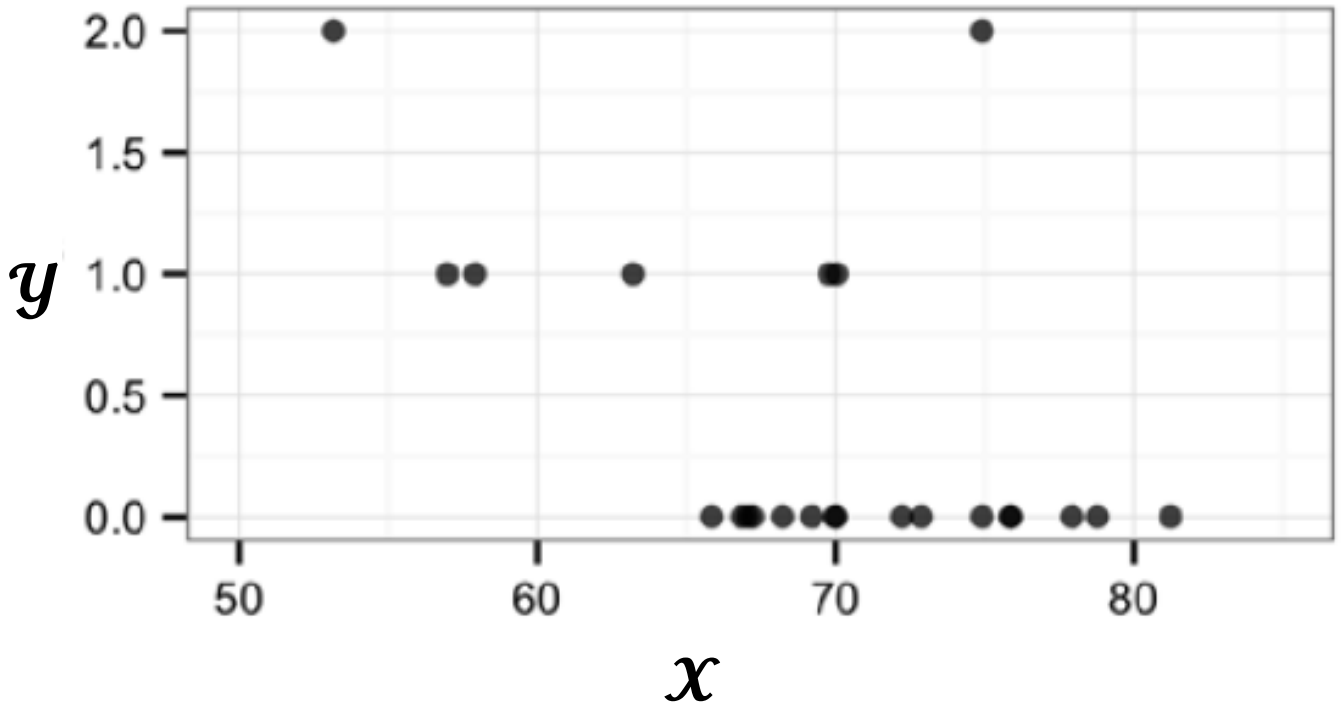
Regression Methods 回归方法

- **Simple linear regression**: a single independent variable
- **Multiple linear regression**: two or more independent variables
- **Logistic regression**: model a binary categorical outcome
- **Poisson regression**: models integer count data
- **Multinomial logistic regression**: models a categorical outcome

Simple Linear Regression

$$y = \alpha + \beta x$$

- Estimate α & β 参数估计



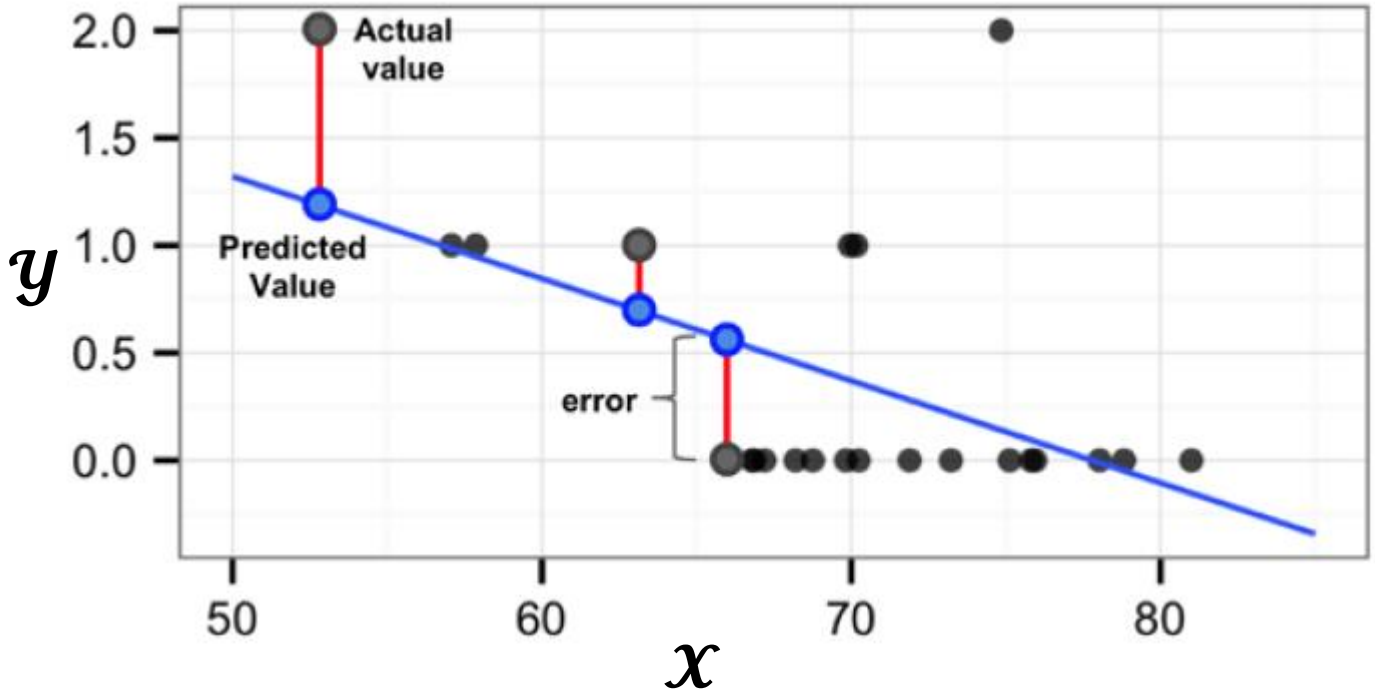
Ordinary Least Squares Estimation 最小二乘

$$y = \alpha + \beta x$$

$$\sum (y_i - \hat{y}_i)^2 = \sum e_i^2$$

$$\alpha = \bar{y} - \beta\,\bar{x}$$

$$\beta = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2}$$



Correlation

$$\mathrm{Cov}(x,y) = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{n}$$

Covariance 协方差

$$\rho_{x,y} = \mathrm{Corr}(x,y) = \frac{\mathrm{Cov}(x,y)}{\sigma_x \sigma_y}$$

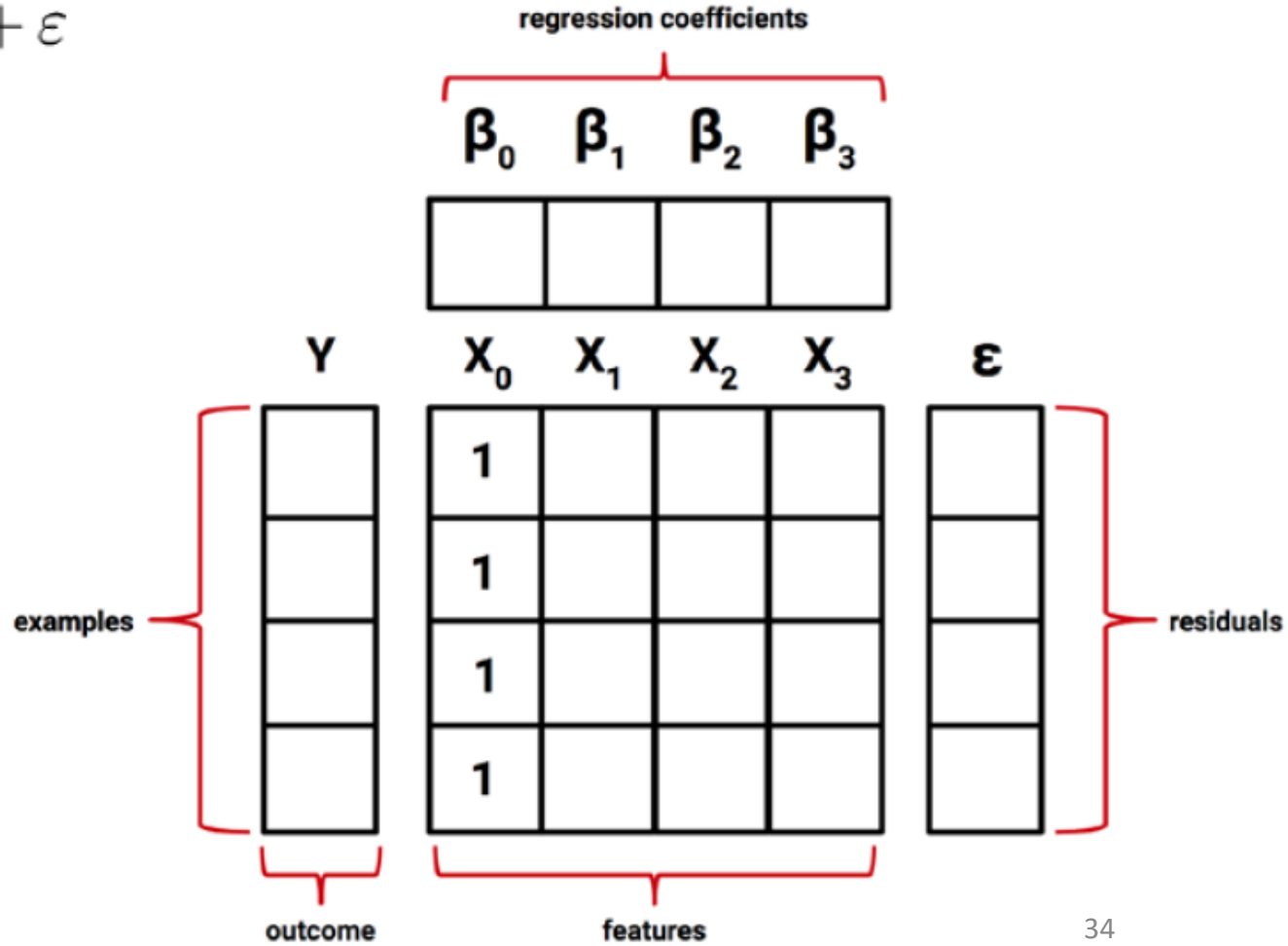
Pearson's Correlation Coefficient 相关系数

Multiple Linear Regression

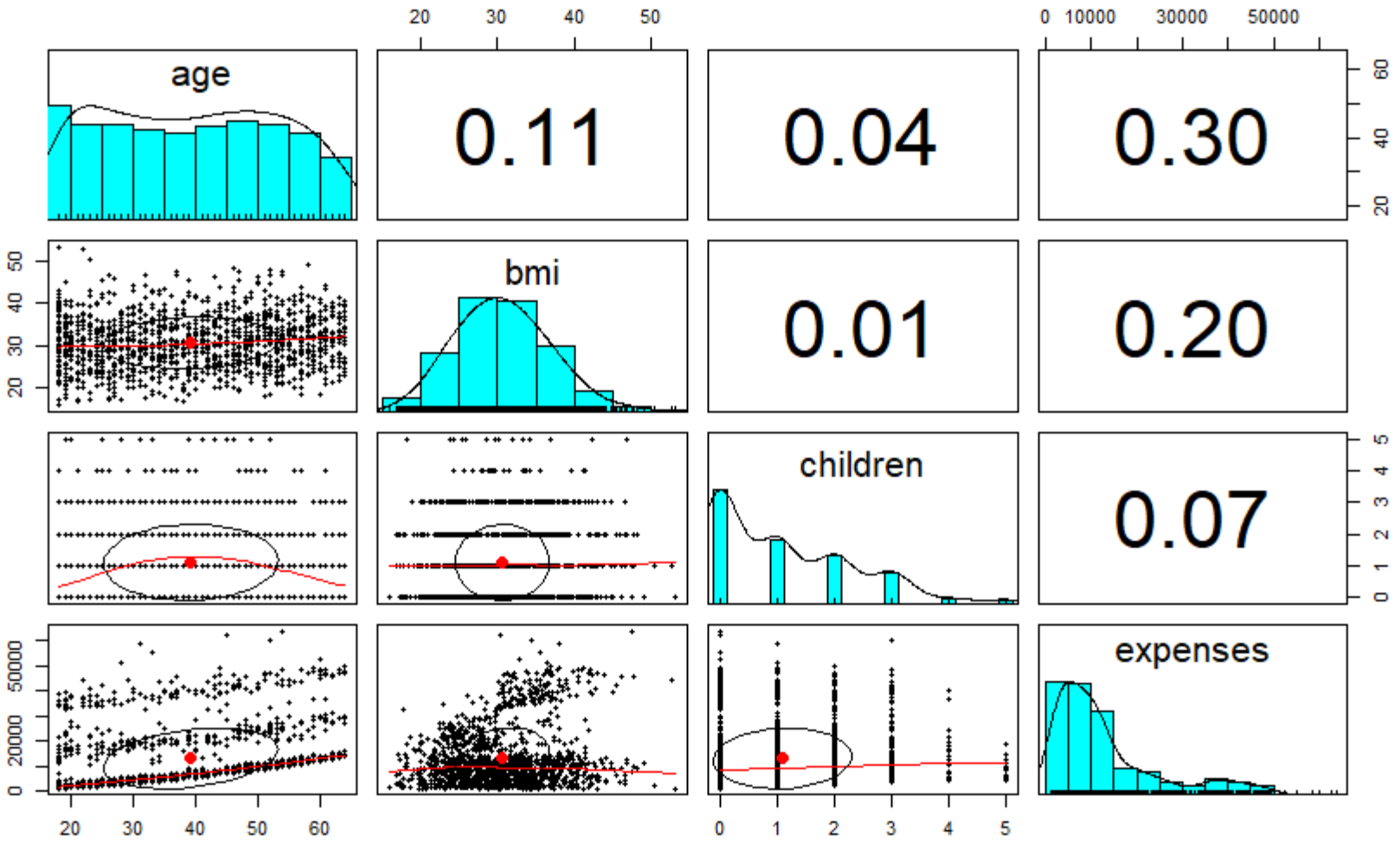
$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_i x_i + \varepsilon$$

$$\mathbf{Y} = \boldsymbol{\beta}\mathbf{X} + \boldsymbol{\varepsilon}$$

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\mathbf{T}\mathbf{X})^{-1}\mathbf{X}^\mathbf{T}\mathbf{Y}$$



Correlation Example in Regression



correlation ellipse

Regression Trees and Model Trees 回归树与模型树

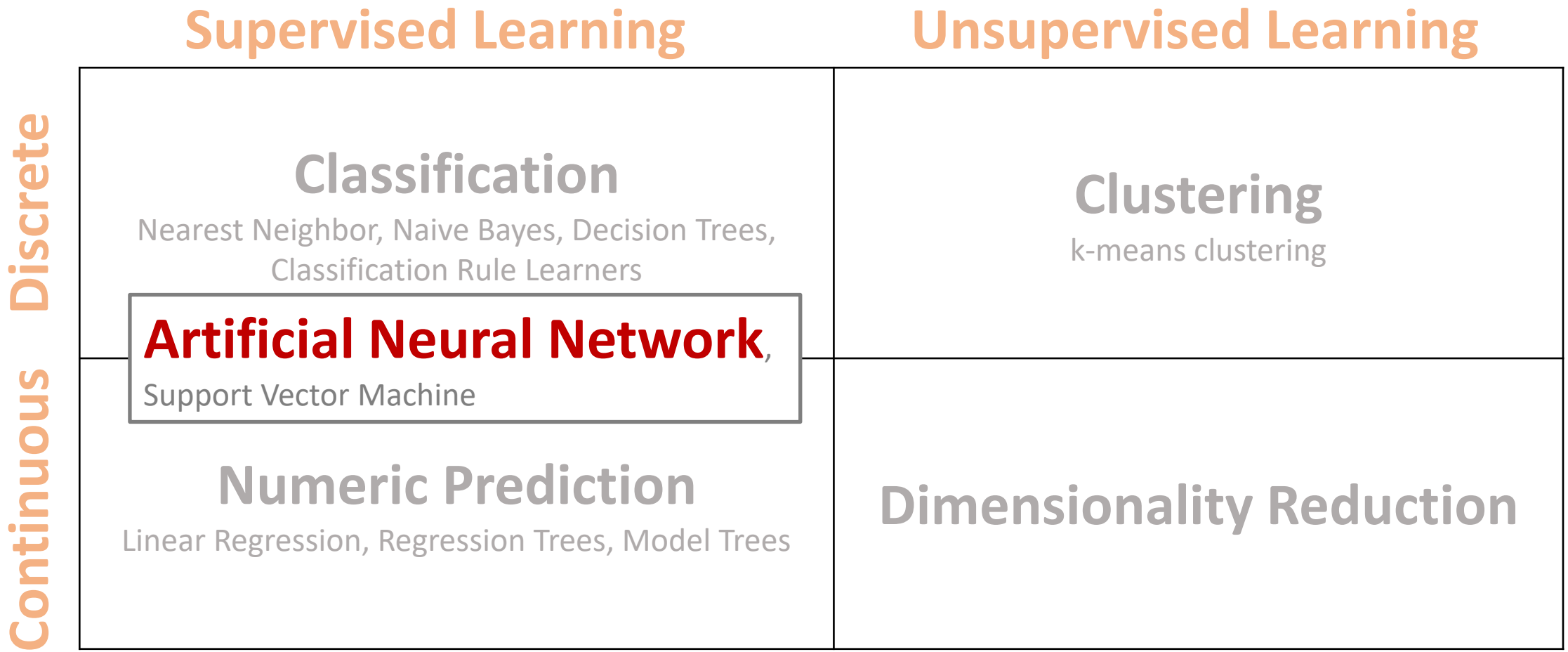
- Applying trees used for numeric prediction differs from trees used for classification
- Trees for numeric prediction fall into two categories:
- The first: known as **Regression Trees**, were introduced in the 1980s as part of the **Classification and Regression Tree** (CART) algorithm
- Regression trees **do not** use linear regression methods
- The second: **Model Trees**. Lesser-known but perhaps more powerful
- Model trees are grown in much the same way as regression trees, but at each leaf, a **multiple linear regression model** is built from the examples reaching that **node**

Adding Regression to Trees

Strengths	Weaknesses
• Combines the strengths of decision trees with the ability to model numeric data • Does not require the user to specify the model in advance • Uses automatic feature selection, which allows the approach to be used with a very large number of features • May fit some types of data much better than linear regression • Does not require knowledge of statistics to interpret the model	• Not as well-known as linear regression • Requires a large amount of training data • Difficult to determine the overall net effect of individual features on the outcome • Large trees can become more difficult to interpret than a regression model

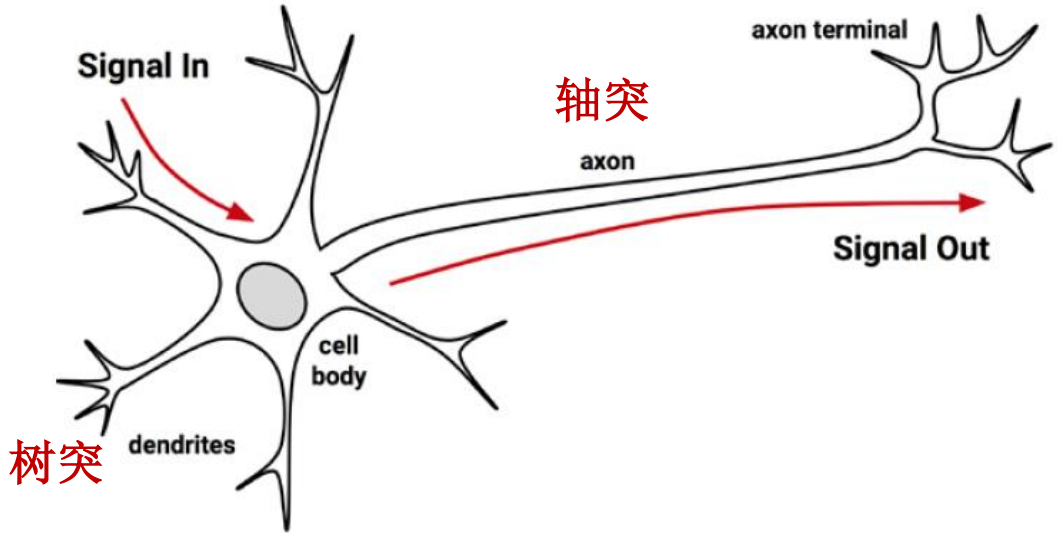
Application of Regression Trees in Transportation

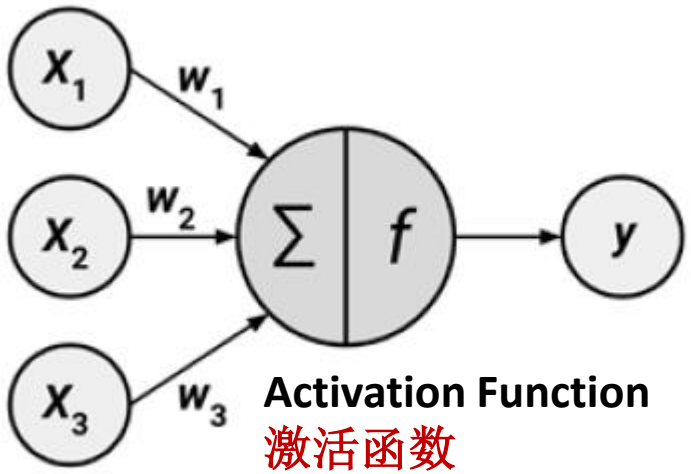
- Evaluation of the Gradient Boosting of Regression Trees Method on Estimating Car-Following behavior (Dabiri, et al., 2018)
- Predicting Human-Driving Behavior to Help Driverless Vehicles Drive: Random Intercept Bayesian Additive Regression Trees (Tan, et al., 2017)
- Prediction of Pedal Cyclists and Pedestrian Fatalities from Total Monthly Accidents And Registered Private Car Numbers Using Regression Trees (Ghasemlou, et al., 2015)
- Vehicular Emissions Prediction with CART-BMNARS Hybrid Models Using Regression Trees (Oduro, et al., 2016)


Supervised Learning
Unsupervised Learning
Discrete
Continuous
Classification
Nearest Neighbor, Naive Bayes, Decision Trees, Classification Rule Learners
Artificial Neural Network,
Support Vector Machine
Clustering
k-means clustering
Numeric Prediction
Linear Regression, Regression Trees, Model Trees
Dimensionality Reduction

Artificial Neural Network (ANN) 人工神经网络

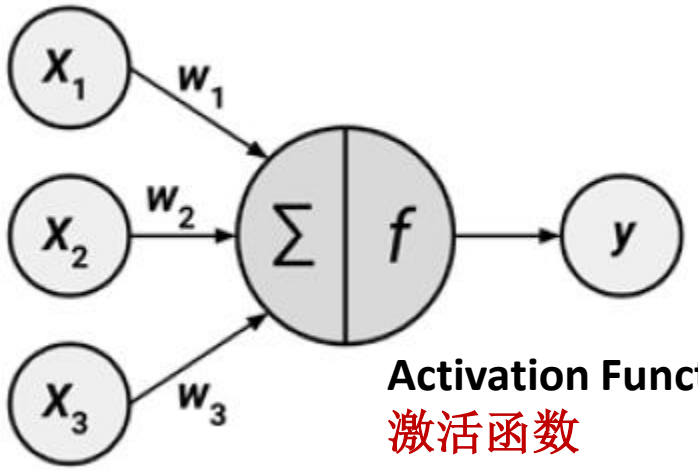
- ANN models the relationship between a set of input signals and an output signal using a model derived from a biological brain responds to stimuli from sensory inputs
- A brain uses a network of interconnected cells called **neurons** (神经元) to create a massive parallel processor, ANN uses a network of **artificial neurons** or **nodes** (节点) to solve learning problems





Artificial Neural Network

$$y(x) = f\left(\sum_{i=1}^{n} w_i x_i\right)$$



Activation Function
激活函数

x_i : signal input received by the dendrites

w_i : weights allow each of the n inputs to contribute a greater or lesser amount to the sum of input signals

$y(x)$: signal output

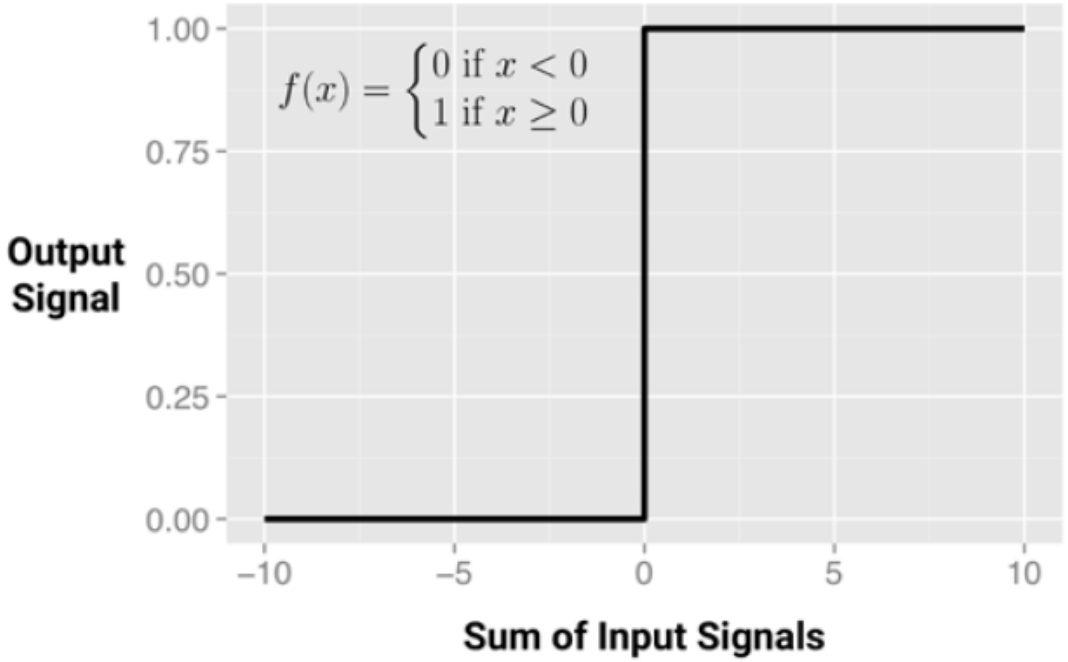
ANN can be defined in terms of the following characteristics:

- Activation Function 激活函数
- Network Topology 网络拓扑
- Training Algorithm 训练算法

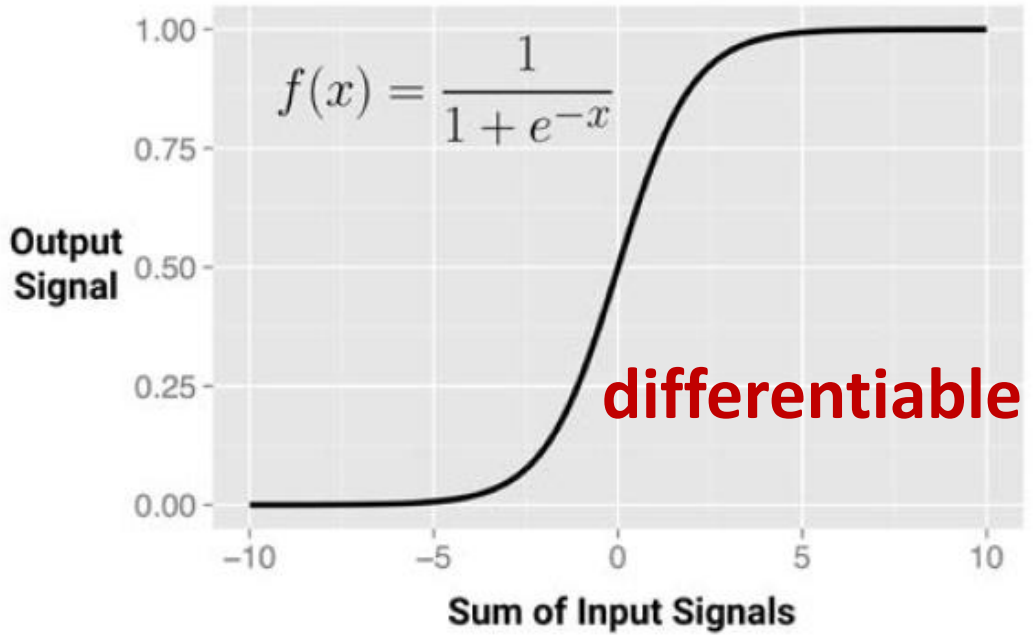
Activation Function

Threshold Activation Function (阈值激活函数)：if sum of total input signals meet the firing threshold, the neuron passes on the signal; otherwise, it does nothing

- Unit Step Activation Function

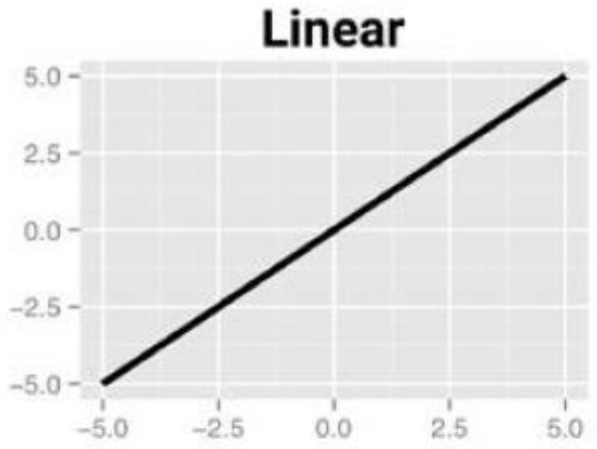


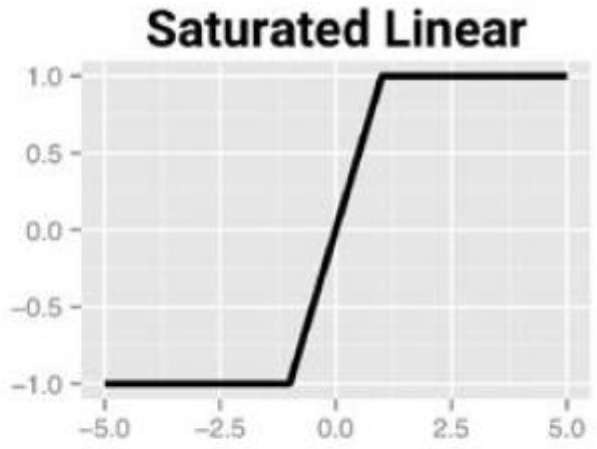
- Sigmoid Activation Function

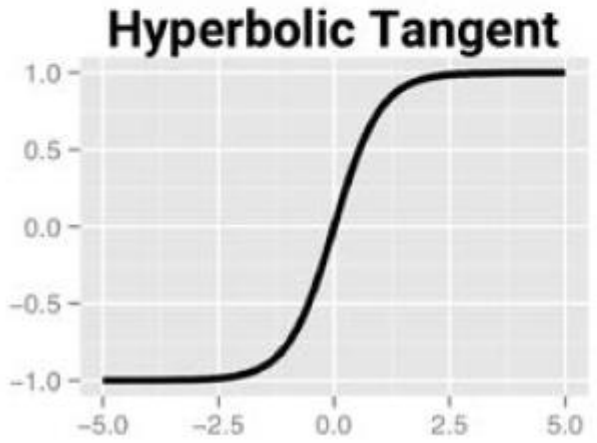


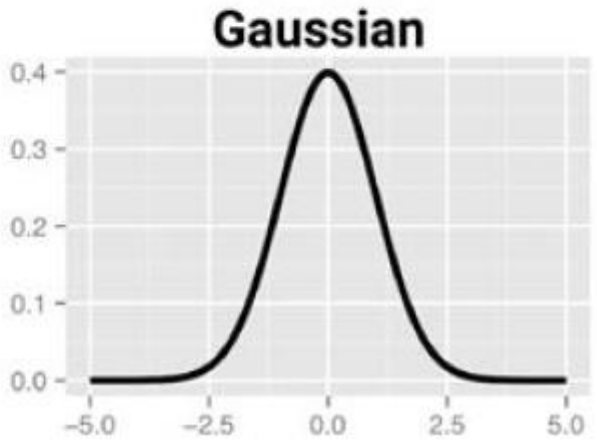
Activation Function

- Output signal range is from (0, 1), (-1, +1), or (-∞, +∞). The choice of activation function biases the neural network
- Certain types of data may fit specialized neural networks
- A **linear activation function** results in a neural network very similar to a **linear regression model**
- a **Gaussian activation function** results in a model called a **Radial Basis Function** (**RBF**) network









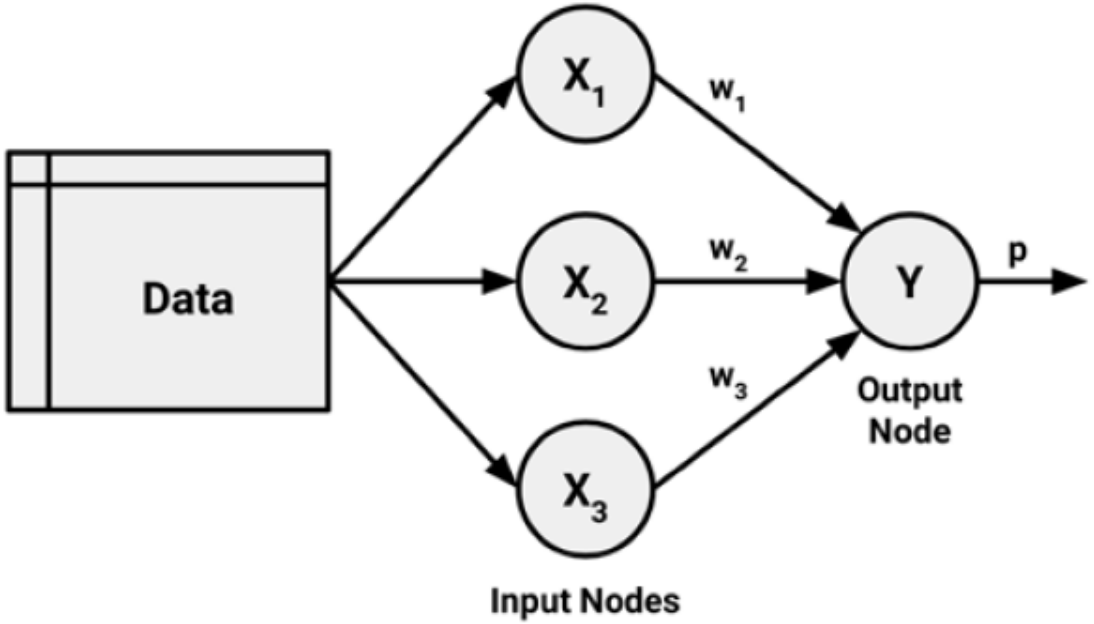
Activation Function

- The **range of input values** that affect the output signal **is relatively narrow**
- For example, in the case of sigmoid, the output signal is always nearly 0 or 1 for an input signal below -5 or above +5, respectively
- This essentially squeezes the input values into a smaller range of outputs, activation functions like the sigmoid are sometimes called **Squashing Functions (压缩函数)**
- Solutions: transform all neural network inputs within a small range around 0, involving **standardizing** or **normalizing** the features
- Prevent large valued features such as **household income** from dominating small-valued features such as the **number of children in the household**

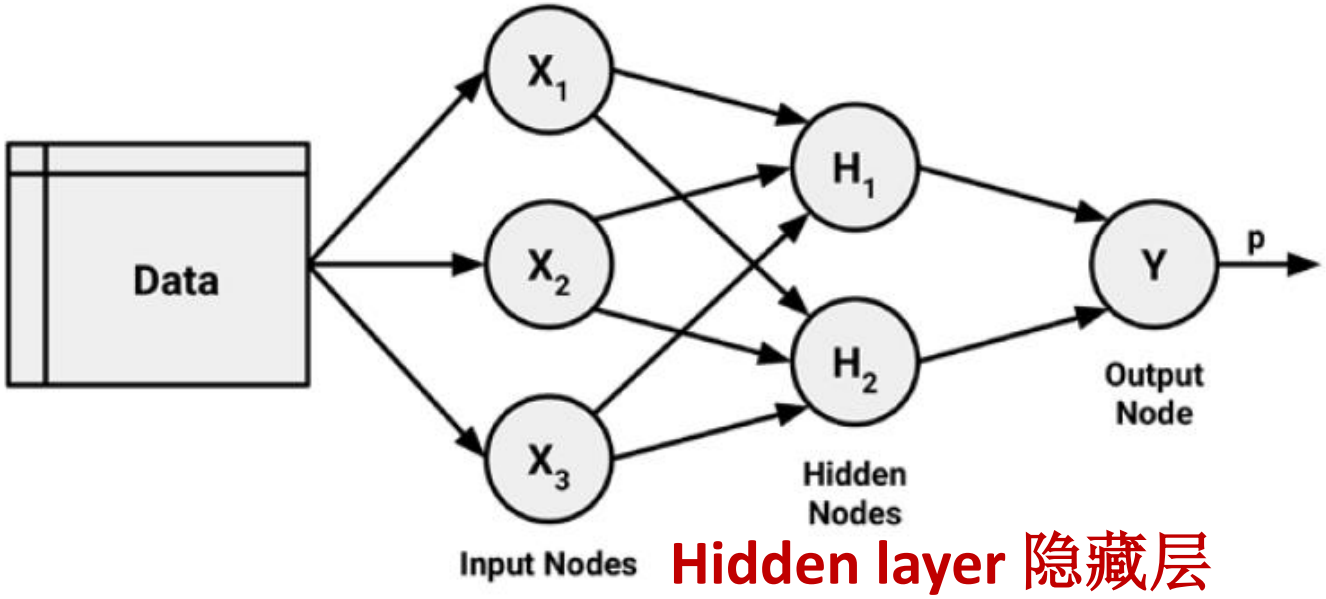
Network Topology

- **Number of Layers**

Single-layer Network 单层网络



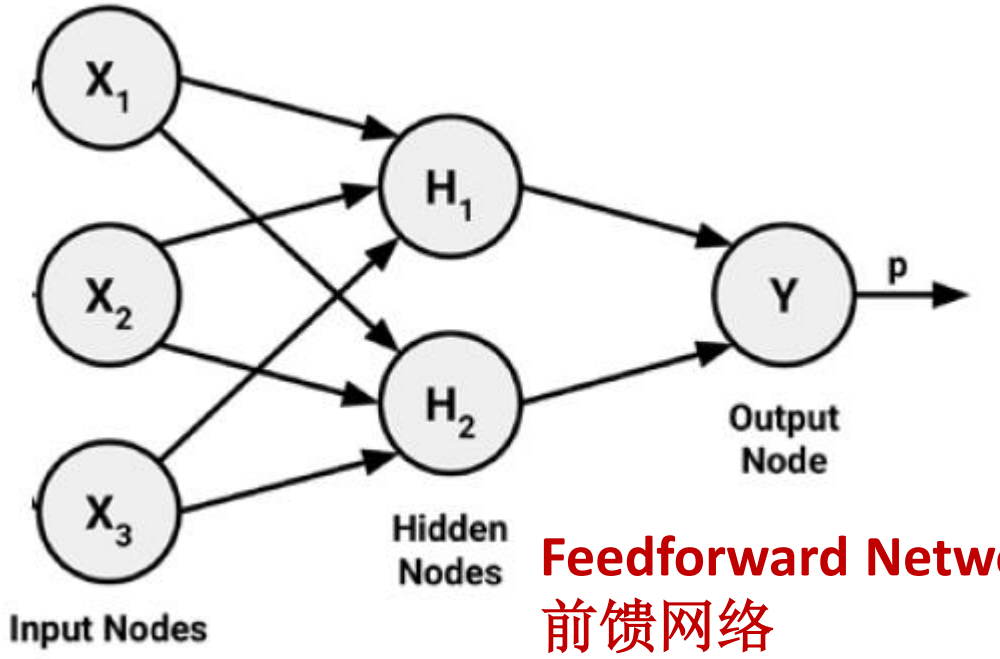
Multi-layer Network多层网络



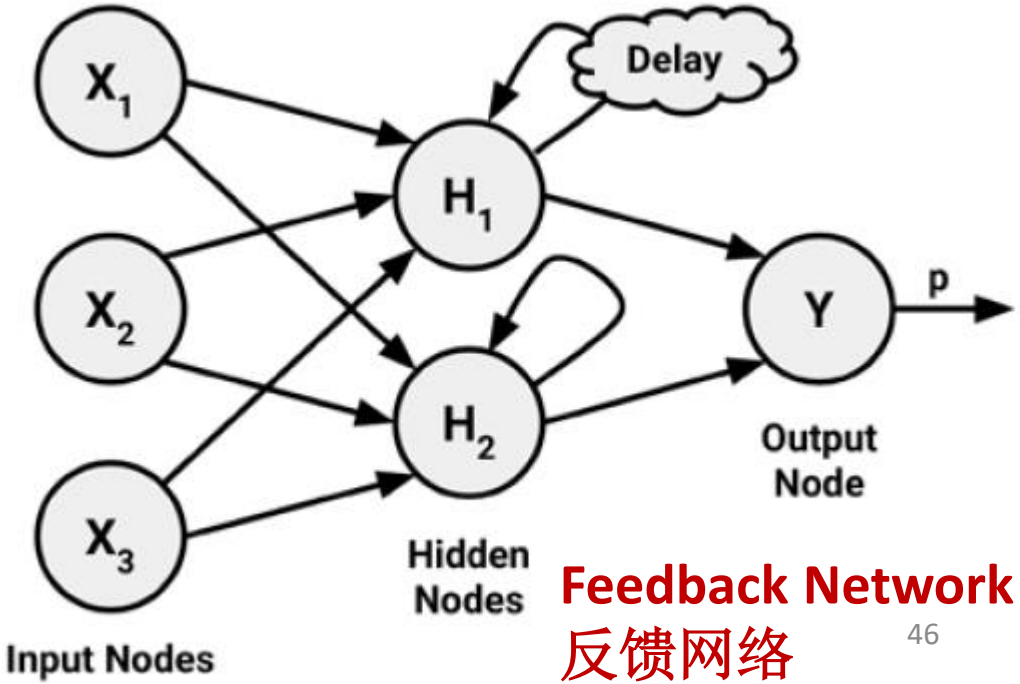
Hidden layer 隐藏层

Network Topology

- **Direction of Information Travel**
- Feedforward Networks vs. Recurrent (Feedback) Network
- **Deep Neural Network (DNN)**: A neural network with multiple hidden layers; training such network is referred to as **Deep Learning 深度学习**
- **Multi-layer Feedforward Network:** called as **Multi-layer Perception (MLP) 多层感知器**



Feedforward Networks
前馈网络



Feedback Network
反馈网络

Training Algorithm - Backpropagation 后向传播

- Backpropagation algorithm iterates through many cycles of two processes. Each cycle is known as an **epoch (时段)**.
- **Starting weights (w_i)** are typically set **at random**. Then the algorithm iterates through the processes, until a stopping criterion is reached.

Each epoch in the backpropagation algorithm includes:

- **Forward Phase (前向阶段)**: the neurons are activated in sequence from the input layer to the output layer, applying **each neuron's weight** and activation function along the way. An output signal is produced.
- **Backward phase (后向阶段)**: **output signal** resulting from the forward phase is compared to the **true target value** in the training data. The difference between output signal and true value results in an error that is propagated **backwards** in the network to modify the **connection weights** between neurons and reduce future errors.

Training Algorithm - Backpropagation

- How to modify weights in each epoch? Backpropagation is commonly used by the **Gradient Descent Optimization (梯度下降法)** algorithm to adjust the weight of neurons by calculating the gradient of the loss function
- Gradient descent is a first-order iterative optimization algorithm for finding the minimum of a function

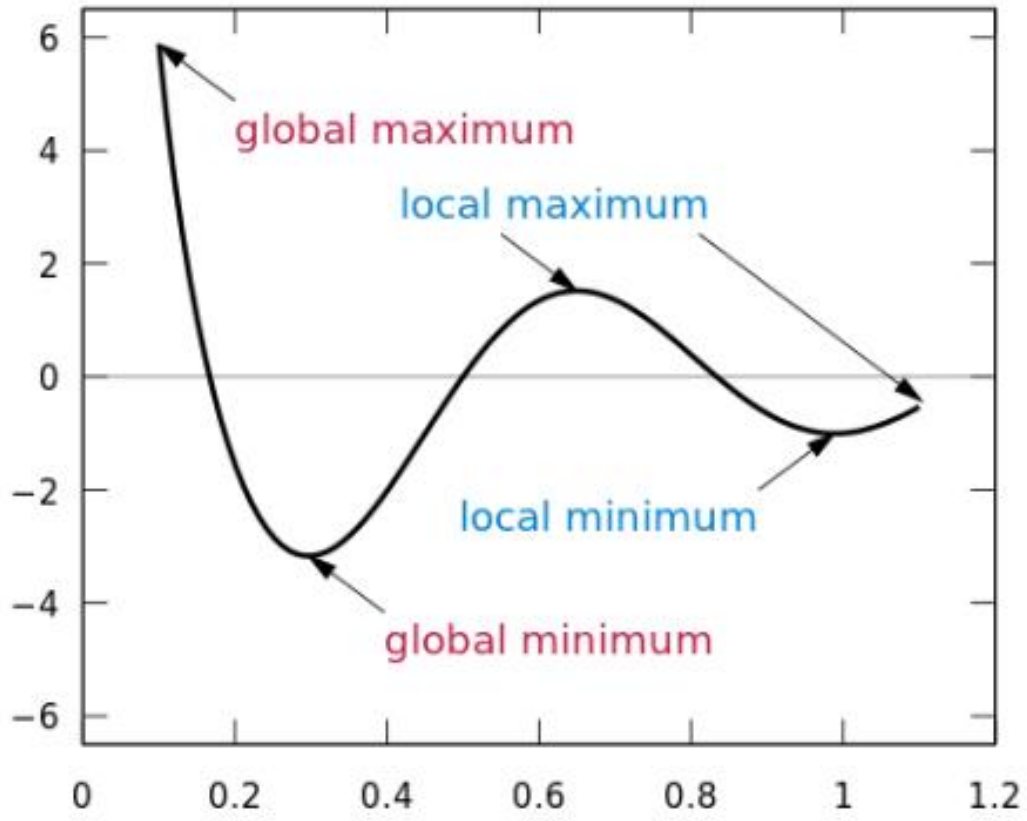
Training Algorithm - Backpropagation

- Multi-layer feedforward networks that use the backpropagation algorithm are now common in the field of data mining

Strengths	Weaknesses
• Can be adapted to classification or numeric prediction problems • Capable of modeling more complex patterns than nearly any algorithm • Makes few assumptions about the data's underlying relationships	• Extremely computationally intensive and slow to train, particularly if the network topology is complex • Very prone to overfitting training data • Results in a complex black box model that is difficult, if not impossible, to interpret

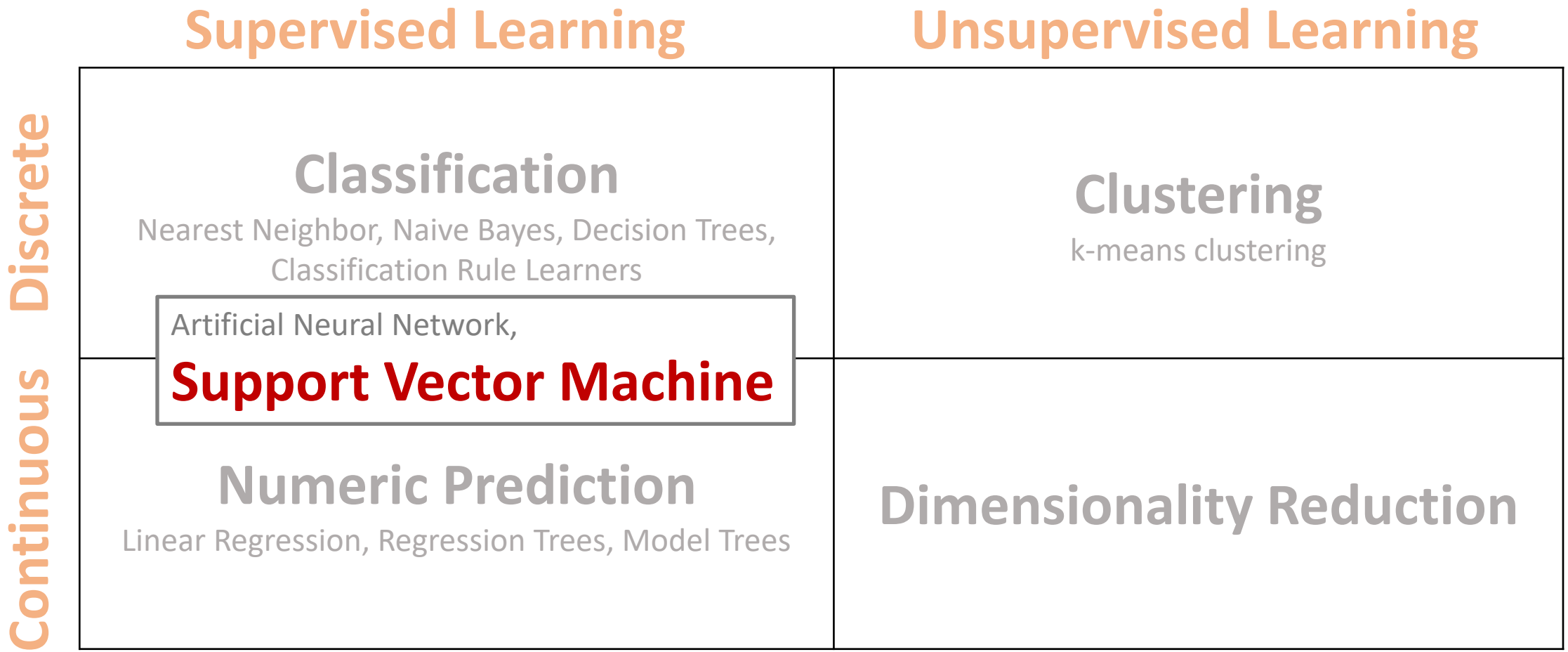
Training Algorithm - Backpropagation

- Another weakness: Gradient descent can find the local minimum instead of the global minimum



Application of ANN in Transportation

- A Comparison of The Performance of ANN and SVM for The Prediction of Traffic Accident Duration (Wang, et al., 2016)
- Development of Artificial Neural Network Models to Predict Driver Injury Severity in Traffic Accidents at Signalized Intersections (Abdelwahab, et al., 2001)
- Incorporating Real-time Traffic and Weather Data to Explore Road Accident Likelihood and Severity in Urban Arterials (Theofilatos, 2016)
- Prediction of Hourly Air Pollutant Concentrations Near Urban Arterials Using Artificial Neural Network Approach(Cai, et al., 2009)

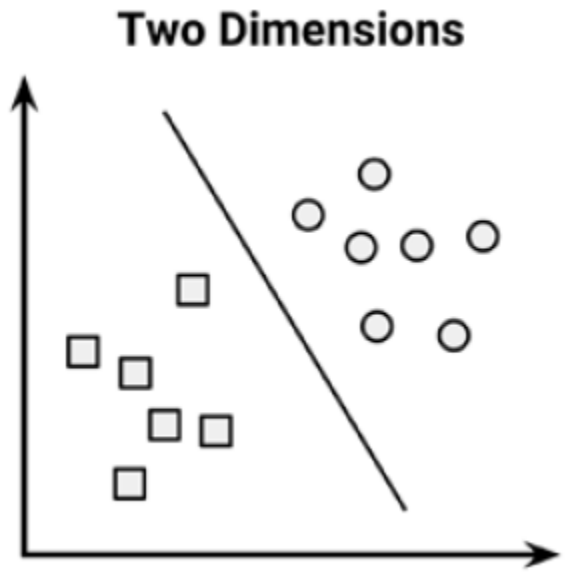

Supervised Learning
Unsupervised Learning
Discrete
Continuous
Classification
Nearest Neighbor, Naive Bayes, Decision Trees, Classification Rule Learners
Artificial Neural Network,
Support Vector Machine
Clustering
k-means clustering
Numeric Prediction
Linear Regression, Regression Trees, Model Trees
Dimensionality Reduction

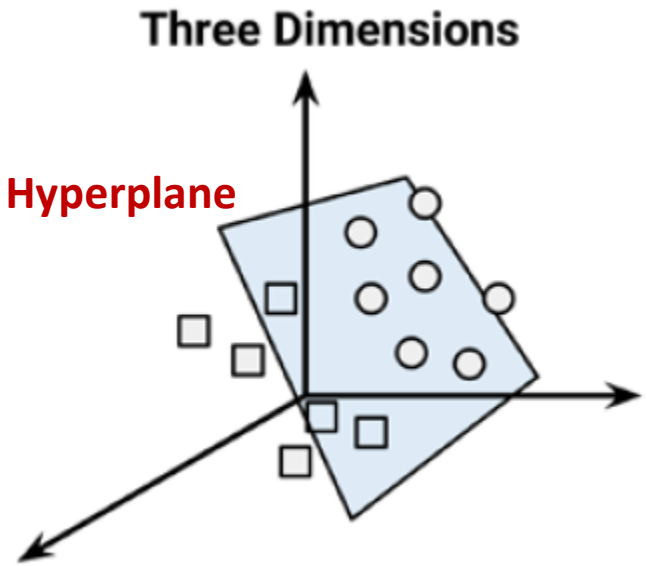
Support Vector Machines (SVM) 支持向量机

- A SVM can be imagined as a surface that creates a boundary between points of data plotted in multidimensional space
- The goal of a SVM is to create a flat boundary called a **hyperplane** (超平面), dividing the space to create fairly homogeneous partitions on either side
- Combine **Nearest Neighbors (k-NN)** and **linear regression** modeling
- Extremely powerful and allowing SVMs to model highly complex relationships, including both **classification** and **numeric prediction**.

Classification with Hyperplanes

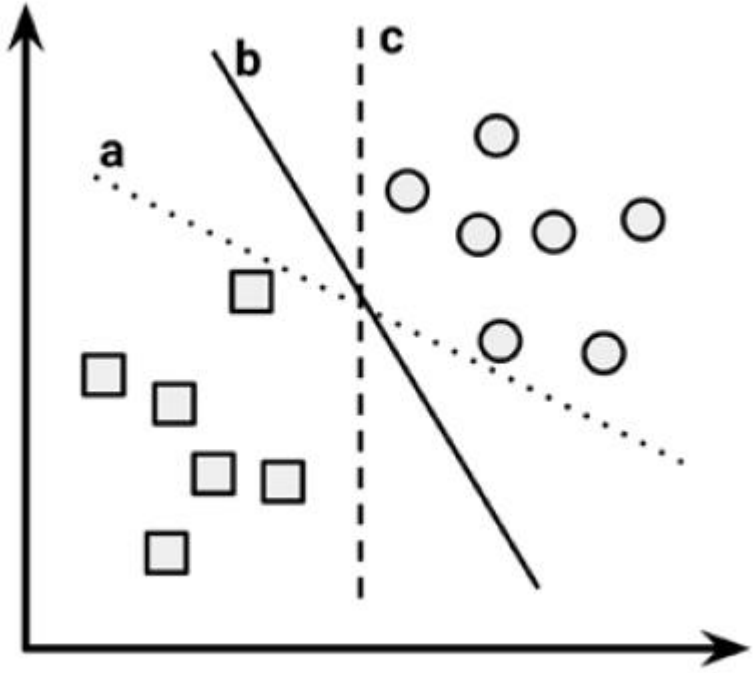
- Linearly Separable Data

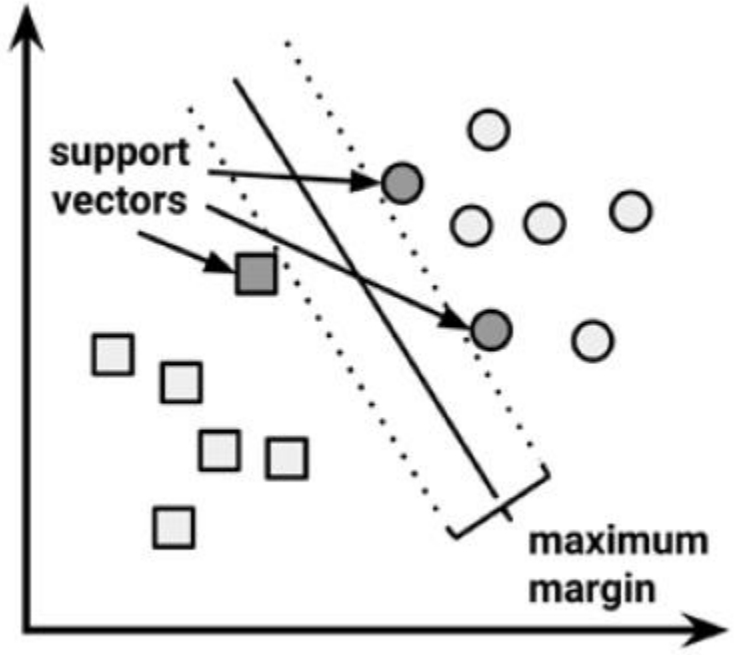




Maximum Margin Hyperplane (MMH)
最大间隔超平面

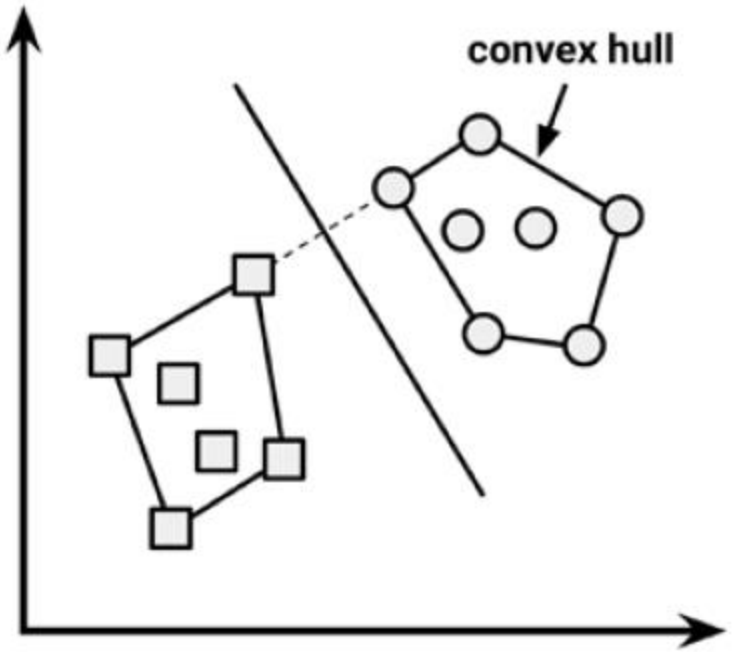
- Creates the greatest separation between two classes
- Support vectors are the points from each class that are the closest to the MMH





Maximum Margin Hyperplane (MMH)

- MMH is as far away as possible from the outer boundaries of the two groups of data points
- Outer boundaries are known as the **convex hull** (凸包).
- The MMH is then the perpendicular bisector of the shortest line between the two convex hulls



Linear SVM

Given a training dataset of n points

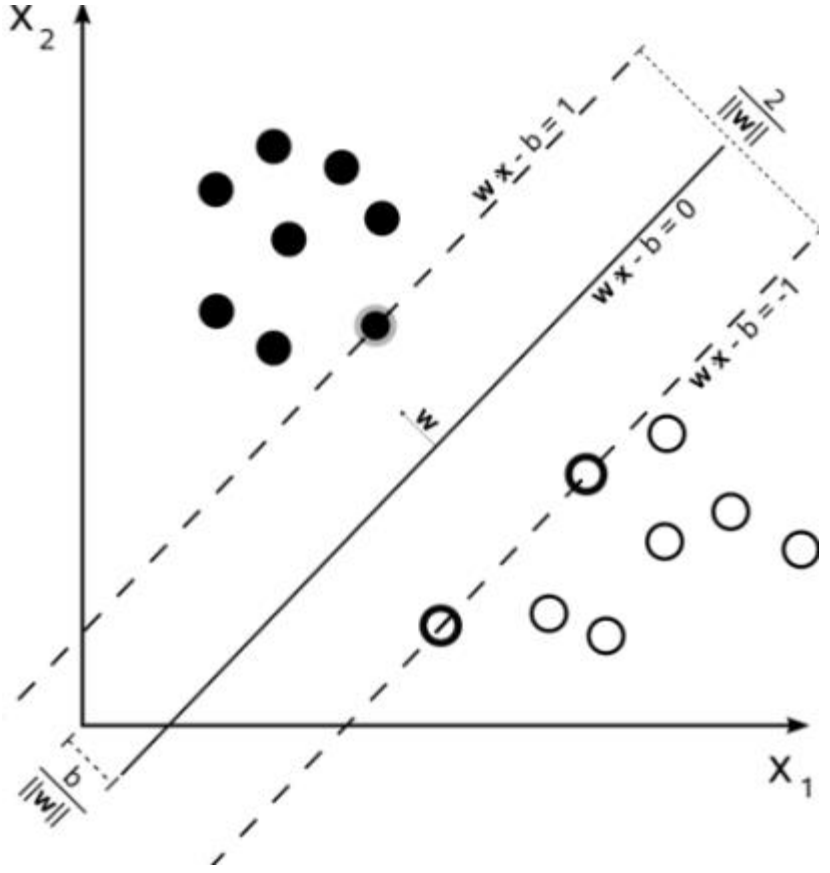
$$(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)$$

where the y_i are either 1 or −1, each indicating the class to which point $\vec{x}_i$ belongs.

$$\vec{w} \cdot \vec{x} - b = 0$$

$\vec{w}$: a vector of n weights $\{w_1, w_2, \ldots, w_n\}$

b: bias, conceptually equivalent to the intercept term in the slope-intercept



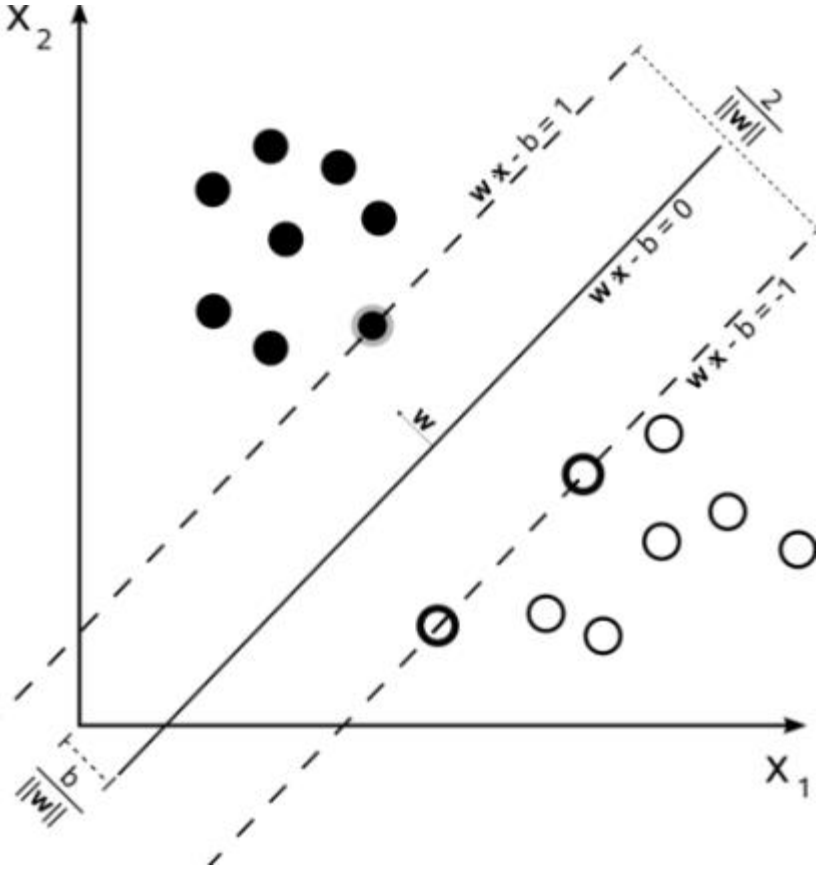
$$\vec{w} \cdot \vec{x} - b = 1$$

$$\vec{w} \cdot \vec{x} - b = -1$$

Linear SVM

- Vector geometry defines the distance between these two planes as:

$$\frac{2}{||\vec{w}||}$$ $||w||$ indicates the **Euclidean norm**



$$\vec{w} \cdot \vec{x}_i - b \geq 1, \text{ if } y_i = 1$$

$$\vec{w} \cdot \vec{x}_i - b \leq -1, \text{ if } y_i = -1$$

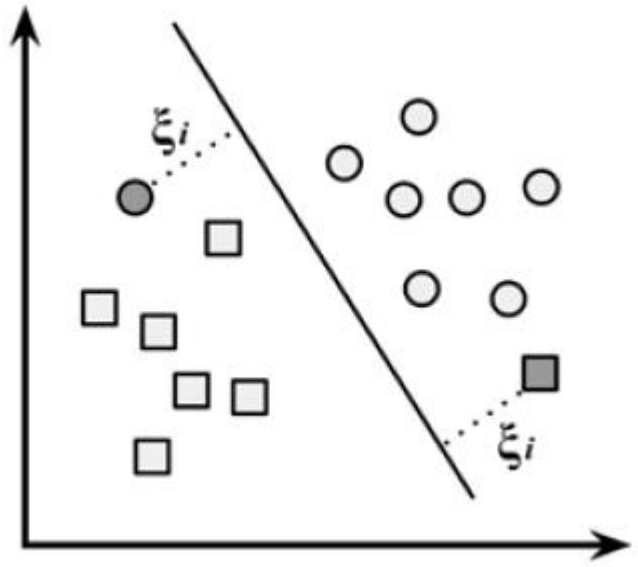
$$\min \frac{1}{2} \|\vec{w}\|^2$$ $||w||$ is in the denominator, to maximize distance, we need to minimize $||w||$

$$s.t. \;\; y_i(\vec{w} \cdot \vec{x}_i - b) \geq 1, \forall \vec{x}_i$$

Nonlinear SVM

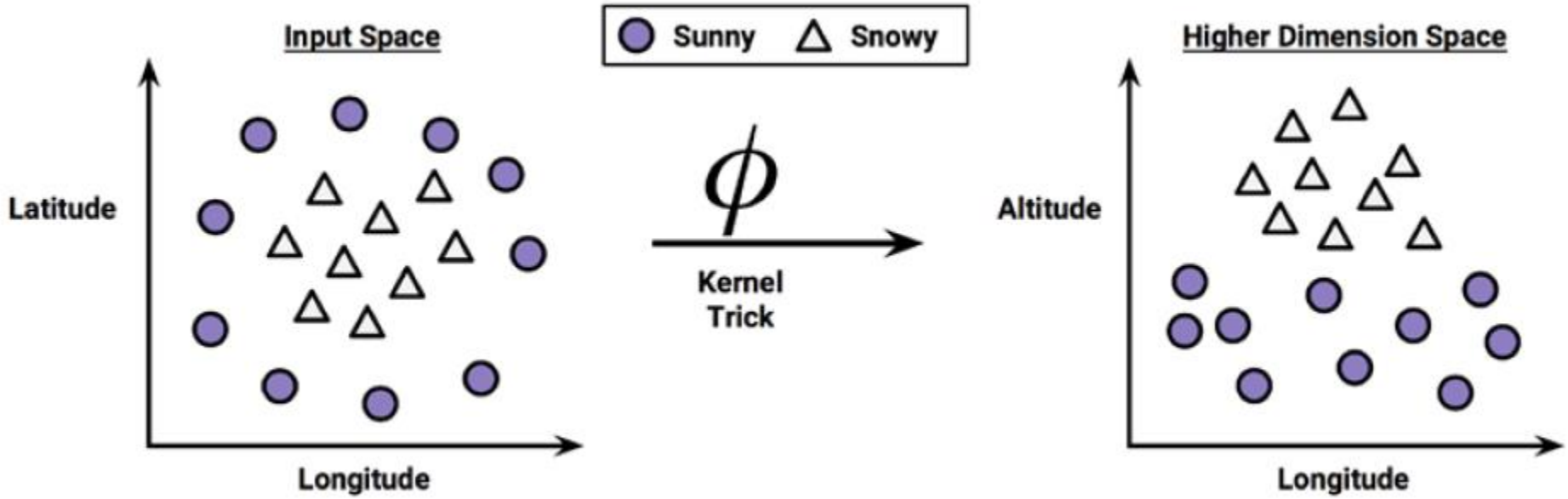
- nonlinearly separable
- Solution: use **slack variable (松弛变量)** ξ_i

$$\min \frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$
$$s.t.\ \ y_i(\vec{w}\cdot\vec{x}_i - b) \geq 1 - \xi_i, \forall \vec{x}_i, \xi_i \geq 0$$



Using Kernels for Nonlinear Spaces

- Slack variable may be used for nonlinear SVM
- A key feature of SVMs: their ability to map the problem into a higher dimension space using a process known as the **kernel trick (核技巧)**.
- Nonlinear relationship may suddenly appear to be quite linear



SVM with Nonlinear Kernels

Strengths	Weaknesses
• Can be used for classification or numeric prediction problems • Not overly influenced by noisy data and not very prone to overfitting • May be easier to use than neural networks, particularly due to the existence of several well-supported SVM algorithms • Gaining popularity due to its high accuracy and high-profile wins in data mining competitions	• Finding the best model requires testing of various combinations of kernels and model parameters • Can be slow to train, particularly if the input dataset has a large number of features or examples • Results in a complex black box model that is difficult, if not impossible, to interpret

Various Kernel Functions

Formula: Dot Product $\mathrm{K}(\vec{x_i}, \vec{x_j}) = \phi(\vec{x_i}) \cdot \phi(\vec{x_j})$

- Linear Kernel 线性核函数

$$\mathrm{K}(\vec{x_i}, \vec{x_j}) = \vec{x_i} \cdot \vec{x_j}$$

- Polynomial Kernel 多项式核函数

$$\mathrm{K}(\vec{x_i}, \vec{x_j}) = (\vec{x_i} \cdot \vec{x_j} + 1)^d$$

- Sigmoid Kernel S型核函数

$$\mathrm{K}(\vec{x_i}, \vec{x_j}) = \tanh(\kappa\, \vec{x_i} \cdot \vec{x_j} - \delta)$$

- Gaussian RBF Kernel 高斯RBF核函数

$$\mathrm{K}(\vec{x_i}, \vec{x_j}) = e^{\frac{-||\vec{x_i} - \vec{x_j}||^2}{2\sigma^2}}$$

Application of SVM in Transportation

- A Comparison of The Performance of ANN And SVM for the Prediction of Traffic Accident Duration (Yu, et al., 2016)
- Inferring Hybrid Transportation Modes from Sparse GPS Data Using a Moving Window SVM Classification (Bolbol, et al., 2012)
- Pedestrian Recognition for Intelligent Transportation Systems Using SVM-Based Classifier (Parra, et al., 2012)
- Short-Term Prediction of Freeway Exiting Volume Based on SVM And KNN (Wang, et al., 2015)

	Supervised Learning	Unsupervised Learning
Discrete	**Classification** Nearest Neighbor, Naive Bayes, Decision Trees, Classification Rule Learners Artificial Neural Network, Support Vector Machine	**Clustering** **k-means clustering**
Continuous	**Numeric Prediction** Linear Regression, Regression Trees, Model Trees	**Dimensionality Reduction**

Clustering with k-means k均值聚类

- Clustering is an **unsupervised machine learning** task that automatically divides the data into clusters, or groups of similar items
- k-means clustering aims to **partition n observations into k clusters** in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster.
- Clustering is used for **knowledge discovery** rather than **prediction**
- The goal of clustering is to **minimize the differences within each cluster** and **maximize the differences between the clusters**

Clustering with 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* (≤ *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)

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 = \arg\min_{\mathbf{S}} \sum_{i=1}^{k} |S_i| \operatorname{Var} S_i$$

where $\boldsymbol{\mu}_i$ is the mean of points in S_i

- The problem is computationally difficult (**NP-hard**); **heuristic algorithms** are commonly employed and converge quickly to a **local optimum**

k-means Algorithm

- Also referred to as **Lloyd's algorithm**
- Given an initial set of k means $m_1^{(1)}$,…, $m_k^{(1)}$, the algorithm proceeds by alternating between two steps:
- **Assignment step**: Assign each observation to the cluster whose mean has the least squared **Euclidean distance**, this is intuitively the "nearest" mean. (Mathematically, this means partitioning the observations according to the **Voronoi diagram** generated by the means

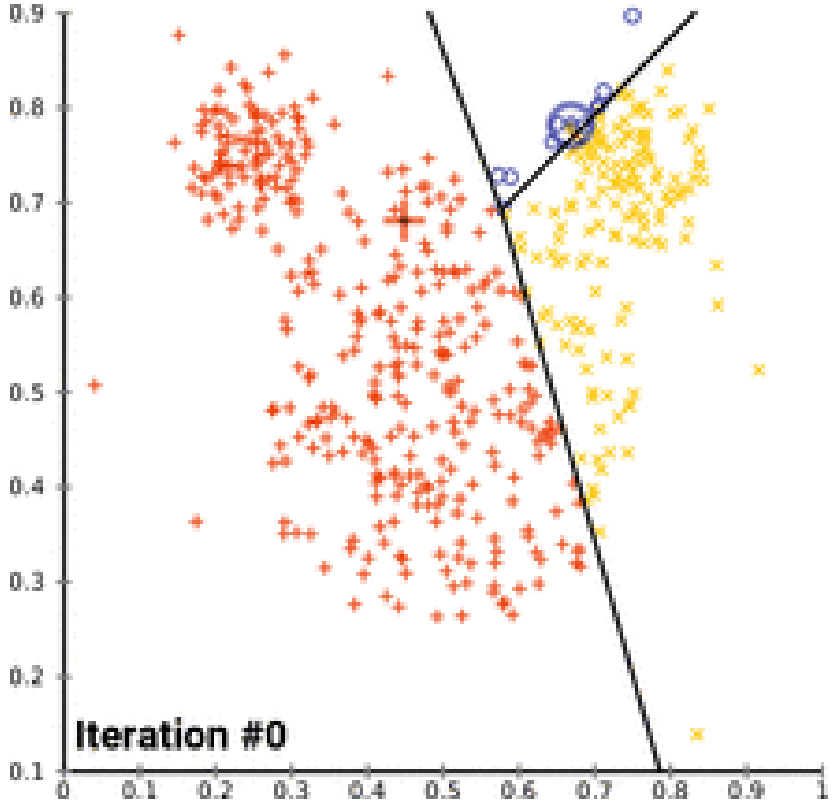
$$S_i^{(t)} = \left\{ x_p : \left\| x_p - m_i^{(t)} \right\|^2 \leq \left\| x_p - m_j^{(t)} \right\|^2 \ \forall j, 1 \leq j \leq k \right\}$$

- **Update step**: Calculate the new means to be the centroids of the observations in the new clusters

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

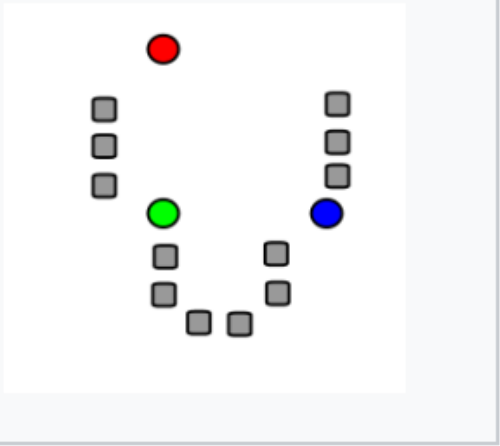
k-means Algorithm

- Convergence of k-means

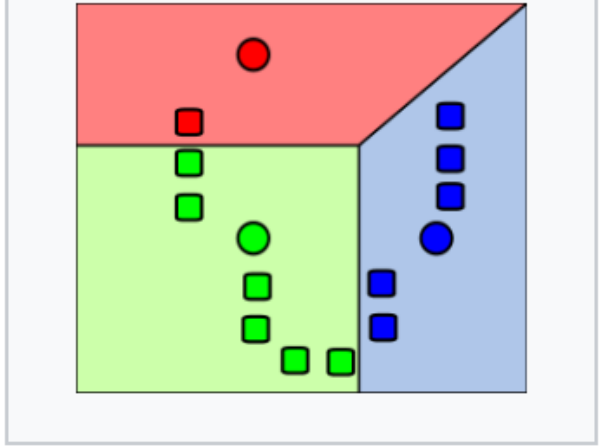


K-means Standard Algorithm

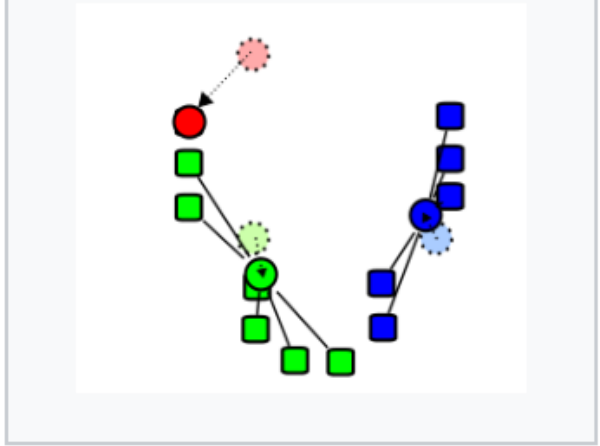
Demonstration of the standard algorithm



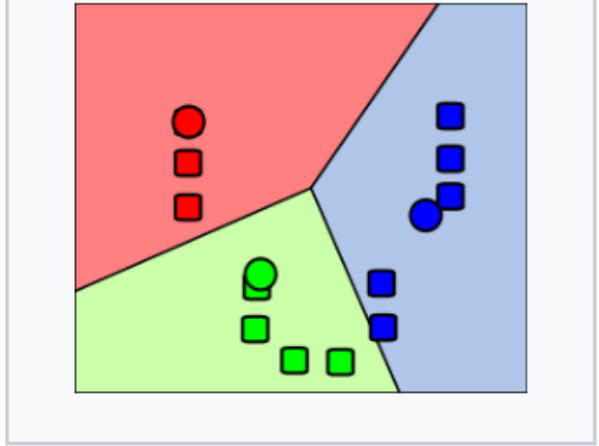
1. *k* initial "means" (in this case *k*=3) are randomly generated within the data domain (shown in color).



2. *k* clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.



3. The centroid of each of the *k* clusters becomes the new mean.



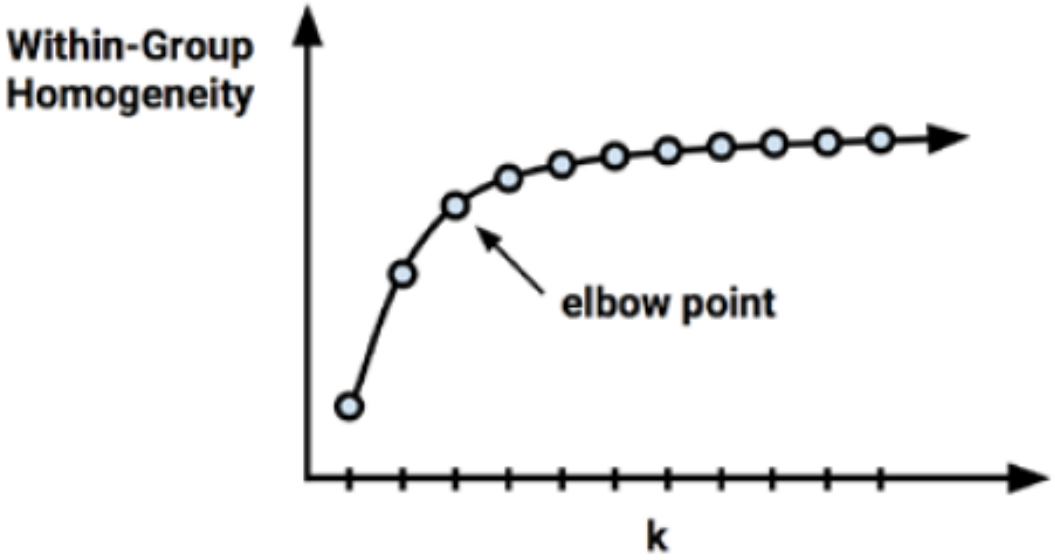
4. Steps 2 and 3 are repeated until convergence has been reached.

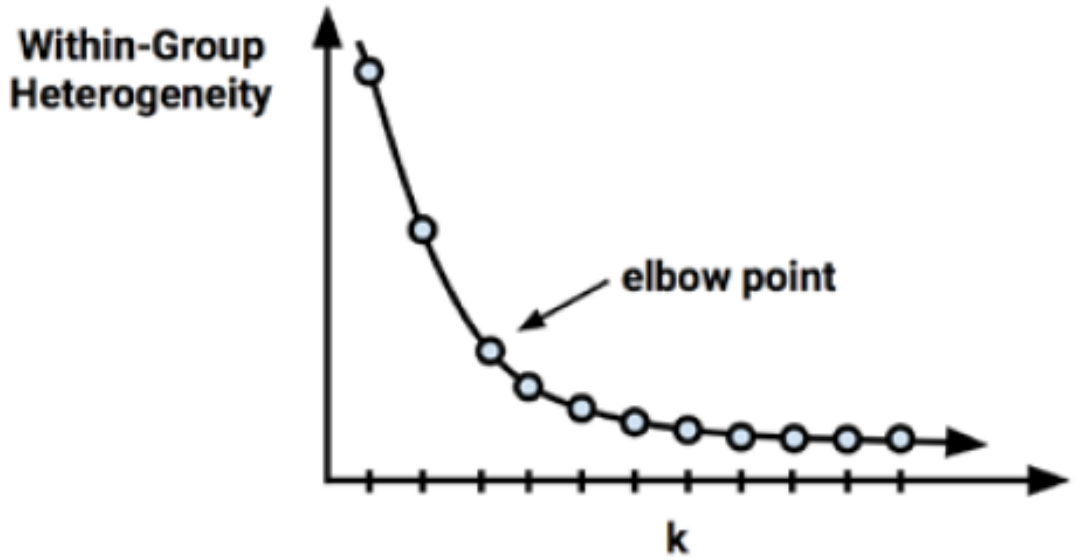
k-means Algorithm

Strengths	Weaknesses
• Uses simple principles that can be explained in non-statistical terms • Highly flexible, and can be adapted with simple adjustments to address nearly all of its shortcomings • Performs well enough under many real-world use cases	• Not as sophisticated as more modern clustering algorithms • Because it uses an element of random chance, it is not guaranteed to find the optimal set of clusters • Requires a reasonable guess as to how many clusters naturally exist in the data • Not ideal for non-spherical clusters or clusters of widely varying density

Choosing the Appropriate Number of Clusters

- **Elbow Method (肘部法)**





Application of k-means in Transportation

- Bus Stop Selection for Employees with Bi-Objective Particle Swarm Optimization Approach Using k-means: Case Study (Deliktas, et al., 2017)
- Kernel Density Estimation and k-means Clustering to Profile Road Accident Hotspots (Anderson, 2008)
- Modified k-means Clustering for Travel Time Prediction Based on Historical Traffic Data (Nath, et al., 2010)
- Using k-means Clustering to Identify Time-of-Day Break Points for Traffic Signal Timing Plans (Wang, et al., 2005)
- Tehran Driving Cycle Development Using the k-means Clustering Method (Fotouhi, et al., 2013)

Improving Model Performance

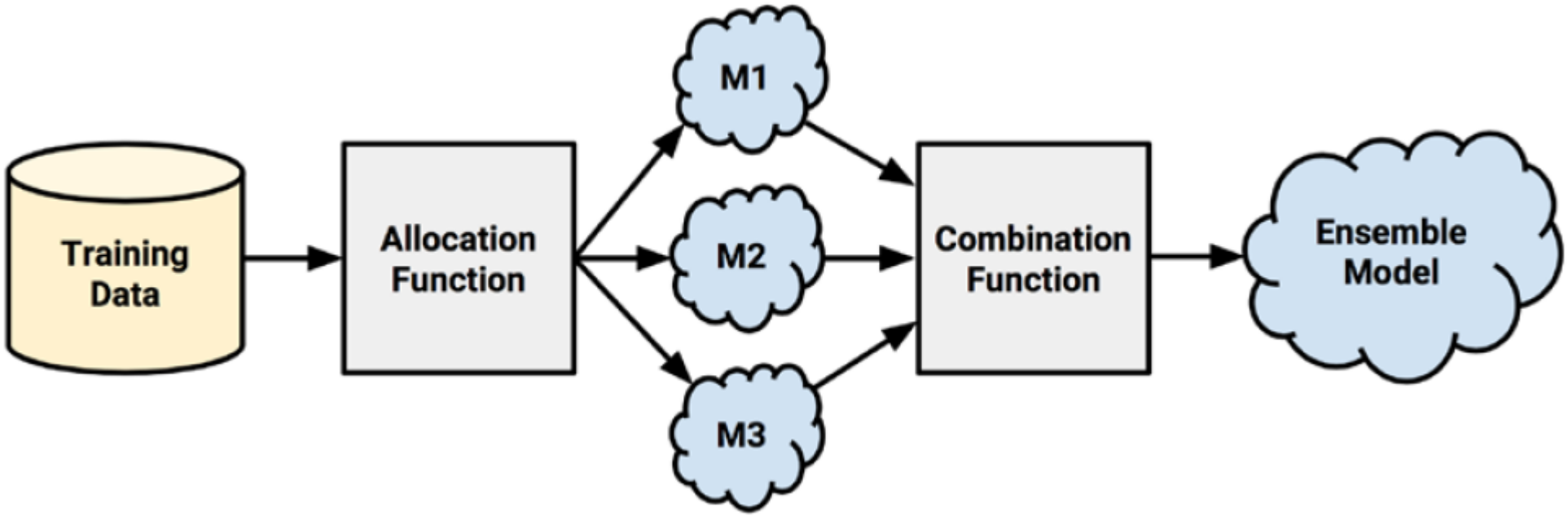
Parameter Tuning 参数调整

- For example, in C5.0 decision tree, attempt to improve its performance by adjusting the trials parameter to increase the number of **boosting (自助抽样)** iterations to increase the model's accuracy

Meta-Learning 元学习

- Combine several models to form a powerful team
- **Ensembles (集成学习)**

Ensembles Approach



Meta-Learning

Bagging 自助汇聚法

- Generates a number of training datasets by bootstrap sampling the original training data. These datasets are then used to generate a set of models using a single learning algorithm. The models' predictions are combined using voting (for classification) or averaging (for numeric prediction)
- For additional information on bagging, refer to Breiman L. Bagging predictors. Machine Learning. 1996; 24:123-140.

Meta-Learning

Boosting 自助抽样法

- **Similar to bagging**, boosting uses ensembles of models trained on resampled data and a vote to determine the final prediction.

There are two key distinctions：

- First, the resampled datasets in boosting are constructed specifically to generate **complementary learners** 产生互补模型
- Second, rather than giving each learner an equal vote, boosting gives each learner's vote **a weight** based on its past performance. Models that perform better have greater influence over the ensemble's final prediction.

Meta-Learning

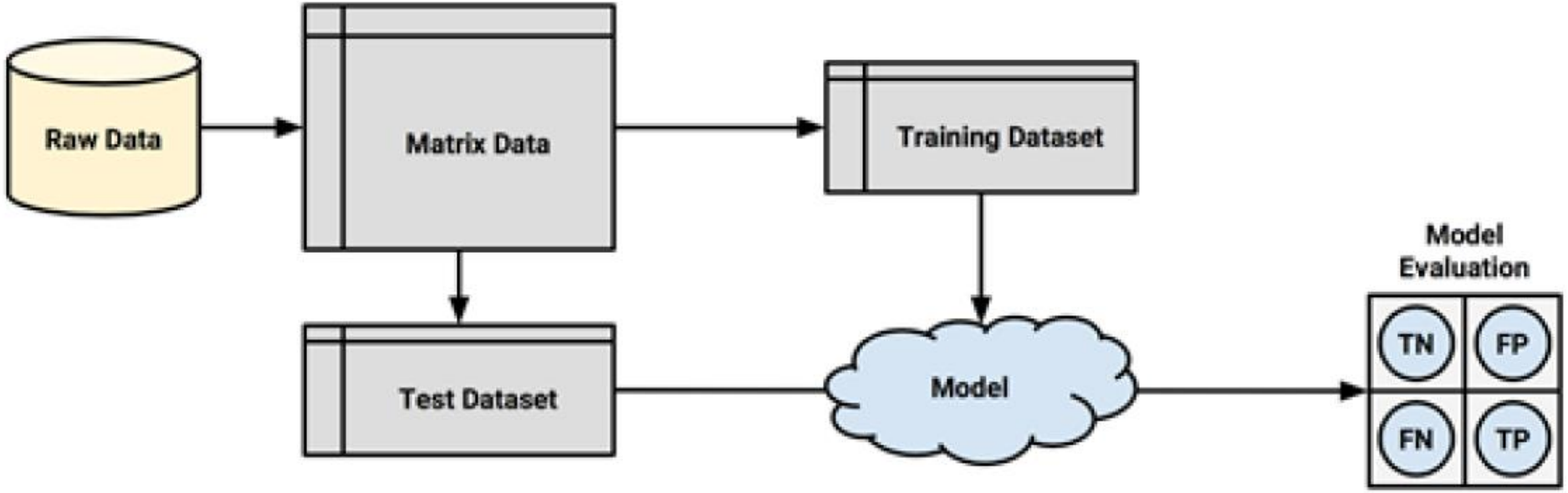
- **Random Forests / Decision Tree Forests 随机森林 / 决策树森林**
- Focuses only on ensembles of decision trees
- Combines the **base principles of bagging** with **random feature selection** to add additional diversity to the decision tree models.
- After the ensemble of trees (the forest) is generated, the model uses a **vote** to combine the trees' predictions

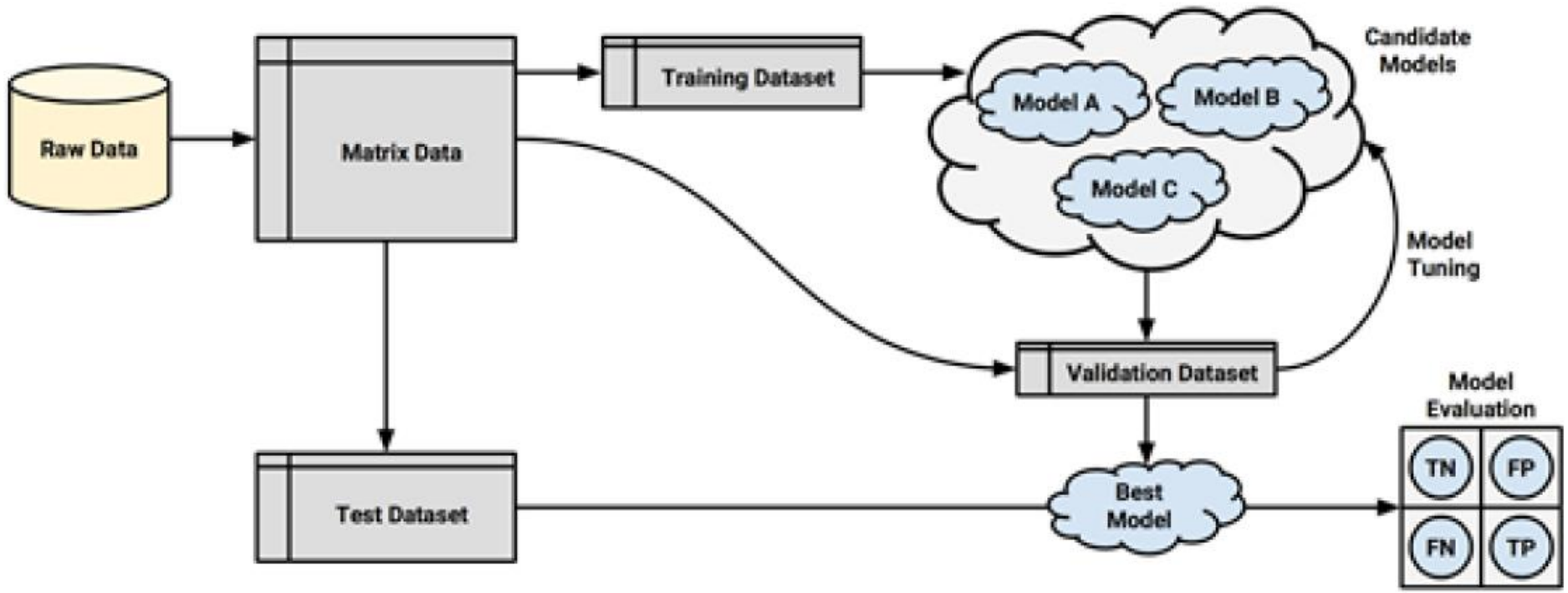
Estimating Future Performance

- **Resubstitution Error (再带入误差)**, such as confusion matrices and performance measures, occurs when the training data is incorrectly predicted in spite of the model being built directly from this data
- The resubstitution error is not a very useful marker of future performance
- **Holdout Method (保持法)**

  Usually: **75%** for training set, **25%** for test set

  Modified Method: **50%** for training, **25%** for validation, **25%** for test


Raw Data
Matrix Data
Training Dataset
Test Dataset
Model
Model Evaluation
TN
FP
FN
TP


Raw Data
Matrix Data
Training Dataset
Model A
Model B
Model C
Candidate Models
Model Tuning
Validation Dataset
Test Dataset
Best Model
Model Evaluation
TN
FP
FN
TP

Estimating Future Performance

- **k-fold Cross-Validation:** mainly **10-folder CV**
- For each of the 10 folds (each comprising 10 percent of the total data), a machine learning model is built on the remaining 90 percent of data

Estimating Future Performance

- **Kappa statistic**: adjusts accuracy by accounting for the possibility of a correct prediction by chance alone
- Kappa values range from 0 to a maximum of 1, which indicates perfect agreement between the model's predictions and the true values
    - Poor agreement = less than 0.20
    - Fair agreement = 0.20 to 0.40
    - Moderate agreement = 0.40 to 0.60
    - Good agreement = 0.60 to 0.80
    - Very good agreement = 0.80 to 1.00