《机器学习基础》

无监督学习

内容

- 无监督学习
  - 主成分分析
  - 聚类分析

监督学习和无监督学习

- 在监督学习中，我们有一个包含输入特征（X）和对应的目标输出（y）的训练数据集。模型通过学习输入和输出之间的映射关系，以便在给定新的输入时能够预测输出。

- 无监督学习是在没有给定明确的目标输出的情况下，让模型自动从数据中发现模式和结构。也就是说，数据集中只有输入特征X，没有对应的y。

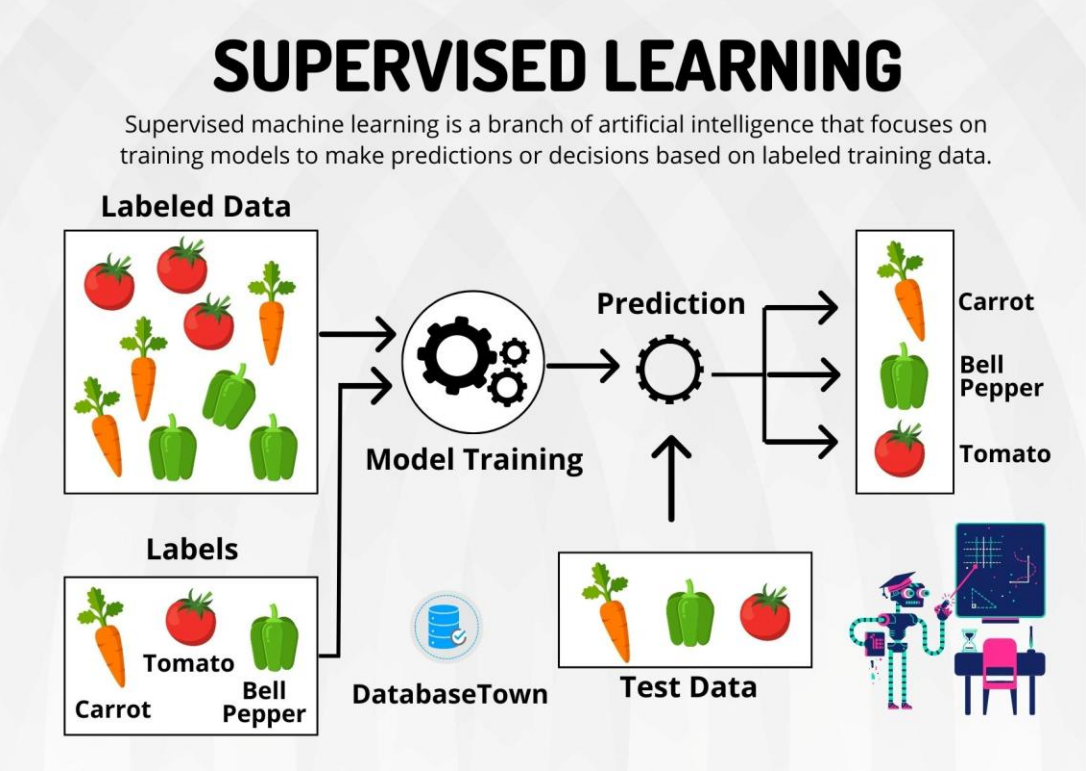
无监督学习（Unsupervised Learning）

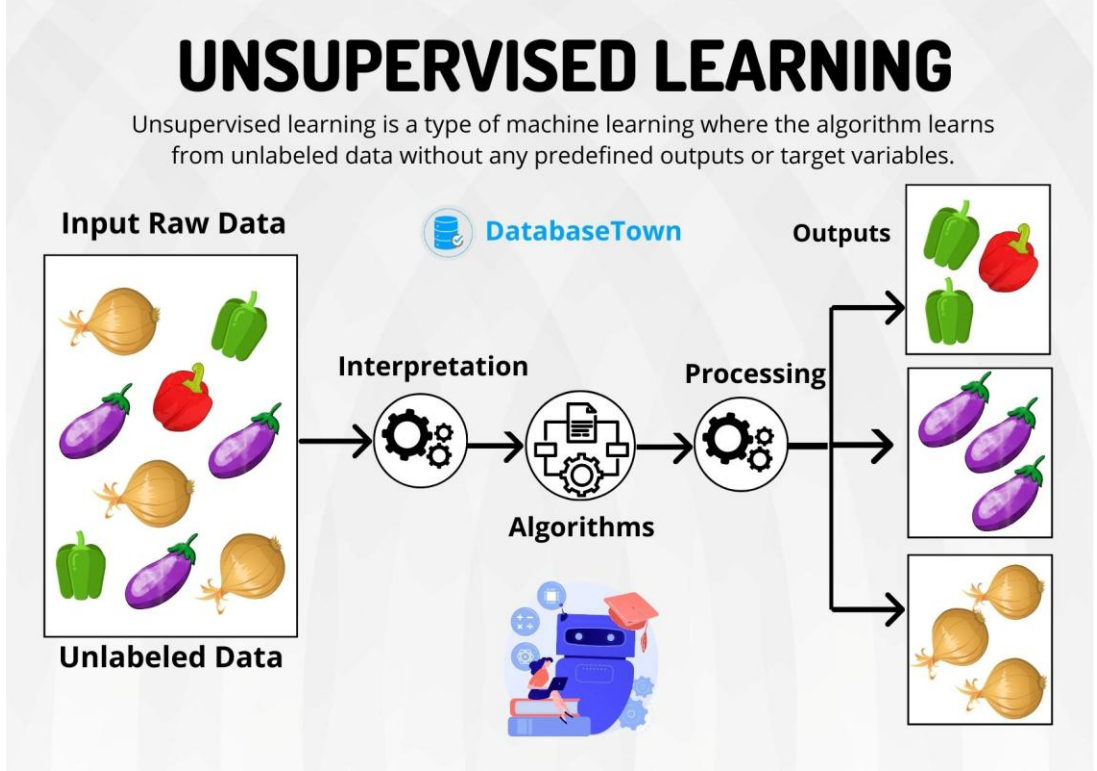
- 监督学习
  - 建立映射关系 $f: x \rightarrow y$

- 无监督学习
  - 指从无标签的数据中学习出一些有用的模式。

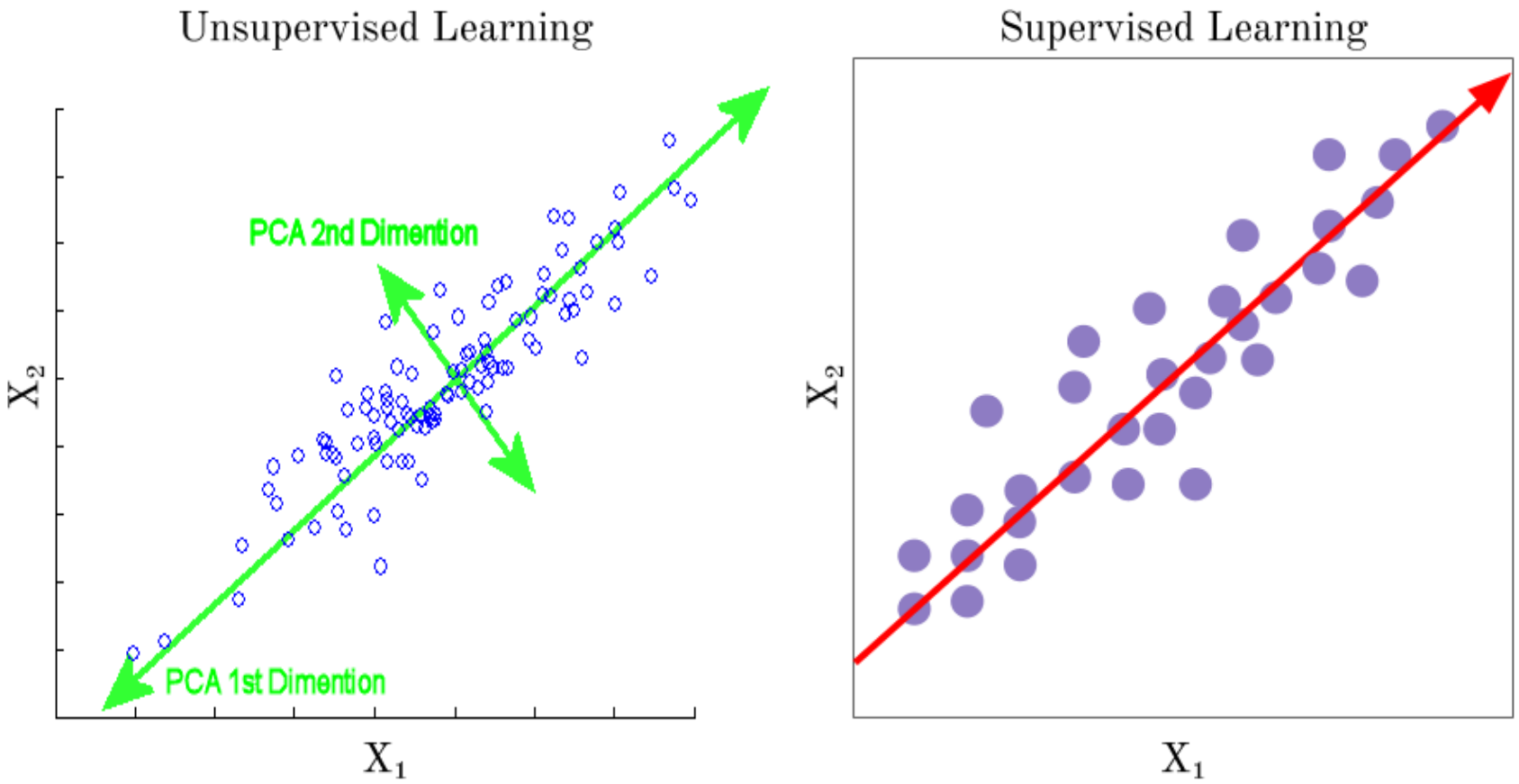
  - 聚类：建立映射关系 $f: x \rightarrow y$
    - 不借助于任何人工给出标签或者反馈等指导信息
  - 特征学习

监督学习和无监督学习


SUPERVISED LEARNING
Supervised machine learning is a branch of artificial intelligence that focuses on training models to make predictions or decisions based on labeled training data.
Labeled Data
Labels
Carrot
Tomato
Bell Pepper
Model Training
Prediction
Carrot
Bell Pepper
Tomato
DatabaseTown
Test Data


UNSUPERVISED LEARNING
Unsupervised learning is a type of machine learning where the algorithm learns from unlabeled data without any predefined outputs or target variables.
Input Raw Data
Unlabeled Data
DatabaseTown
Interpretation
Algorithms
Processing
Outputs

监督学习和无监督学习


Unsupervised Learning
PCA 2nd Dimention
PCA 1st Dimention
X2
X1
Supervised Learning
X2
X1

为什么要无监督学习?

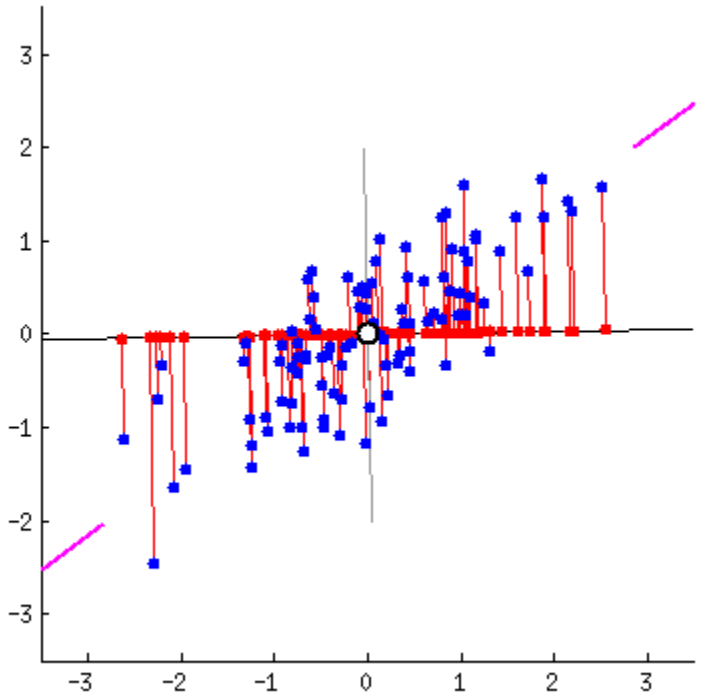
大脑有大约10^{14}个突触，我们只能活大约10^{9}秒。所以我们有比数据更多的参数。这启发了我们必须进行大量无监督学习的想法，因为感知输入（包括本体感受）是我们可以获得每秒10^{5}维约束的唯一途径。

-- Geoffrey Hinton, 2014 AMA on Reddit

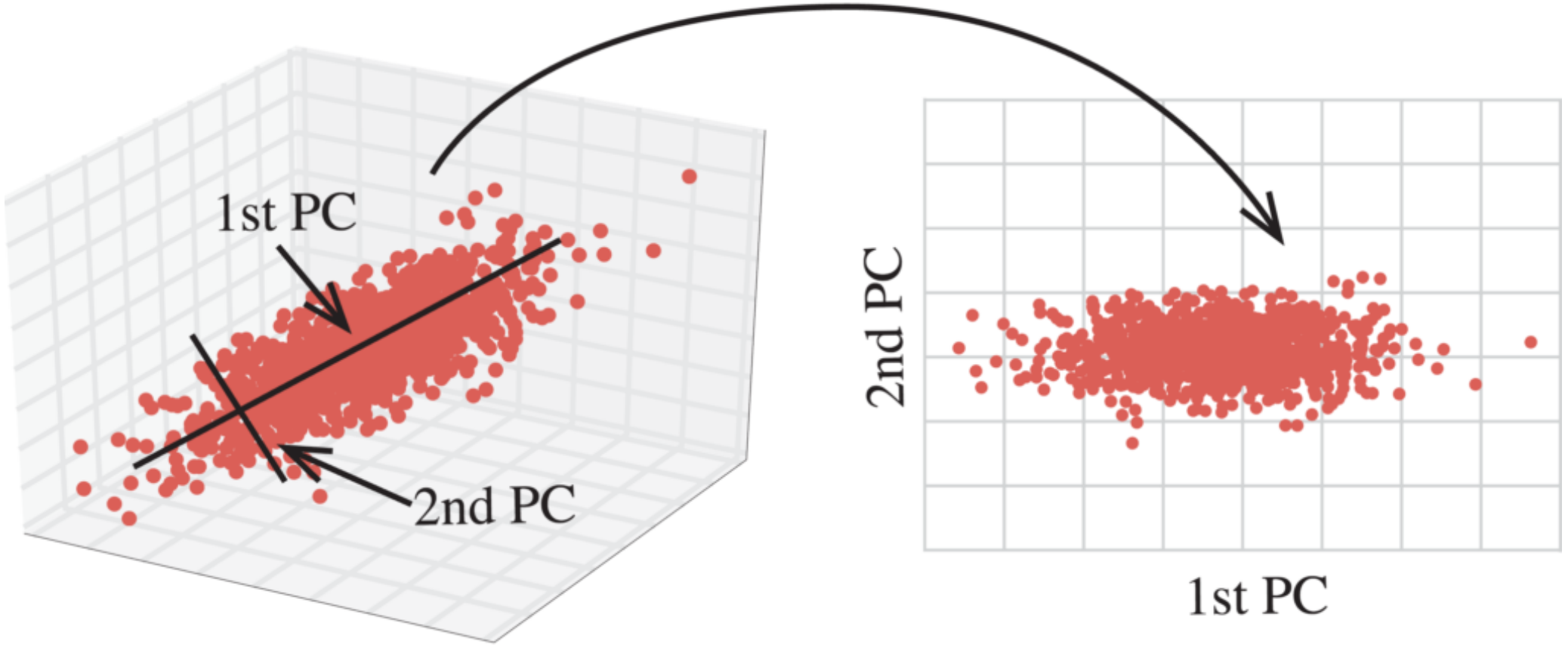
主成分分析

PCA优化目标

- 一种最常用的数据降维方法，使得在转换后的空间中数据的方差最大。



主成分分析的优化目标



PCA推导

- PCA的目标是找到一组新的正交基 $\{u_1, u_2, \cdots, u_k\}$（从$m$维下降到$k$维），使得数据点在该正交基构成的平面上投影后，数据间的距离最大，即数据间的方差最大。

- 如果数据在每个正交基上投影后的方差最大，那么同样满足在正交基所构成的平面上投影距离最大。

PCA推导

- 设正交基u_j，数据点x_i在该基底上的投影距离为$x_i u_j$，所以所有数据在该基底上投影的方差J_j为：

$$J_j = \frac{1}{n}\sum_{i=1}^{n}\left(x_i u_j - x_{center} u_j\right)^2$$

- 由于在数据运算之前对数据x进行0均值初始化，上式可写作

$$J_j = \frac{1}{n}\sum_{i=1}^{n}\left(x_i u_j\right)^2$$

$$= \frac{1}{n}\sum_{i=1}^{n}\left(u_j^T x_i^T \cdot x_i u_j\right) = u_j^T \cdot \frac{1}{n}\sum_{i=1}^{n}\left(x_i^T x_i\right) \cdot u_j$$

PCA推导

- 写成矩阵形式，有

$$J_j = \frac{1}{n} u_j^T X^T X u_j$$

- 则优化问题为

$$\max_{u_j} J_j = \frac{1}{n} u_j^T X^T X u_j$$
$$s.t。 \quad u_j^T u_j = 1$$

PCA推导

- 只提取一个主成分：

$$\max_{u_j} J_j = \frac{1}{n} u_j^T X^T X u_j$$
$$s.t。 \quad u_j^T u_j = 1$$

- 提取k个主成分：

$$\max_{U_k} J_k = \frac{1}{n} tr(U_k^T X^T X U_k)$$
$$s.t. \quad U_k^T U_k = I_k$$

主成分分析

- 如何得到包含最大差异性的主成分方向?

- 通过计算数据矩阵的协方差矩阵，然后得到协方差矩阵的特征值特征向量，**选择特征值最大(即方差最大)的k个特征所对应的特征向量组成的矩阵。**这样就可以将数据矩阵转换到新的空间当中，实现数据特征的降维。

- 两种实现方法：基于特征值分解协方差矩阵实现PCA算法、基于SVD分解协方差矩阵实现PCA算法。

特征值分解

- 如果一个向量v是矩阵A的特征向量，将一定可以表示成下面的形式：

$$Av = \lambda v$$

- 其中，λ是特征向量v对应的特征值，一个矩阵的一组特征向量是一组正交向量。
- 对于矩阵A，有一组特征向量v，将这组向量进行正交化单位化，就能得到一组正交单位向量。特征值分解就是将矩阵A分解为如下式：

$$A = V\Sigma V^{-1}$$

- 其中，V是矩阵A的特征向量组成的矩阵，Σ则是一个对角阵，对角线上的元素就是特征值。

SVD分解

- 奇异值分解是一个能适用于任意矩阵的一种分解的方法，对于任意矩阵X总是存在一个奇异值分解：
  - $X = W\Sigma U^T$
- 设X是一个$n \times m$的矩阵，那么得到的U是一个$m \times m$的方阵，U里面的正交向量被称为左奇异向量。
- Σ是一个$n \times m$的矩阵，Σ除了对角线其它元素都为0，对角线上的元素称为奇异值。
- 一般来讲，算法会将Σ上的值按从大到小的顺序排列。

主成分分析

- 数据集 $X = \{x_1, x_2, \ldots, x_n\} \in R^{n \times m}$,其行为数据样本，列为数据类别，并经过了去平均值处理。则X的奇异值分解为
  - $X = W\Sigma U^T$
- 据此，

$$\begin{aligned} Y &= XU \\ &= W\Sigma U^{\mathrm{T}} U \\ &= W\Sigma \end{aligned}$$

- 我们可以利用U_k把 X 映射到一个只应用前面k个向量的低维空间中去：
  - $Y = XU_k = W\Sigma U^{\mathrm{T}} U_{\mathrm{k}} = \mathrm{W}\Sigma_k$

选择降维后的维度K(主成分的个数)

- x_i:原始数据，y_i:PCA降维后的数据
  - average squared projection error:

$$\frac{1}{n}\sum_{i=1}^{n}||x_i - y_i||^2$$

- total variation in the data:

$$\frac{1}{n}\sum_{i=1}^{n}||x_i||^2$$

选择降维后的维度K(主成分的个数)

- 选择不同的K值，然后用下面的式子不断计算，选取能够满足下列式子条件的最小K值即可。

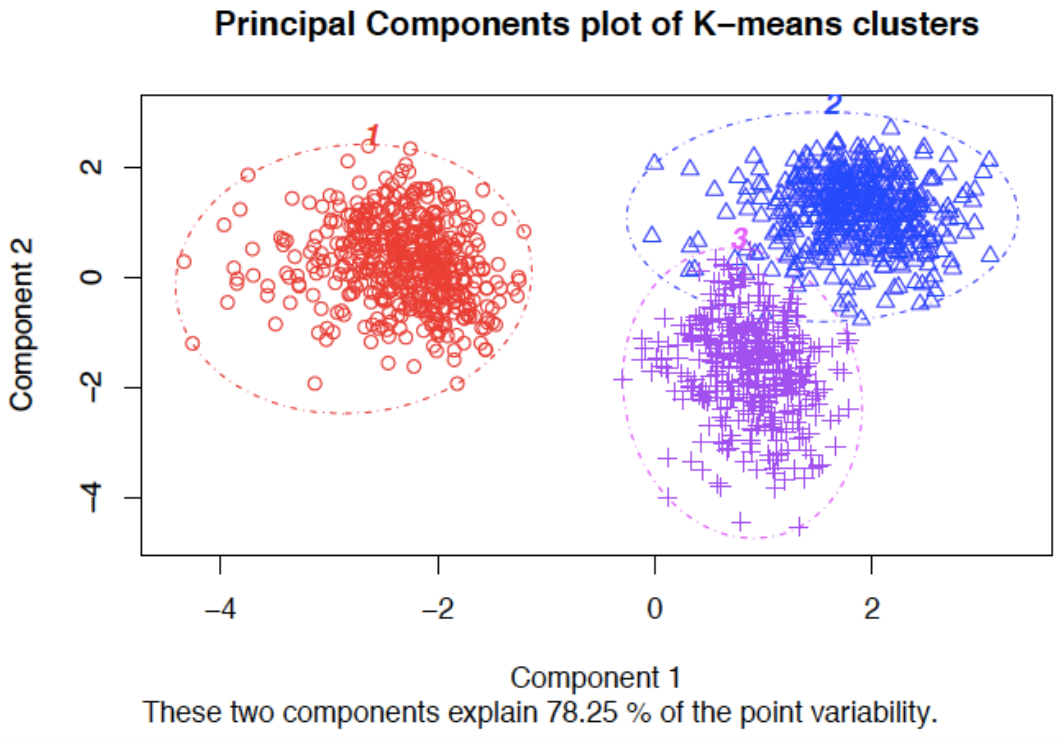
$$\frac{\frac{1}{n}\sum_{i=1}^{n}||x_i - y_i||^2}{\frac{1}{n}\sum_{i=1}^{n}||x_i||^2} \leq t$$

- 其中t值可以由自己定，比如t值取0.01，则代表了该PCA算法保留了99%的主要信息。当你觉得误差需要更小，你可以把t值设置的更小。

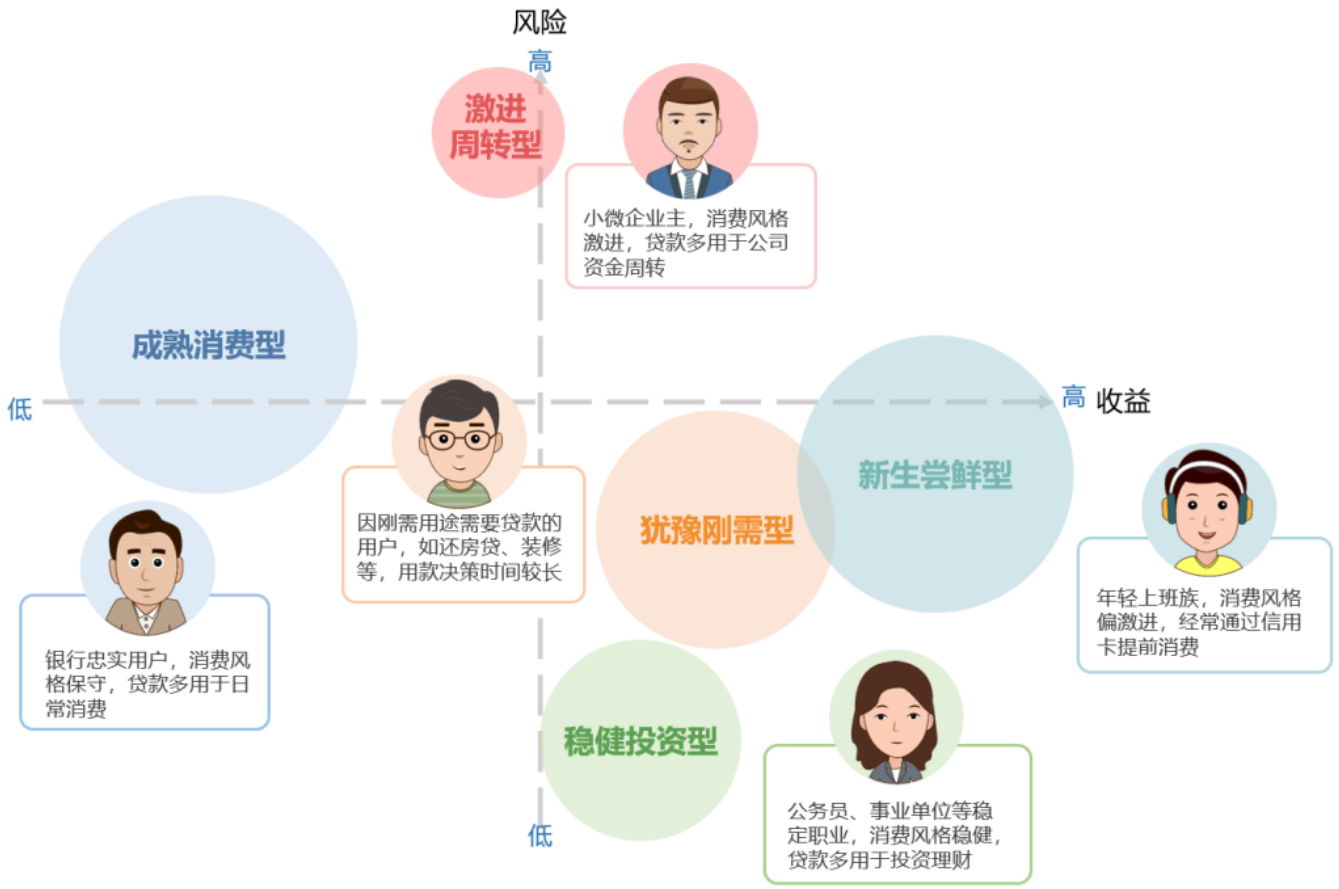
聚类分析

聚类

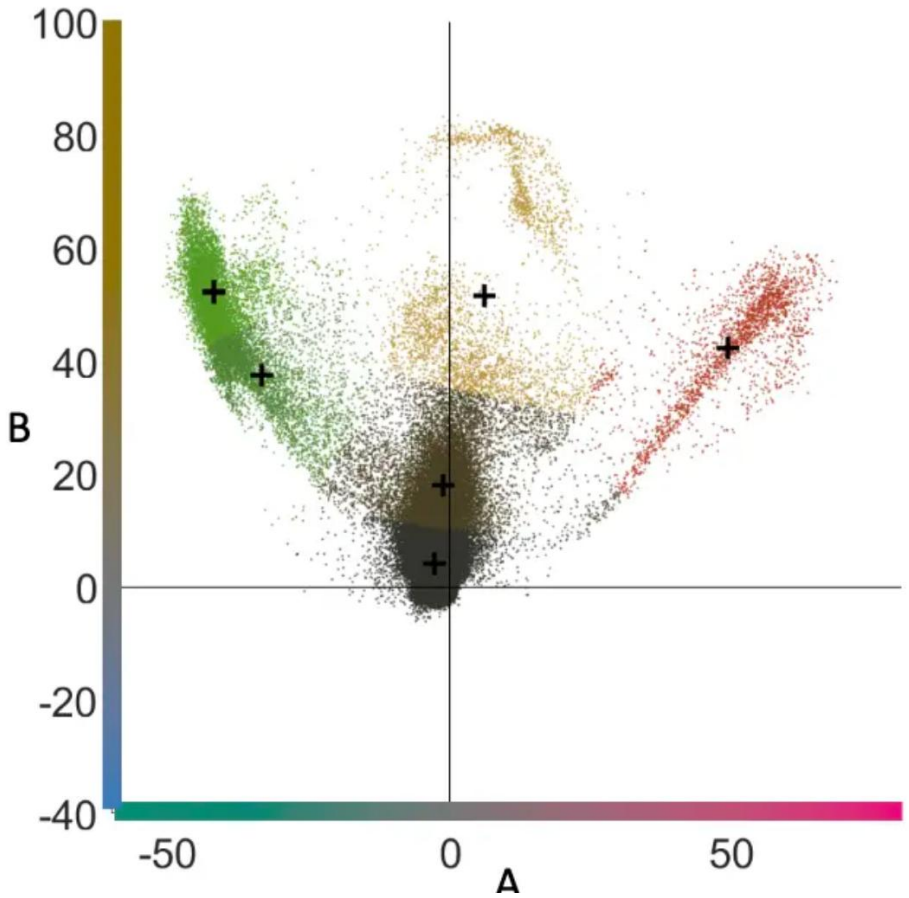
- 聚类分析（Cluster Analysis）：将对象的集合分组为由类似对象组成的多个类的分析过程。
- 发现数据中的自然结构，使得同一类中的对象彼此相似，不同类中的对象相互差异较大。
- 聚类分析不需要对分类的数目和结构作出预先假定，属于无监督学习（Unsupervised Learning）。



聚类的应用

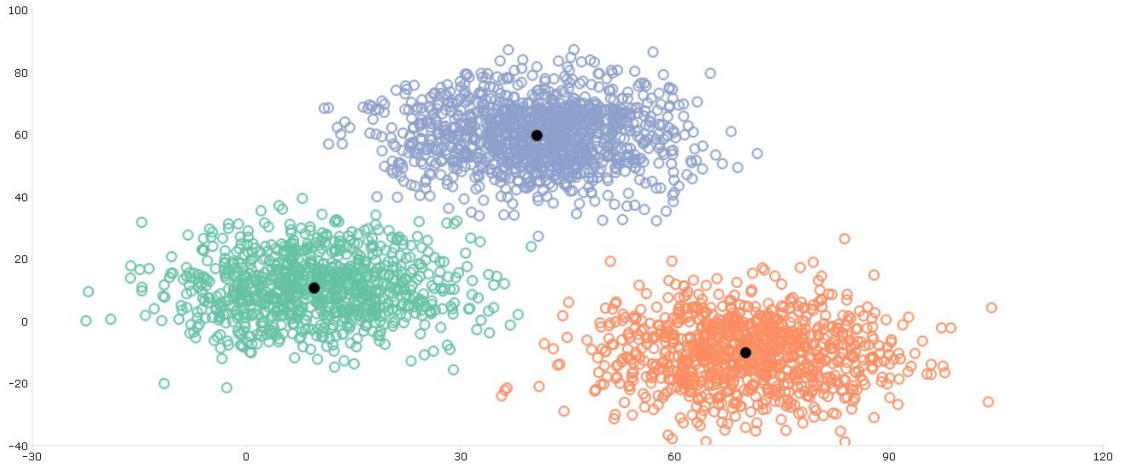

风险
高
激进
周转型
小微企业主，消费风格
激进，贷款多用于公司
资金周转
成熟消费型
低
高 收益
因刚需用途需要贷款的
用户，如还房贷、装修
等，用款决策时间较长
犹豫刚需型
新生尝鲜型
年轻上班族，消费风格
偏激进，经常通过信用
卡提前消费
银行忠实用户，消费风
格保守，贷款多用于日
常消费
稳健投资型
公务员、事业单位等稳
定职业，消费风格稳健，
贷款多用于投资理财
低

聚类的应用



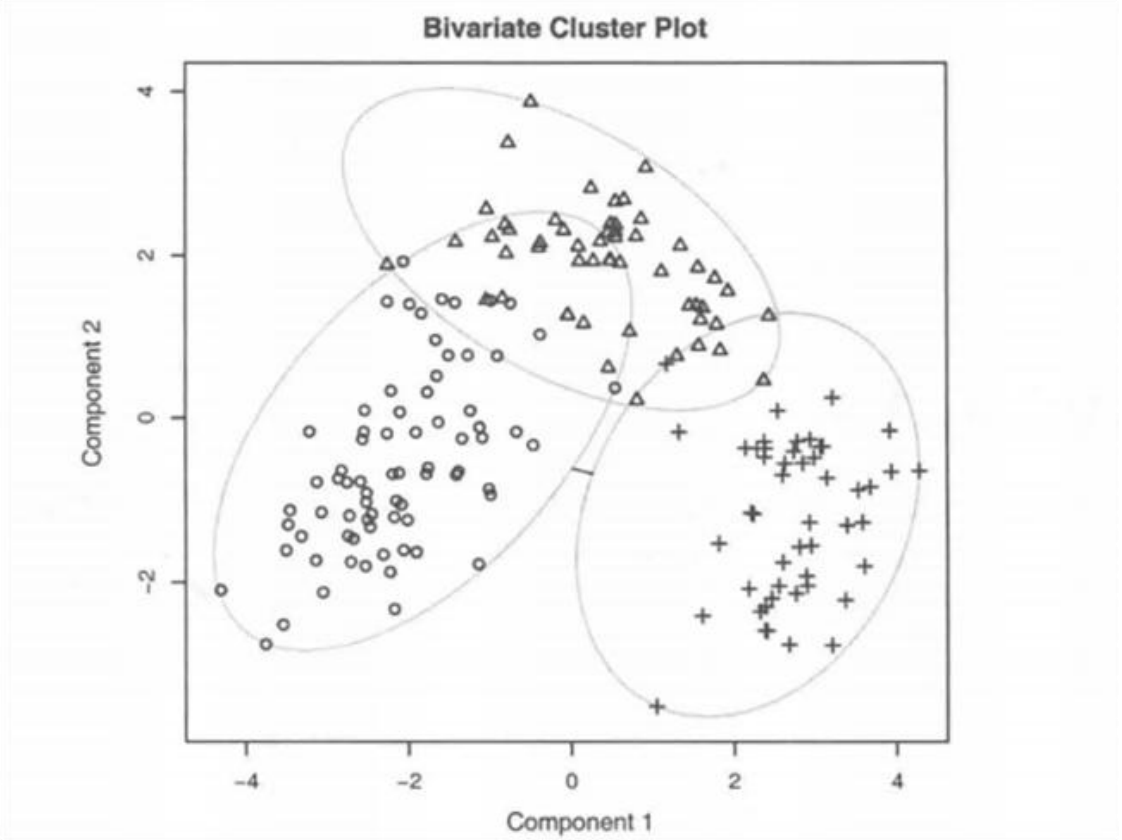
Q型聚类：

- 也称为样本聚类（Sample Clustering）或硬聚类（Hard Clustering）。
  - Q 型聚类基于样本之间的相似性度量，将性质相似的样本归为一类。
  - 每个样本只属于一个类，类与类之间没有交集。



R型聚类

- 也称为变量聚类（Variable Clustering）或属性聚类。
  - R 型聚类基于变量之间的相关性度量，将相关性高的变量归为一类。
  - 与Q型聚类不同，R型聚类的每个样本可以属于多个分类，类与类之间可能存在交集。



距离

- 距离度量是用来评估数据点之间相似性或差异性的一种方法。
- 选择合适的距离度量对于聚类结果的质量和解释性至关重要。
- 常用的距离有：
  - 欧式距离（Euclidean Distance）
  - 曼哈顿距离（Manhattan Distance）
  - 闵可夫斯基距离（Minkowski Distance）
  - 汉明距离（Hamming Distance）
  - 兰氏距离（Lance-Williams Distance）

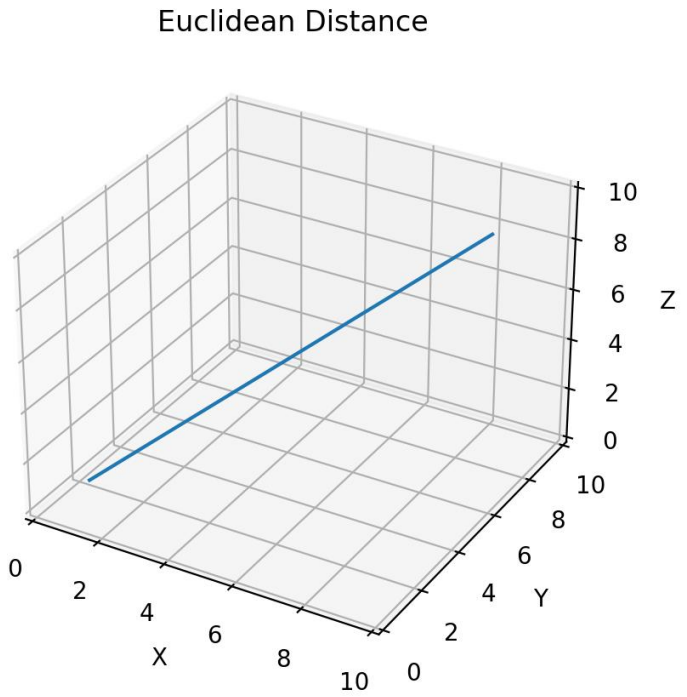
欧氏距离（Euclidean Distance）

- 定义为两点之间的直线距离
- 在二维空间中，如果两点的坐标分别为 (x_1, y_1) 和 (x_2, y_2)，那么它们之间的欧氏距离 d 计算公式为：

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

- 推广到多维空间，两点 x 和 y 之间的欧氏距离为：

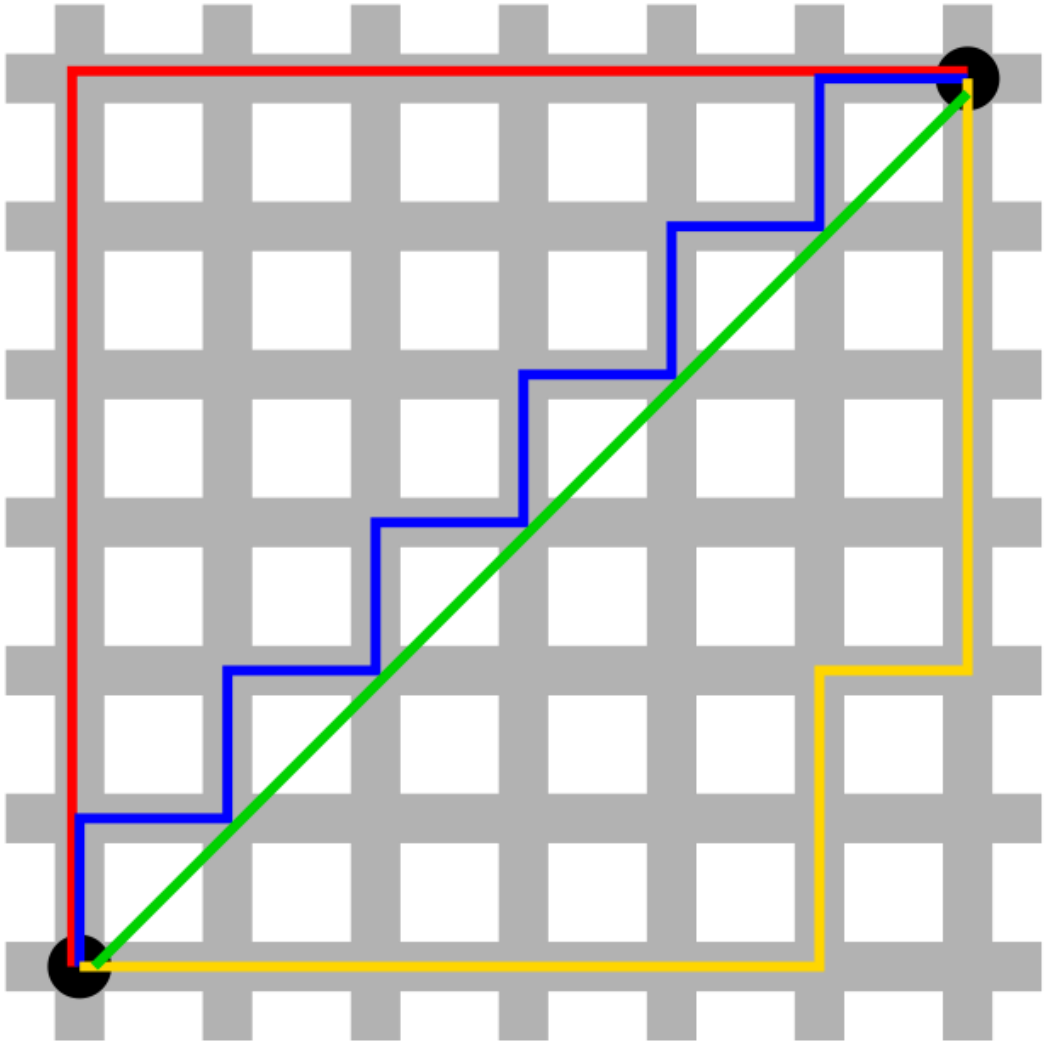
$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$



曼哈顿距离（Manhattan Distance）

- 也称为城市街区距离
- 在城市中，出租车通常只能沿着街道行驶，所以这个距离度量反映了出租车行驶的最短路径。
- 其公式为两点在坐标轴上的绝对轴距总和：

$$d(x,y)=\sum_{i=1}^{n}|x_i-y_i|$$

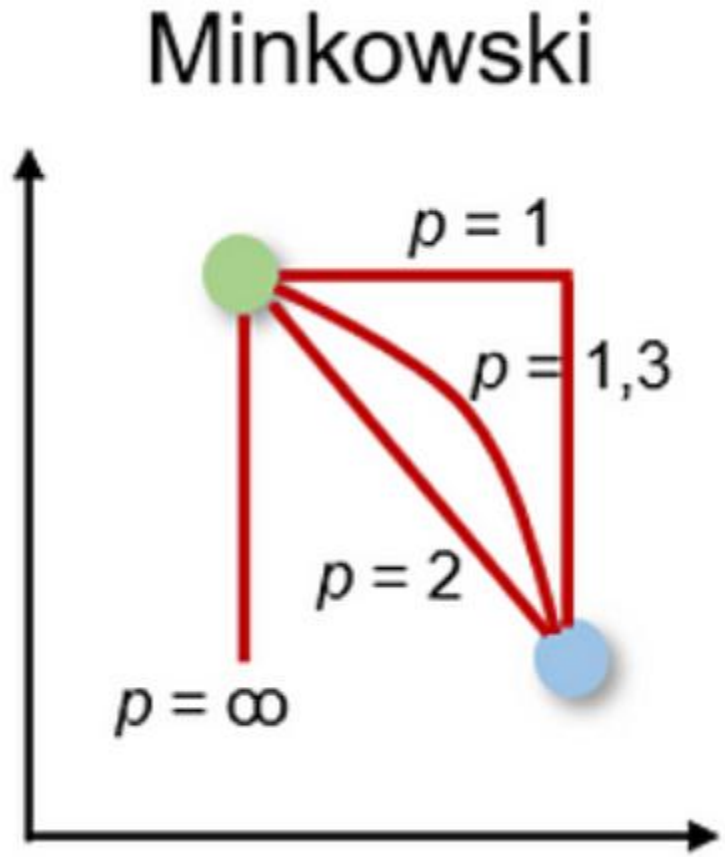


闵可夫斯基距离（Minkowski Distance）

- 闵可夫斯基距离可以表示为：

$$d(x,y)=\left(\sum_{i=1}^{n}|x_i-y_i|^p\right)^{\frac{1}{p}}$$

- 其中 p 是一个参数。
- 闵可夫斯基距离是欧氏距离和曼哈顿距离的一般形式
- 当 $p=1$ 时，闵可夫斯基距离就是曼哈顿距离
- 当 $p=2$ 时，就是欧氏距离。

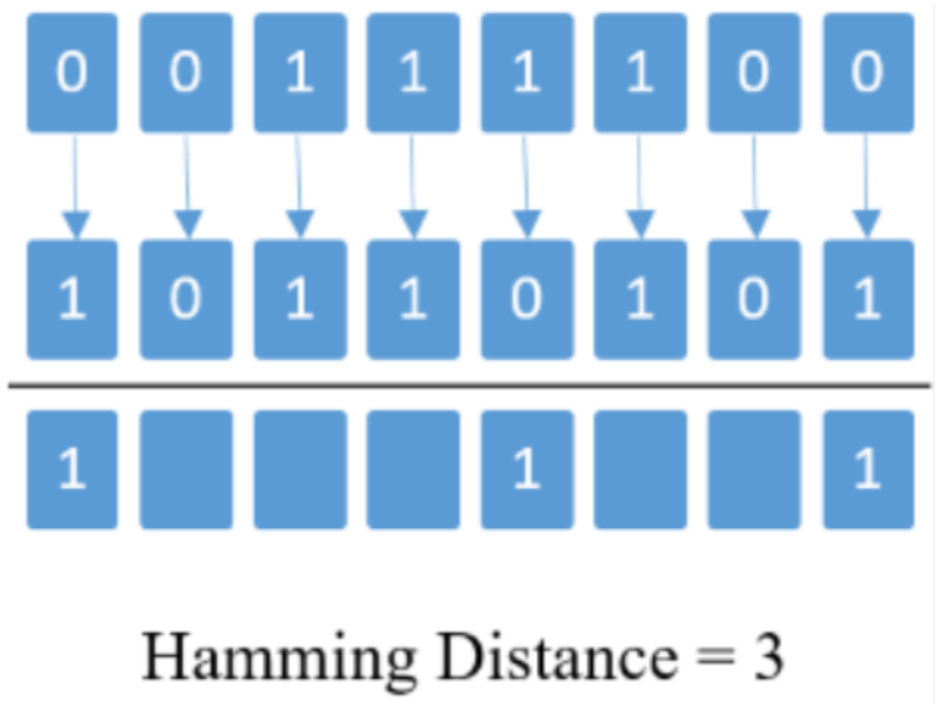


汉明距离（Hamming Distance）

- 汉明距离是比较两个相同长度的数据序列（如二进制序列、字符序列等），统计其中对应位置元素不同的数量。

$$d(x,y) = \sum_{i=1}^{n} I(x_i \neq y_i)$$

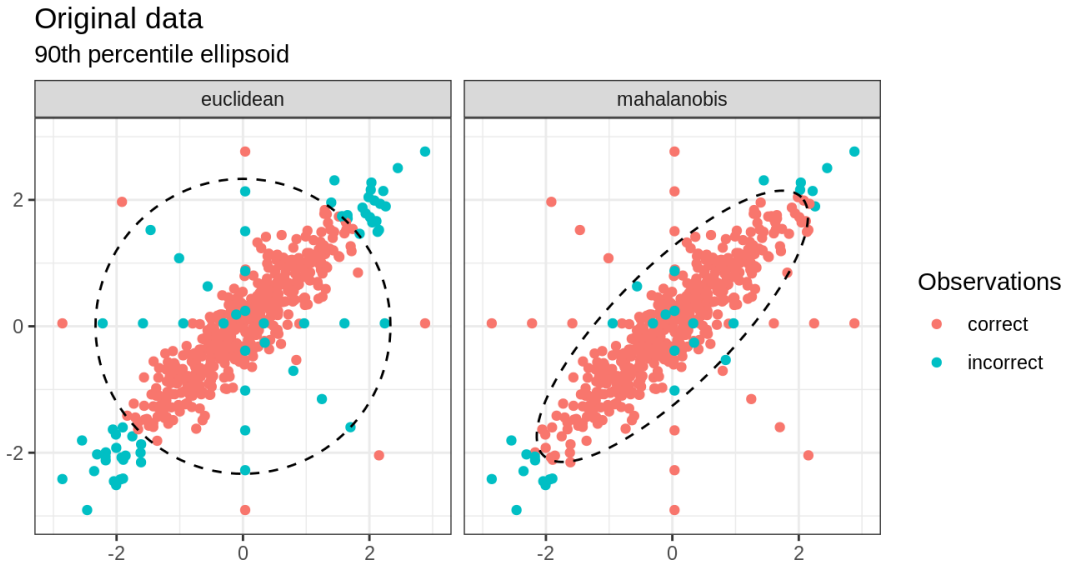
- 其中$I(\cdot)$是指示函数，当括号内条件成立时，I的值为1，反之为0。



马氏距离（Mahalanobis Distance）

- 马氏距离考虑数据的协方差结构的距离度量，用于衡量一个样本点与一个分布中心之间的距离，或者两个样本点在某个分布下的相对距离。
  - 其计算公式为：

$$d^2(x,y) = (x-y)^T \Sigma^{-1} (x-y)$$

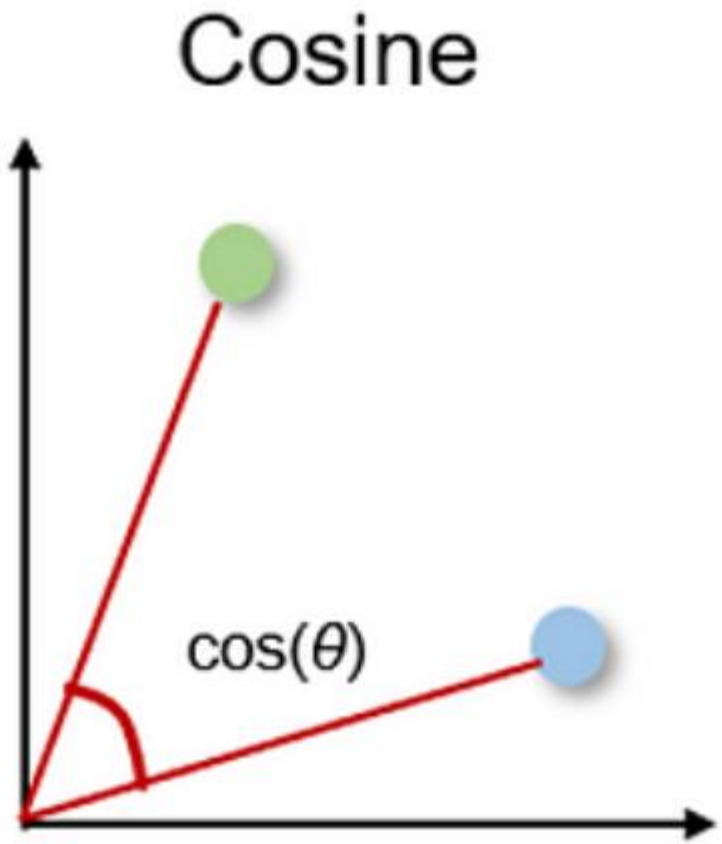


余弦相似度（Cosine Similarity）

- 通过计算两个向量的夹角余弦值来评估它们的相似度，
  - 对于两个非零向量 x 和 y，余弦相似度 s 定义为：

$$s(x,y) = \frac{x \cdot y}{|x||y|}$$

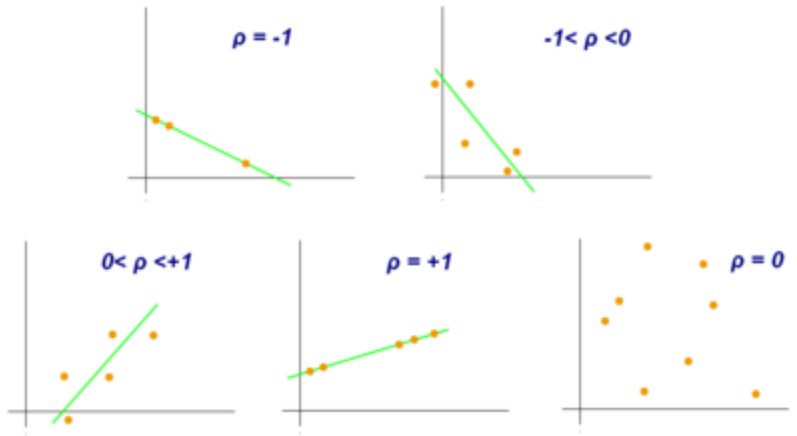
$$= \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$



相关系数（Pearson Correlation Coefficient）

- 最常用的相关系数是皮尔逊相关系数（Pearson Correlation Coefficient），
  - 计算公式为：

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}}$$



秩相关系数（Spearman's Rank Correlation）

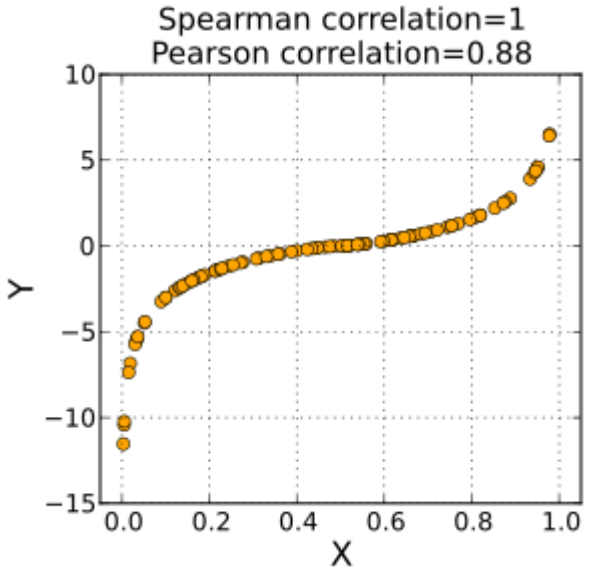
- 基于变量的秩次（排序后的位置）而不是变量的实际数值来计算相关性。
  - 先分别对两个变量X和Y的观测值进行排序，得到它们的秩次$R(X)$和$R(Y)$。然后计算秩次之间的差异

$$d_i = R(X_i) - R(Y_i)$$

  - 秩相关系数r_s的计算公式为

$$r_s = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)}$$

  - 其中n是观测值的数量。



秩相关系数（Spearman's Rank Correlation）

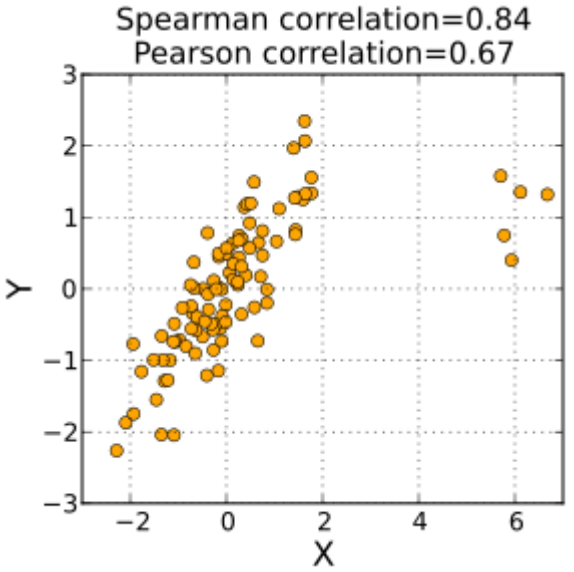
- 基于变量的秩次（排序后的位置）而不是变量的实际数值来计算相关性。
  - 先分别对两个变量X和Y的观测值进行排序，得到它们的秩次$R(X)$和$R(Y)$。然后计算秩次之间的差异

$$d_i = R(X_i) - R(Y_i)$$

  - 秩相关系数r_s的计算公式为

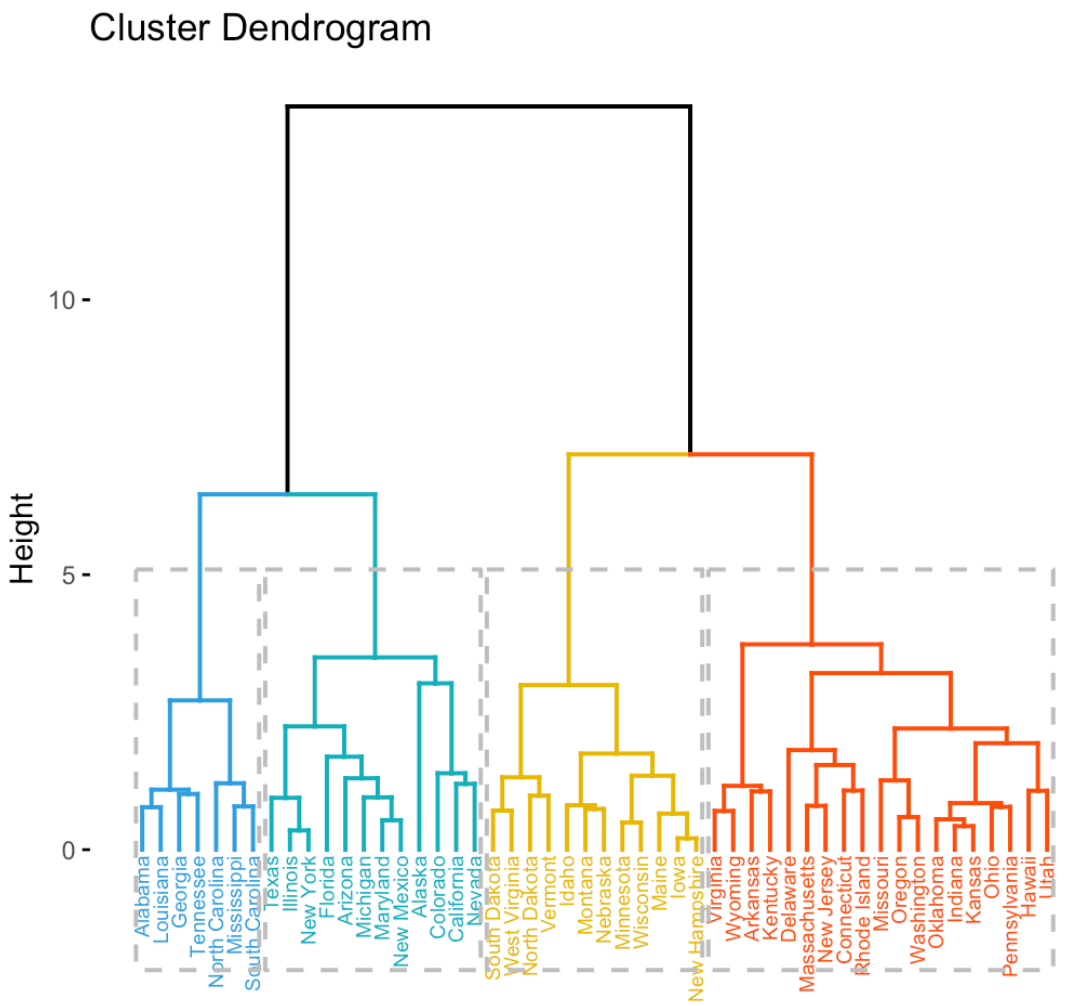
$$r_s = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)}$$

  - 其中n是观测值的数量。



层次聚类法

- 层次聚类法（Hierarchical Clustering），又称为系统聚类法。
- 它不需要事先指定聚类的数量，而是生成一个由层次结构组成的聚类树（Dendrogram），
- 通过不断地合并或者分裂数据子集来构建聚类层次，最终形成一个树形的聚类结构。
- 这个树可以刻画聚类的数量和聚类的层次。

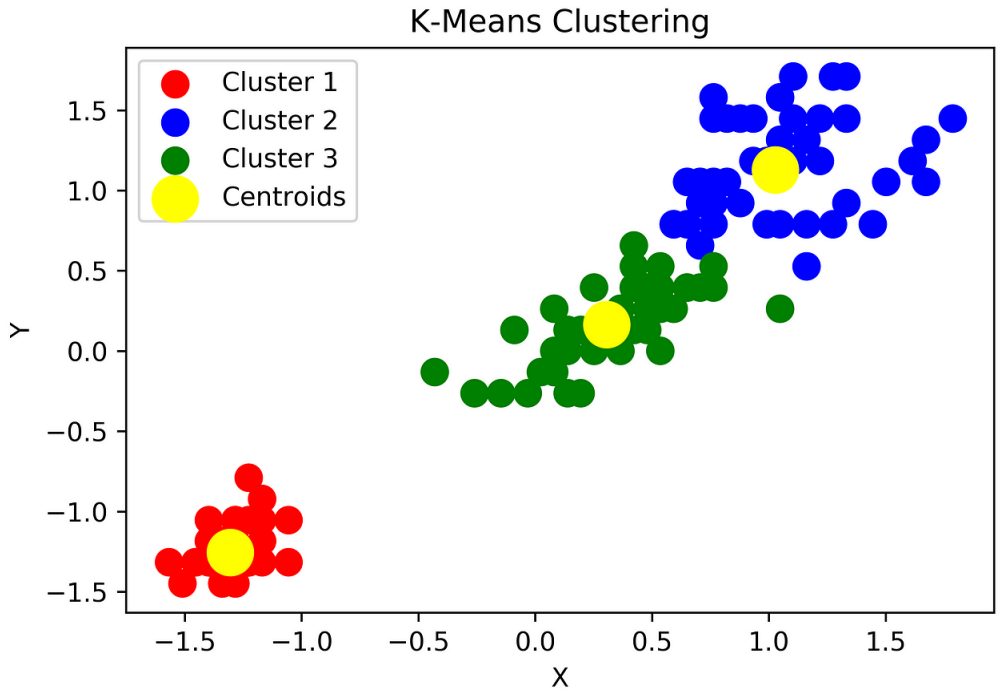


层次聚类法

- 在层次聚类法中，类的合并通常基于以下距离度量方法之一：
  - **最短距离法（Single Linkage）**：根据两个类中最近的成员之间的距离来合并类。
  - **最长距离法（Complete Linkage）**：根据两个类中最远成员之间的距离来合并类。
  - **类平均距离法（Average Linkage）**：使用两个类所有成员之间距离的平均值来合并类。
  - **中心距离法（Centroid Linkage）**：使用两个类的质心之间的距离来合并类。

K-means聚类方法

- K-means聚类法将数据分为多个类，使得类内的点尽可能相似，而类间的点尽可能不同。
- 通过迭代的方式，将数据点分配到距离其最近的聚类中心所在的类中，并且不断更新聚类中心的位置，直到聚类中心不再发生明显变化或者达到预设的迭代次数。
- K-means聚类方法是一种效率非常高的聚类方法，但需要事先确定类的数量。
- 如果K值选择不当，会导致聚类结果不符合实际情况。



K-means聚类方法

K-means聚类法的基本步骤：

1.初始化：选择一个初始的类数目k，然后随机选择k个数据点作为初始的类中心（质心）。

2.分配：对于数据集中的每个点，计算它与每个类中心的距离，并将其分配到最近的类中心，形成k个类。

3.更新：重新计算每个类的中心，通常是类内所有点的均值。

4.迭代：重复步骤2和3，直到满足以下停止条件之一：

1. 类中心不再显著变化，即连续两次迭代的类中心变化量小于某个预设的阈值。
2. 达到预设的迭代次数

K-means聚类方法

- 优点：
  - 简单、直观，易于理解和实现。
  - 计算效率高，适合处理大型数据集。

- 缺点：
  - 对初始类中心敏感，可能导致局部最优解。
  - 需要预先指定类的数目k，但在很多情况下k并不容易选择。
  - 对噪声和异常值敏感，可能会影响聚类结果。
  - 只能发现球形的类，对于非球形分布的数据可能不是最佳选择。