

# 1

Machine Learning

## 机器学习

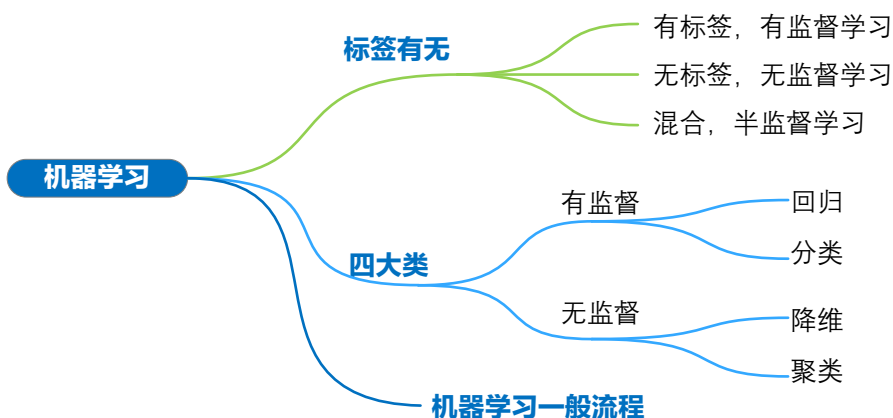
四大类算法：回归、分类、降维、聚类



卓越从来都不是偶然。卓越永远都是志存高远、百折不挠、有勇有谋的结果；它代表了明智之选。选择，而不是机会，决定了你的命运。

*Excellence is never an accident. It is always the result of high intention, sincere effort, and intelligent execution; it represents the wise choice of many alternatives. Choice, not chance, determines your destiny.*

—— 亚里士多德 (Aristotle) | 古希腊哲学家 | 384 ~ 322 BC



# 1.1 什么是机器学习?

鸢尾花书《编程不难》第 28 章回答过这个问题，下面我们把部分“答案”抄过来。

## 人工智能、机器学习、深度学习、自然语言处理

**人工智能** (Artificial Intelligence, AI) 的外延十分宽泛，泛指指计算机系统通过模拟人的思维和行为，实现类似于人的智能行为。人工智能领域包含了很多技术和方法，如机器学习、深度学习、自然语言处理、计算机视觉等。

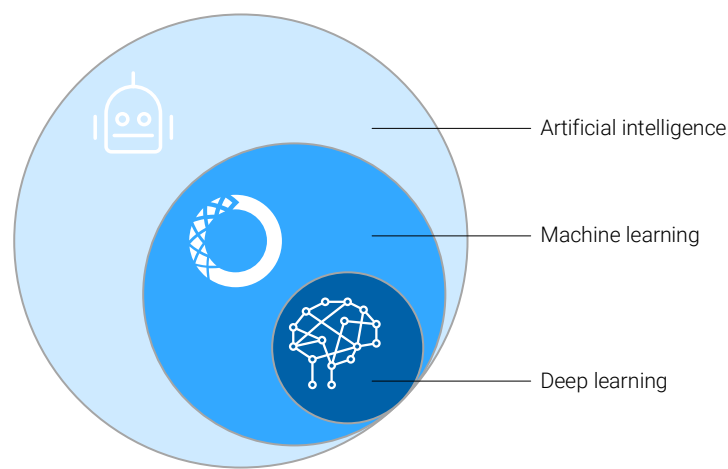


图 1. 人工智能、机器学习、深度学习

**机器学习** (Machine Learning, ML) 是人工智能的一个子领域，是通过计算机算法自动地从数据中学习规律，并用所学到的规律对新数据进行预测或者分类的过程。

机器学习算法的特点是，从样本数据中分析并获得某种规律，再利用这个规律对未知数据进行预测。它是涉及概率、统计、矩阵论、代数学、优化方法、数值方法、算法学等多领域的交叉学科。

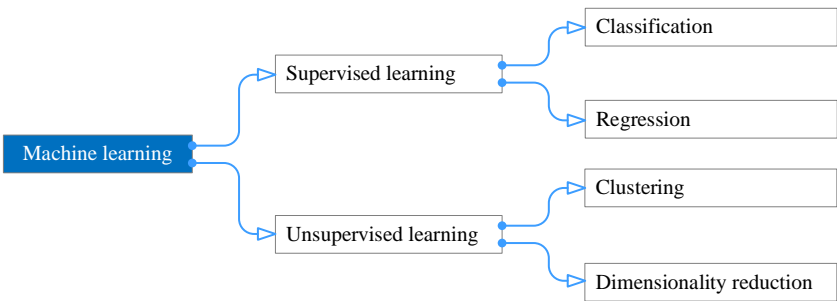


图 2. 机器学习分类

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。  
版权归清华大学出版社所有，请勿商用，引用请注明出处。  
代码及 PDF 文件下载：<https://github.com/Visualize-ML>  
本书配套微课视频均发布在 B 站——生姜 DrGinger: <https://space.bilibili.com/513194466>  
欢迎大家批评指教，本书专属邮箱：[jiang.visualize.ml@gmail.com](mailto:jiang.visualize.ml@gmail.com)

机器学习适合处理的问题有如下特征：(a) 大数据；(b) 黑箱或复杂系统，难以找到**控制方程** (governing equations)。机器学习需要通过数据的训练。

如图 2 所示，简单来说，机器学习可以分为以下两大类：

- ◀ **有监督学习** (supervised learning, 也叫监督学习, 训练有标签值样本数据并得到模型, 通过模型对新样本进行推断。
- ◀ **无监督学习** (unsupervised learning) 训练没有标签值的数据, 并发现样本数据的结构和分布。

此外, **半监督学习** 结合无监督学习和监督学习。

**深度学习** (Deep Learning, DL) 是一种机器学习的子领域, 它是通过建立多层**神经网络** (neural network) 模型, 自动地从原始数据中学习更高级别的特征和表示, 从而实现对复杂模式的建模和预测。

Python 中常用的深度学习工具有 TensorFlow、PyTorch、Keras 等, 这些工具不在本书讨论范围内。

**自然语言处理** (Natural Language Processing, NLP) 是计算机科学与人工智能领域的一个重要分支, 旨在通过计算机技术对人类语言进行分析、理解和生成。自然语言处理主要应用于自然语言文本的处理和分析, 如文本分类、情感分析、信息抽取、机器翻译、问答系统等。

### 有标签数据、无标签数据

根据输出值有无标签, 如图 3 所示, 数据可以分为**有标签数据** (labelled data) 和**无标签数据** (unlabelled data)。简单来说, 有标签数据对应**有监督学习** (supervised learning), 无标签数据对应**无监督学习** (unsupervised learning)。

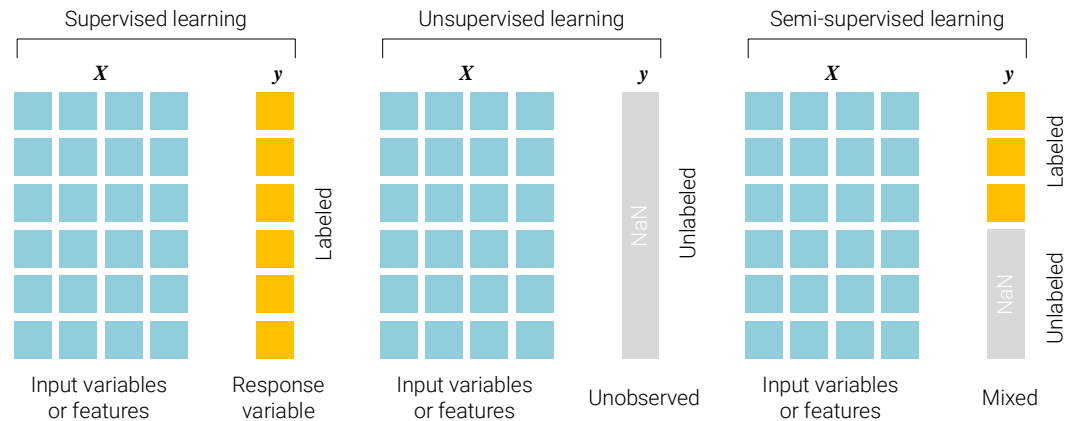


图 3. 根据有无标签分类数据

### 四大类算法

有监督学习中, 如果标签为连续数据, 对应的问题为**回归** (regression), 如图 4 (a)。如果标签为分类数据, 对应的问题则是**分类** (classification), 如图 4 (c)。简单来说, 分类问题与

本 PDF 文件为作者草稿, 发布目的为方便读者在移动终端学习, 终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有, 请勿商用, 引用请注明出处。

代码及 PDF 文件下载: <https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger: <https://space.bilibili.com/513194466>

欢迎大家批评指教, 本书专属邮箱: [jiang.visualize.ml@gmail.com](mailto:jiang.visualize.ml@gmail.com)

离散的输出相关，目标是将数据划分为不同的类别或标签；而回归问题与连续的输出相关，目标是预测数值型数据的结果。

无监督学习中，样本数据没有标签。如果目标是寻找规律、简化数据，这类问题叫做**降维** (dimensionality reduction)，比如**主成分分析** (Principal Component Analysis) 目的之一就是找到数据中占据主导地位的成分，如图 4 (b)。如果模型的目标是根据无标签数据特征将样本分成不同的组别，这种问题叫做**聚类** (clustering)，如图 4 (d)。

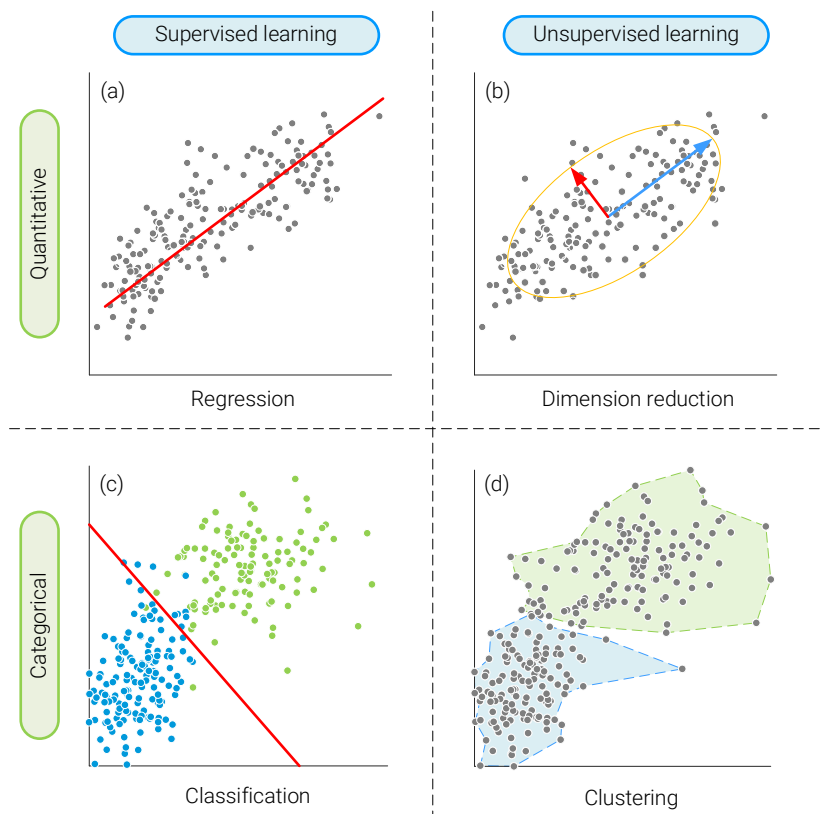


图 4. 根据数据是否有标签、标签类型细分机器学习算法

## 1.2 回归：找到自变量与因变量关系

回归问题是指根据已知的输入和输出数据，建立一个数学模型来预测输出值。给定一个输入，回归模型的目标是预测它的输出值，如房价预测、股票价格预测和天气预测等。

图 5 总结鸢尾花书系列丛书涉及的各种回归算法。

下面回顾回归算法中涉及的重要概念。

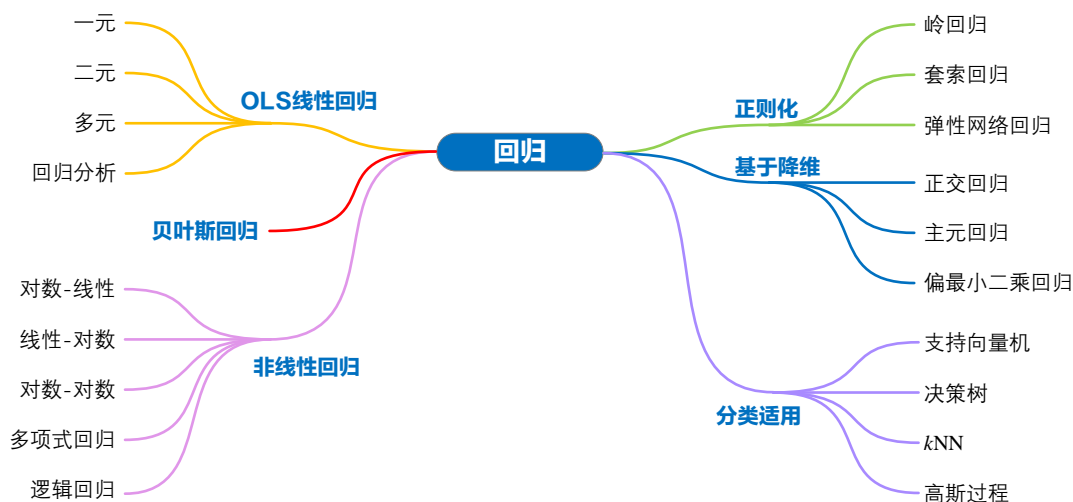


图 5. 回归方法分类

## 最小二乘算法

**线性回归** (linear regression) 通过构建一个线性模型来预测目标变量。最简单的线性回归算法是一元线性回归，多元线性回归则是利用多个特征来预测目标变量。线性回归离不开最小二乘法。

相信鸢尾花书读者对于**最小二乘** (Ordinary Least Squares, OLS) 线性回归已经烂熟于心。下面想强调几点。

首先，希望大家能够从多重视角理解 OLS 线性回归，比如优化 (图 6)、条件概率 (图 7)、几何 (图 8)、投影 (图 9)、数据、线性组合、SVD 分解、QR 分解、最大似然 MLE、最大后验 MAP 等视角。

此外，回归模型不能拿来就用，需要通过严格的回归分析。另外，要注意最小二乘线性回归的基本假设前提。

再提到 OLS 线性回归时，希望大家闭上眼睛，脑中不仅仅浮现各种多彩的图像，而且能够用 OLS 线性回归把代数、几何、线性代数、概率统计、优化等数学板块有机地联结起来！



丛书讲解 OLS 线性回归时可谓抽丝剥茧、层层叠叠。对于这些视角感到生疏的话，请回归《数学要素》第 24 章、《矩阵力量》第 9、25 两章、《统计至简》第 24 章。

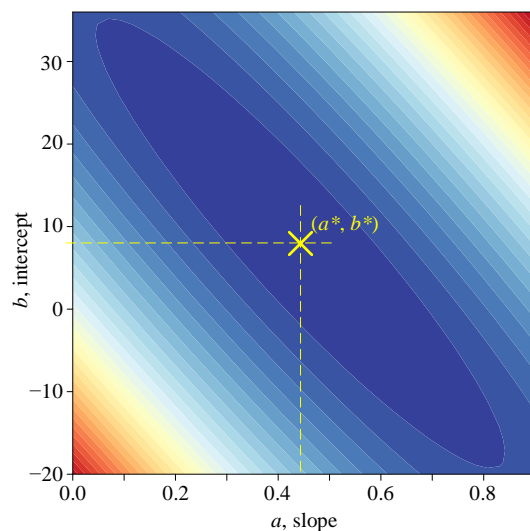


图 6. 一元 OLS 回归目标函数，图片来自《数学要素》第 24 章

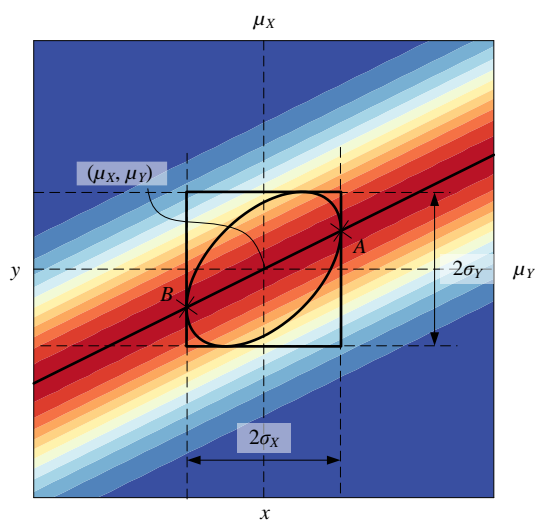
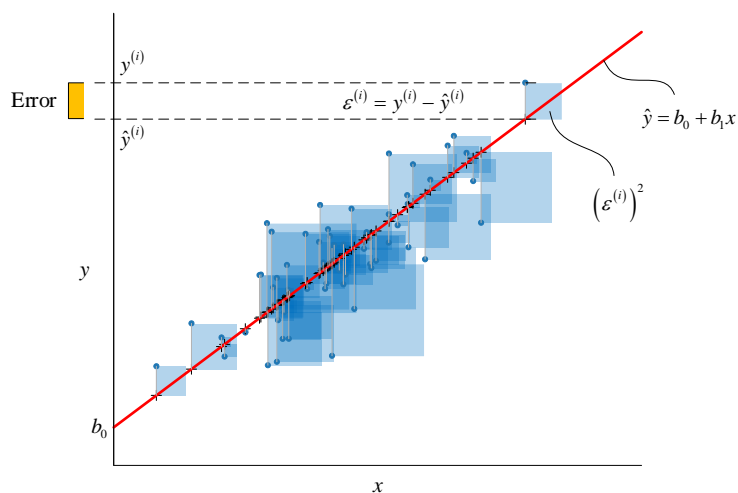


图 7. 条件期望视角看 OLS 线性回归，图片来自《统计至简》第 12 章



本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger: <https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：[jiang.visualize.ml@gmail.com](mailto:jiang.visualize.ml@gmail.com)

图 8. 残差平方和的几何意义，图片来自《统计至简》第 24 章

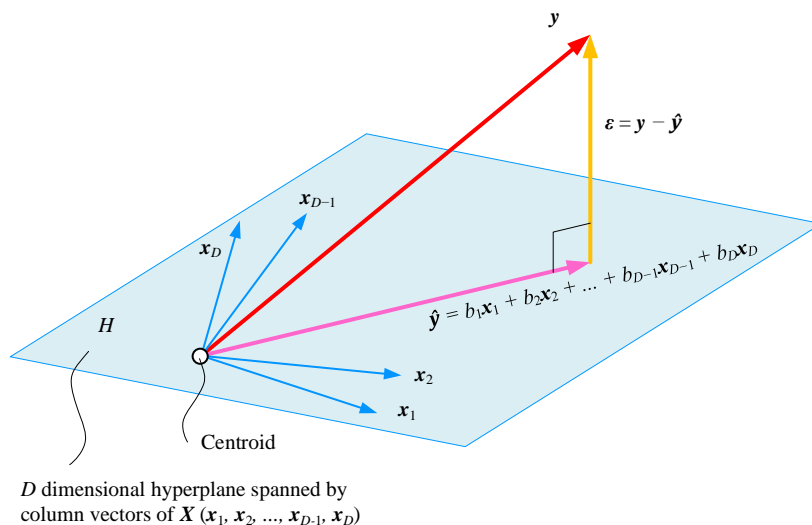
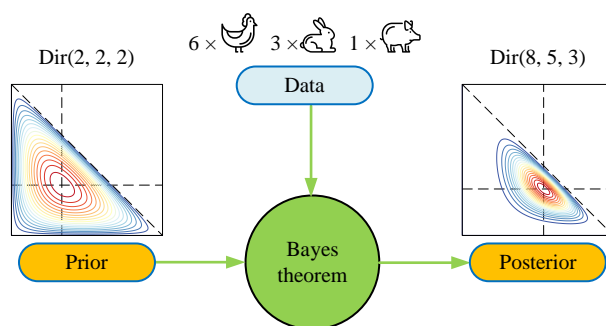


图 9. 投影角度解释多元最小二乘法线性回归，图片来自《数据有道》第 11 章

## 贝叶斯回归

**贝叶斯回归** (Bayesian regression) 是一种基于贝叶斯定理的回归算法，它可以用来估计连续变量的概率分布。贝叶斯回归可以视作一种特殊的贝叶斯推断。

**贝叶斯推断** (Bayesian inference) 把模型参数看作随机变量。根据主观经验和既有知识给出未知参数的概率分布，称为先验分布。从总体中得到样本数据后，根据贝叶斯定理，基于给定的样本数据，得出模型参数的后验分布。

图 10. 先验  $\text{Dir}(2, 2, 2)$  + 样本  $\rightarrow$  后验  $\text{Dir}(8, 5, 3)$ ，图片来自《统计至简》第 22 章

贝叶斯回归的优化问题对应最大后验 MAP。贝叶斯推断中，后验  $\propto$  似然  $\times$  先验，是最重要的关系，希望大家牢记。



欢迎大家回顾《统计至简》第 20、21、22 三章有关贝叶斯推断的内容。

## 非线性回归

**非线性回归** (nonlinear regression) 目标变量与特征之间的关系不是线性的。**多项式回归** (polynomial regression) 是非线性回归的一种形式，通过将特征的幂次作为新的特征来构建一个多项式模型。**逻辑回归** (logistic regression) 既是一种二分类算法，可以用于非线性回归。

此外，大家会发现  $k$ -NN、高斯过程算法完成的回归也都可以归类为非线性回归。

⚠ 请大家特别注意，逻辑回归不但可以用来回归，也可以用来分类。

## 正则化

**正则化** (regularization) 正则化通过向目标函数中添加惩罚项来避免模型的过拟合。常用的正则化方法有岭回归、Lasso 回归、弹性网络回归。岭回归通过向目标函数中添加 L2 惩罚项来控制模型复杂度。Lasso 回归通过向目标函数中添加 L1 惩罚项，它不仅能够控制模型复杂度，还可以进行特征选择。弹性网络是岭回归和 Lasso 回归的结合体，它同时使用 L1 和 L2 惩罚项。

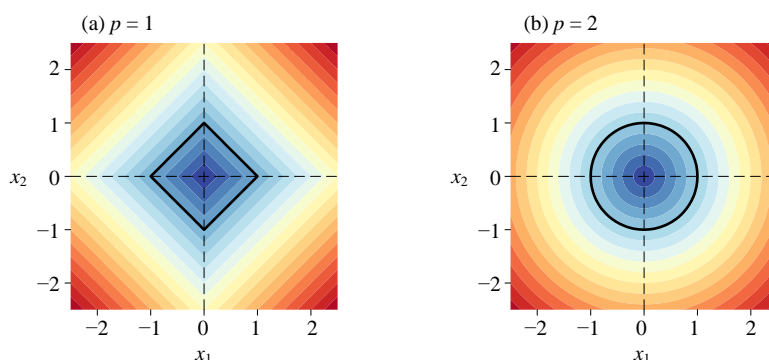


图 11. 两个范数示例

## 基于降维算法的回归

本书还要特别介绍两种基于主成分分析的回归方法——正交回归、主元回归。

平面上，最小二乘法线性回归 OLS 仅考虑纵坐标方向上误差，如图 12 (a) 所示；而正交回归 TLS 同时考虑纵横两个方向误差，如图 12 (b) 所示。

主元回归的因变量则来自于主成分分析结果。



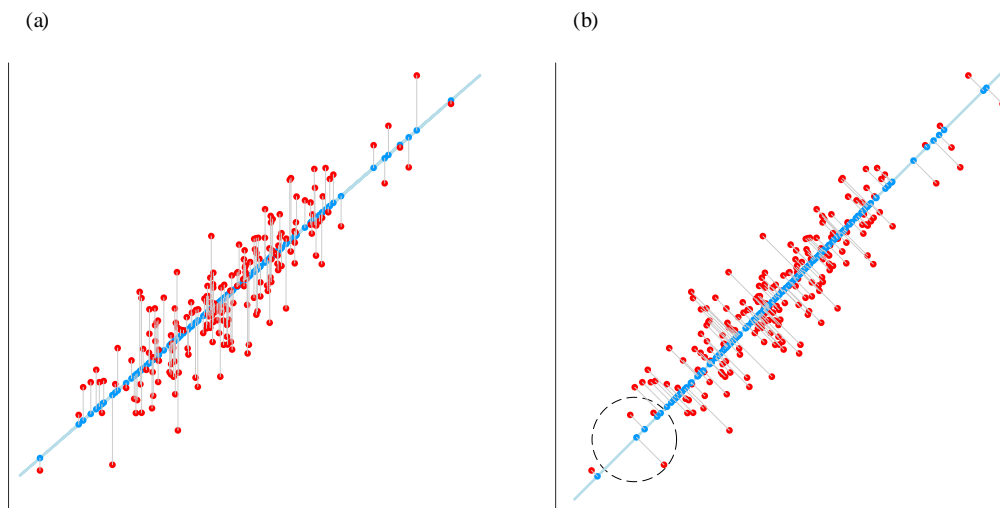


图 12. 对比 OLS 和 TLS 线性回归，图片来自《数据有道》第 18 章

### 基于分类算法的回归

实际上，监督学习的很多算法都兼顾分类、回归两项任务，比如逻辑回归、 $k$ -NN、支持向量机、高斯过程等等。 $k$ NN 算法是一种基于距离度量的分类算法，但也可以用于回归任务。**支持向量回归** (Support Vector Regression, SVR) 则是一种基于**支持向量机** (Support Vector Machine, SVM) 的回归算法。

## 1.3 分类：针对有标签数据

本书前文介绍过，**分类** (classification) 是**有监督学习** (supervised learning) 中的一类问题。分类是指根据给定的数据集，通过对样本数据的学习，建立分类模型来对新的数据进行分类的过程。

分类问题是指将数据集划分为不同的类别或标签。给定一个输入，分类模型的目标是预测它所属的类别，如垃圾邮件分类、图像识别和情感分析等。分类问题的输出是一个离散值或类别标签。

如图 13 所示，大家已经清楚鸢尾花数据集分三类 (setosa ●、versicolor ●、virginica ●)。

以**花萼长度** (sepal length)、**花萼宽度** (sepal width) 作为特征，大家如果采到一朵鸢尾花，测量后发现这朵花的花萼长度为 6.5 厘米，花瓣长度为 4.0 厘米，即图 13 中  $x$ ，又叫**查询点** (query point)。

根据已有数据，猜测这朵鸢尾花属于 setosa ●、versicolor ●、virginica ● 三类的哪一类可能性更大，这就是一个简单的分类问题。

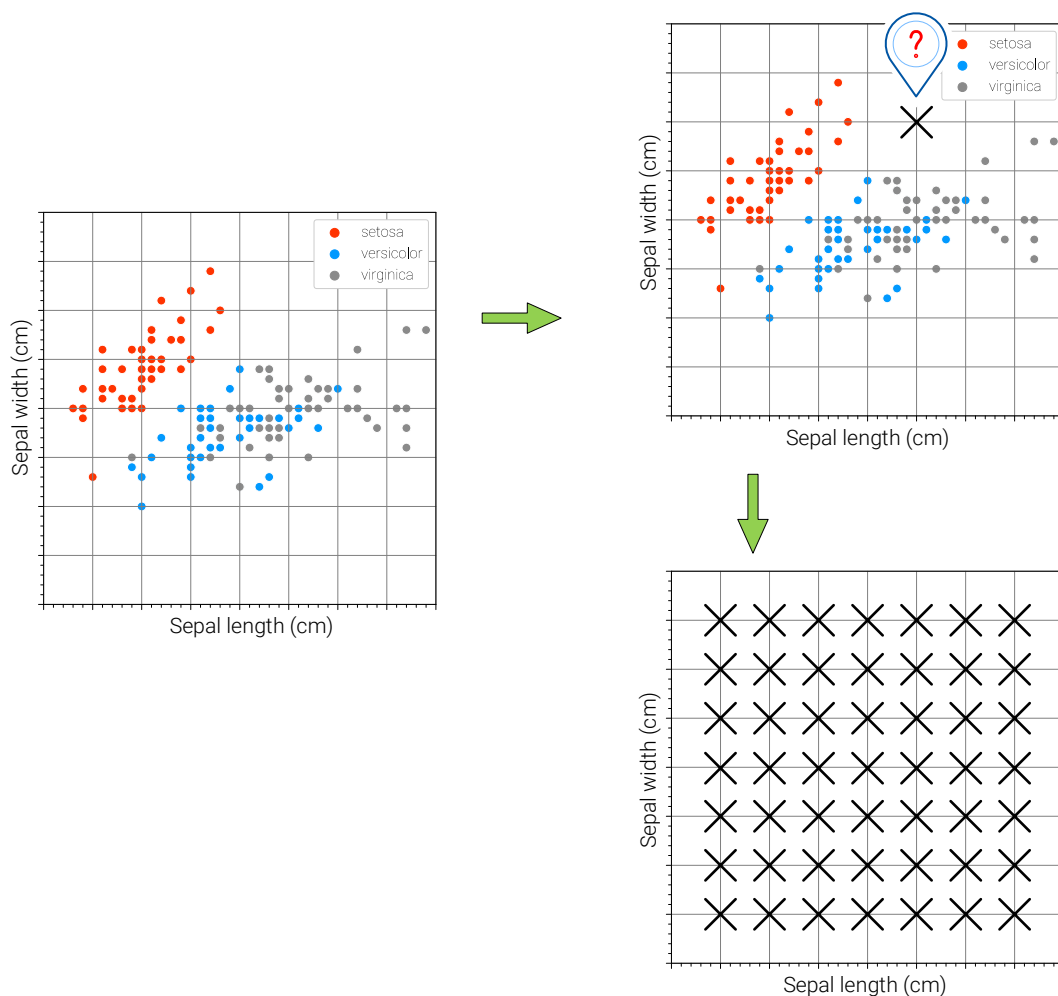


图 13. 用鸢尾花数据介绍分类算法

**决策边界** (decision boundary) 是分类模型在特征空间中划分不同类别的分界线或边界。通俗地说，决策边界就像是一道看不见的墙，把不同类别的数据点分隔开。

对于鸢尾花数据集，决策边界就是将 **setosa** ●、**versicolor** ●、**virginica** ● 这三类点“尽可能准确地”区分开的线或曲线。

在简单的情况下，决策边界可能是一条直线；但在复杂的问题中，决策边界可能是一条弯曲的曲线，甚至是多维空间中的超平面。

模型训练过程就是调整模型的参数，使得决策边界能够最好地拟合训练数据，并且在未见过的数据上也能表现良好。

要注意的是，决策边界的好坏直接影响分类模型的性能。一个好的决策边界能够很好地将数据分类，而一个不合适的决策边界可能导致模型预测错误。因此，选择合适的分类算法和调整模型参数是非常重要的，以获得有效的决策边界和准确的分类结果。

在机器学习中，分类是指根据给定的数据集，通过对样本数据的学习，建立分类模型来对新的数据进行分类的过程。下面简述一些常用的分类算法。

- ▶ **最近邻算法 (kNN)**: 基于样本的特征向量之间的距离进行分类预测，即找到与待分类数据距离最近的  $k$  个样本，根据它们的类别进行投票决策。
- ▶ **朴素贝叶斯算法 (Naive Bayes)**: 利用贝叶斯定理计算样本属于某个类别的概率，并根据概率大小进行分类决策。
- ▶ **支持向量机 (SVM)**: 利用间隔最大化的思想来进行分类决策，可以通过**核技巧 (kernel trick)** 将低维空间中线性不可分的样本映射到高维空间进行分类。
- ▶ **决策树算法 (Decision Tree)**: 通过对样本数据的特征进行划分，构建一个树形结构，从而实现对新数据的分类预测。

## 1.4 降维：降低数据维度，提取主要特征

**降维 (dimensionality reduction)** 是机器学习和数据分析领域中的重要概念，指的是将高维数据映射到低维空间中的过程。

在现实世界中，很多数据集都具有很高的维度，每个数据点可能包含大量特征或属性。然而，高维数据在处理和析时可能会面临一些问题，例如计算复杂度增加、维度诅咒、可视化困难等。而降维的目标是通过保留尽可能多的信息，将高维数据投影到一个更低维的子空间，以便更有效地处理和分析数据，减少计算负担，提高模型的性能和可解释性。

图 14 总结几种常见降维的算法。

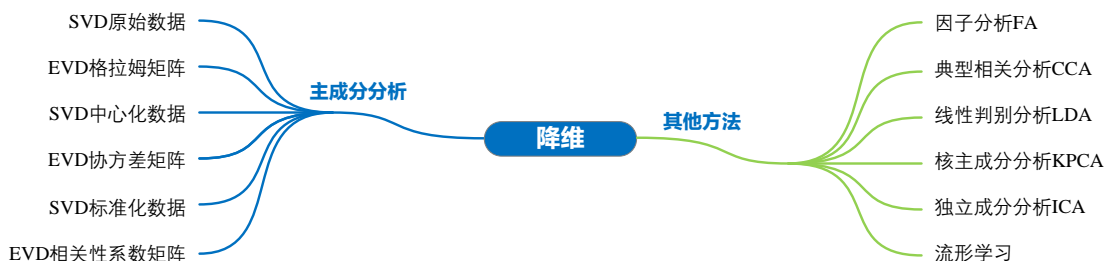


图 14. 常用降维算法

### 主成分分析

鸢尾花书对主成分分析着墨颇多。**主成分分析 (Principal Component Analysis, PCA)** 通过线性变换将高维数据映射到低维空间。利用特征值分解、奇异值分解都可以完成主成分分析。

PCA 将原始数据的特征转换为新的特征，这些新特征按照重要性递减排列。通过选取前面的几个主成分，可以实现对数据的压缩和可视化。主成分分析常用于数据预处理、数据可视化和特征提取等领域。它能够剔除冗余的特征信息，简化数据模型，提高模型的效率和准确性，是机器学习中非常重要的技术之一。

和 OLS 线性回归类似，主成分分析也可以从几何 (图 15)、投影、数据、线性组合、特征值分解、SVD 分解、优化、概率统计等视角来理解。

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微视频均发布在 B 站——生姜 DrGinger: <https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：[jiang.visualize.ml@gmail.com](mailto:jiang.visualize.ml@gmail.com)

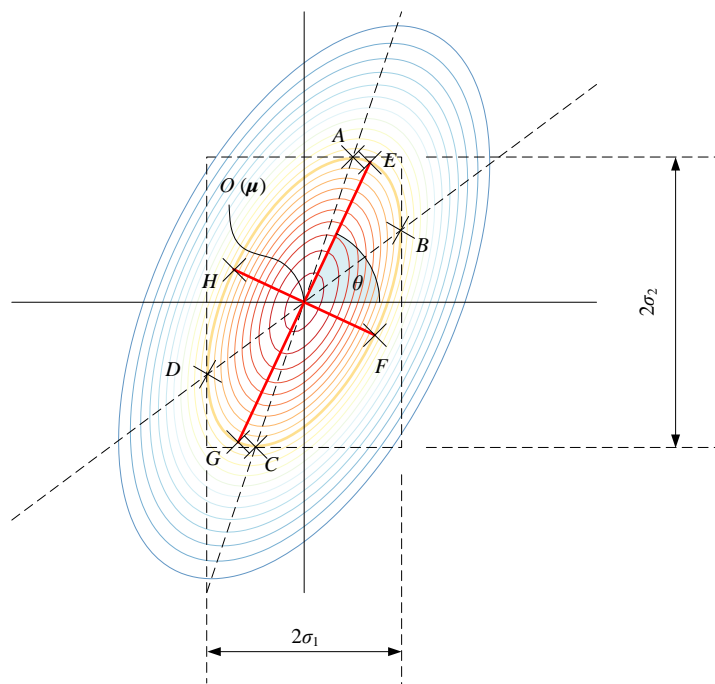


图 15. 主成分分析和椭圆的关系，图片来自《统计至简》第 25 章

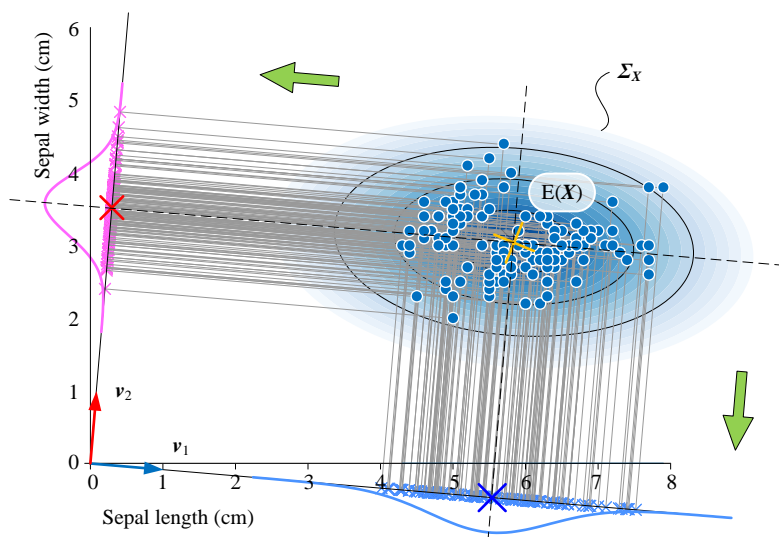


图 16. 投影视角看 PCA，图片来自《统计至简》第 14 章

## 增量 PCA

当 PCA 需要处理的数据矩阵过大，以至于内存无法支持，可以使用**增量主成分分析** (Incremental PCA, IPCA) 替代主成分分析。IPCA 分批处理输入数据，以便节省内存使用。Scikit-learn 中专门做增量 PCA 的函数为 `sklearn.decomposition.IncrementalPCA()`。

有关增量 PCA，大家可以参考下例。

[https://scikit-learn.org/stable/auto\\_examples/decomposition/plot\\_incremental\\_pca.html](https://scikit-learn.org/stable/auto_examples/decomposition/plot_incremental_pca.html)

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger: <https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：[jiang.visualize.ml@gmail.com](mailto:jiang.visualize.ml@gmail.com)

## 典型相关分析 CCA

**典型相关分析** (Canonical-Correlation Analysis, CCA) 也可以视作一种降维算法。典型相关分析是一种用于探究两组变量之间相关关系的统计方法，通常用于多个变量之间的关系分析。典型相关分析可以找出两组变量中最相关的线性组合，从而找到它们之间的相关性。

典型相关分析的目的是提取出两组变量之间的共性信息，用于预测和解释数据。CCA 也可以从几何、数据、优化、线性组合、统计几个不同视角来理解。

## 核主成分分析

**核主成分分析** (Kernel PCA) 是一种非线性的主成分分析方法，它通过使用核技巧将高维数据映射到低维空间中，从而提取出数据中的主要特征。与传统的 PCA 相比，Kernel PCA 可以更好地处理非线性数据，更准确地保留数据中的非线性结构。

可以这样理解，PCA 是 Kernel PCA 的特例。PCA 中用到的格拉姆矩阵、协方差矩阵、相关性系数矩阵都可以看成是不同线性核。

图 17 (a) 所示数据线性不可分，我们先用非线性映射把数据映射到高维空间，使其线性可分。利用 KPCA 之后的结果如图 17 (b)。这一点和支持向量机中的核技巧颇为类似。

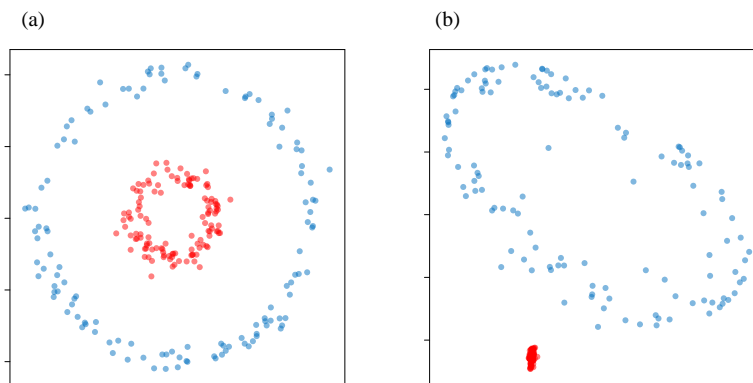


图 17. 核主成分分析

## 独立成分分析

**独立成分分析** (Independent Component Analysis, ICA) 是一种用于从混合信号中恢复原始信号的数学方法。ICA 通过将混合信号映射到独立的成分空间中，从而恢复原始信号。独立成分分析将一个多元信号分解成独立性最强的可加子成分。因此，独立成分分析常用来分离叠加信号。

图 18 比较 PCA 和 ICA 对同一组数据的分解结果。与 PCA 不同的是，ICA 假设原始信号是独立的，而 PCA 假设它们是正交关系。

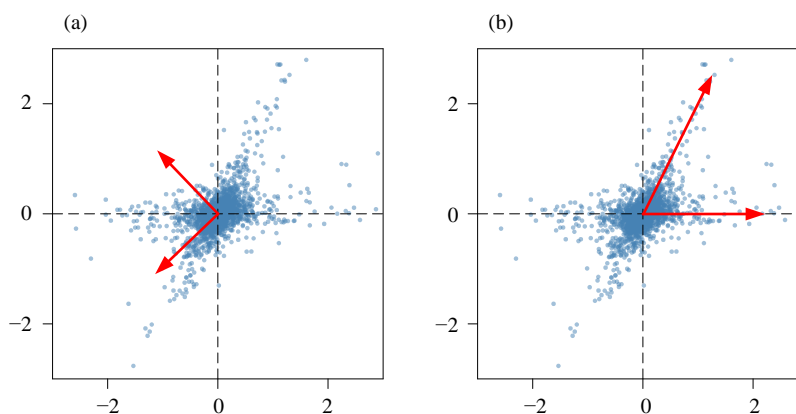


图 18. 比较 PCA 和 ICA

参考自如下示例，请大家自行学习 ICA：

[https://scikit-learn.org/stable/auto\\_examples/decomposition/plot\\_ica\\_vs\\_pca.html](https://scikit-learn.org/stable/auto_examples/decomposition/plot_ica_vs_pca.html)

有关独立成分分析算法原理，请大家参考：

<https://www.emerald.com/insight/content/doi/10.1016/j.aci.2018.08.006/full/html>

## 流形学习

空间的数据可能是按照某种规则“卷曲”，度量点与点之间的“距离”要遵循这种卷曲的趋势。换一种思路，我们可以像展开“卷轴”一样，将数据展开并投影到一个平面上，得到的数据如图 20 所示。在图 20 所示平面上， $A$  和  $B$  两点的“欧氏距离”更好地描述了两点的距离度量，因为这个距离考虑了数据的“卷曲”。

**流形学习** (manifold learning) 核心思想类似图 19 和图 20 所示展开“卷轴”的思想。流形学习用于发现高维数据中的低维结构，也是非线性降维的一种方法。与 PCA 不同的是，流形学习可以更好地处理非线性数据和局部结构，具有更好的可视化效果和数据解释性。

在 scikit-learn 中，流形学习的函数是 sklearn.manifold 模块中的 Isomap、LocallyLinearEmbedding、SpectralEmbedding 和 TSNE 等。其中，Isomap 使用测地线距离来保留流形上的全局结构，LocallyLinearEmbedding 使用局部线性嵌入来保留局部结构，SpectralEmbedding 使用谱分解来发现流形的嵌入表示，TSNE 使用高斯分布来优化样本的嵌入表示，用于可视化高维数据。这些函数提供了一种方便、高效、易于使用的流形学习工具，可帮助大家更好地理解数据结构和特征。本书不展开讲解流形学习，请大家自行探索。



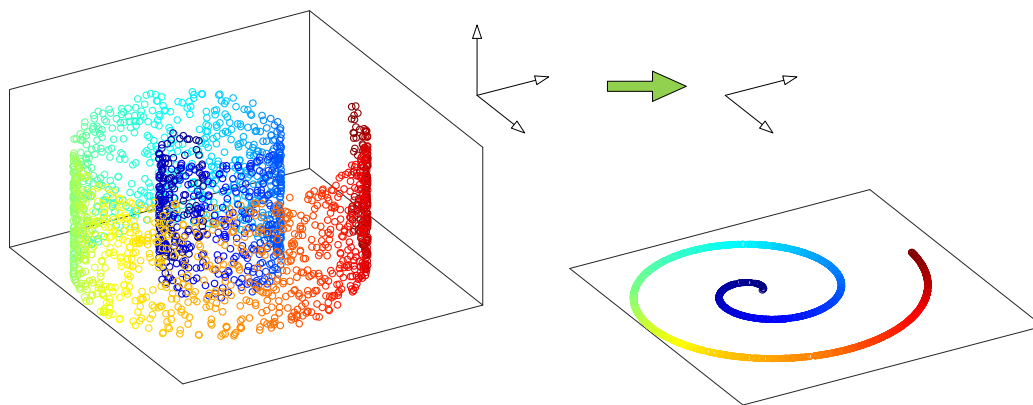


图 19. “卷曲”的数据

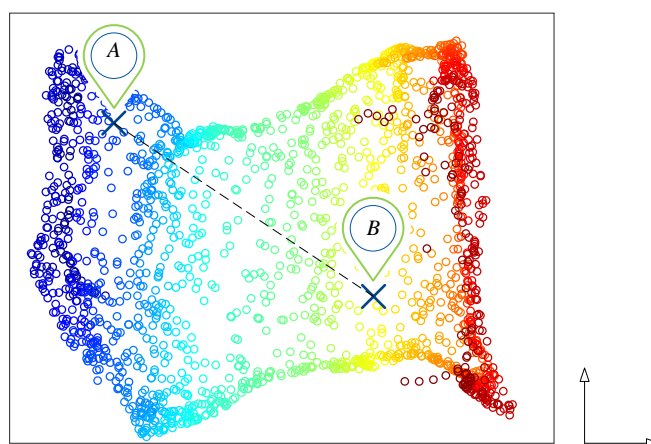


图 20. 展开“卷曲”的数据

想要深入了解 Scikit-learn 中的流形学习工具，请大家参考：

<https://scikit-learn.org/stable/modules/manifold.html>

如下这篇文献介绍了流形学习的数学基础，请大家参考：

<https://arxiv.org/pdf/2011.01307.pdf>

Scikit-learn 中更多有关降维工具，请大家参考：

<https://scikit-learn.org/stable/modules/decomposition.html>

## 1.5 聚类：针对无标签数据

本书前文介绍过，**聚类** (clustering) 是**无监督学习** (unsupervised learning) 中的一类问题。简单来说，聚类是指将数据集中相似的数据分为一类，以便更好地分析和理解数据。

如图 21 所示，删除鸢尾花数据集的标签，即 `target`，仅仅根据鸢尾花**花萼长度**（sepal length）、**花萼宽度**（sepal width）这两个特征上样本数据分布情况，我们可以将数据分成两**簇**（clusters）。

在机器学习中，决定将数据分成多少个簇是一个重要而且有挑战性的问题，通常称为聚类数目的选择或者簇数选择。不同的聚类算法可能需要不同的方法来确定合适的聚类数目。本章后文在介绍具体算法时，会介绍如何选择合适的簇数。

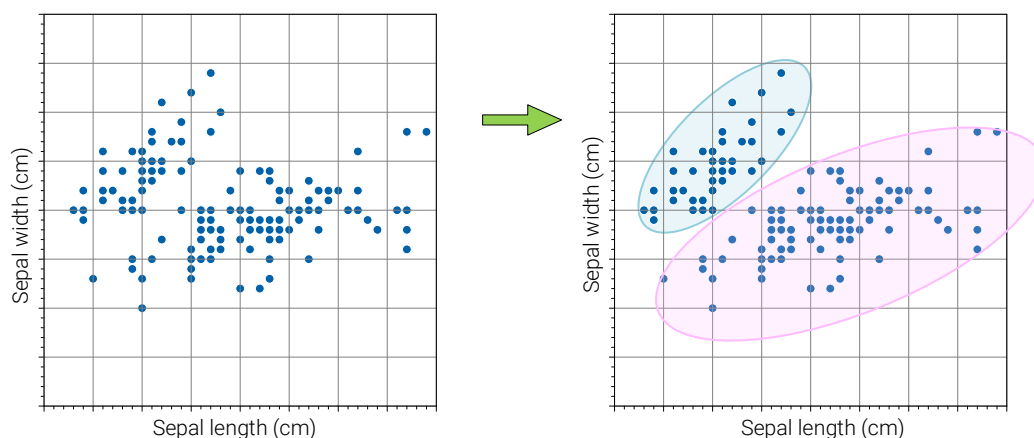


图 21. 用删除标签的鸢尾花数据介绍聚类算法

常用的聚类算法包括。

- ▶  **$k$  均值算法**（`kMeans`）：将样本分为  $k$  个簇，每个簇的中心点是该簇中所有样本点的平均值。
- ▶ **高斯混合模型**（Gaussian Mixture Model, GMM）：将样本分为多个高斯分布，每个高斯分布对应一个簇，采用 EM 算法进行迭代优化。
- ▶ **层次聚类算法**（Hierarchical Clustering）将样本分为多个簇，可以使用自底向上的凝聚层次聚类或自顶向下的分裂层次聚类。
- ▶ **DBSCAN**（Density-Based Spatial Clustering of Applications with Noise）是基于密度的聚类算法，可以自动发现任意形状的簇。
- ▶ **谱聚类算法**（Spectral Clustering）是基于样本之间的相似度来构造拉普拉斯矩阵，然后对其进行特征值分解来实现聚类。

大家在使用 Scikit-Learn 聚类算法时，会发现有些算法有 `predict()` 方法。

也就是说，如图 22 所示，已经训练好的模型，有可能你将全新的数据点分配到确定的簇中。有这种功能的聚类算法叫做**归纳聚类**（inductive clustering）。

本章后文要介绍的  $k$  均值聚类、高斯混合模型都属于归纳聚类。如图 22 所示，归纳聚类算法也有决策边界。这就意味着归纳聚类模型具有一定的泛化能力，可以推广到新的、之前未见过的数据。

不具备这种能力的聚类算法叫做**非归纳聚类**（non-inductive clustering）。



非归纳聚类只能对训练数据进行聚类，而不能将新数据点添加到已有的模型中进行预测。这意味着模型在训练时只能学习训练数据的模式，无法用于对新数据点进行簇分配。比如，层次聚类、DBSCAN 聚类都是非归纳聚类。

归纳聚类强调模型的泛化能力，可以适应新数据，而非归纳聚类则更侧重于建模训练数据内部的结构。

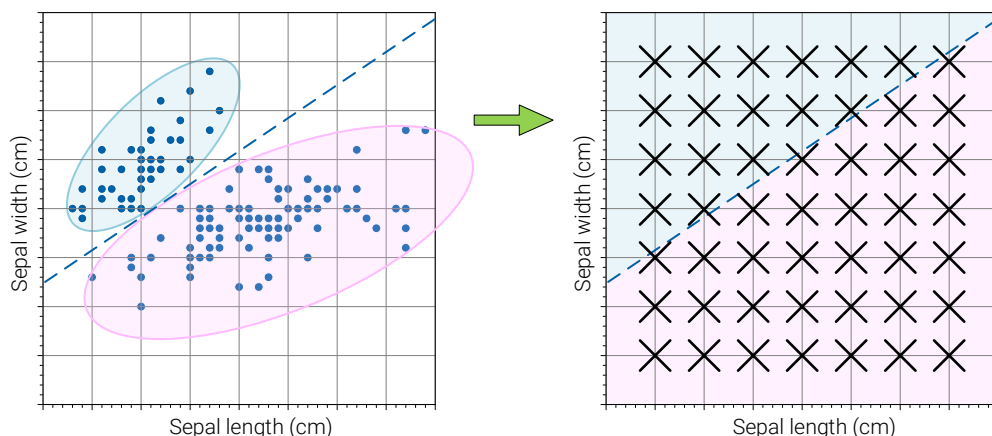


图 22. 归纳聚类算法

## 1.6 机器学习流程

图 23 所示为机器学习的一般流程。具体分步流程通常包括以下步骤：

- ◀ **收集数据**：从数据源获取数据集，这可能包括数据清理、去除无效数据和处理缺失值等。
- ◀ **特征工程**：对数据进行预处理，包括数据转换、特征选择、特征提取和特征缩放等。
- ◀ **数据划分**：将数据集划分为训练集、验证集和测试集等。训练集用于训练模型，验证集用于选择模型并进行调参，测试集用于评估模型的性能。
- ◀ **选择模型**：选择合适的模型，例如线性回归、决策树、神经网络等。
- ◀ **训练模型**：使用训练集对模型进行训练，并对模型进行评估，可以使用交叉验证等方法进行模型选择和调优。
- ◀ **测试模型**：使用测试集评估模型的性能，并进行模型的调整和改进。
- ◀ **应用模型**：将模型应用到新数据中进行预测或分类等任务。
- ◀ **模型监控**：监控模型在实际应用中的性能，并进行调整和改进。

以上是机器学习的一般分步流程，不同的任务和应用场景可能会有一些变化和调整。在实际应用中，还需要考虑数据的质量、模型的可解释性、模型的复杂度和可扩展性等问题。

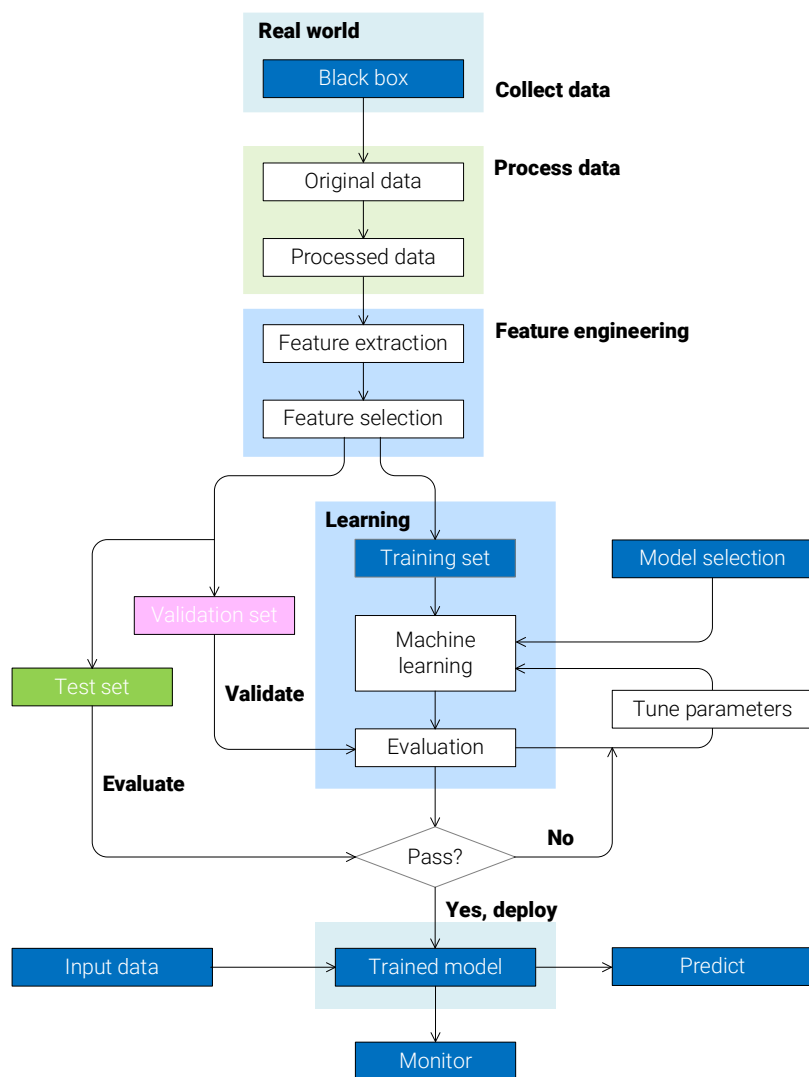


图 23. 机器学习一般流程

## 特征工程

从原始数据中最大化提取可用信息的过程就叫做**特征工程** (feature engineering)。特征很好理解，比如鸢尾花花萼长度宽度、花瓣长度宽度，人的性别、身体、体重等，都是特征。

特征工程是机器学习中非常重要的一个环节，指的是对原始数据进行特征提取、特征转换、特征选择和特征创造等一系列操作，以便更好地利用数据进行建模和预测。特征工程很好的混合了专业知识、数学能力。《数据有道》中介绍的离群值处理、缺失值处理、数据转换都属于特征工程范畴。

具体来说，特征工程包括以下方法。

- ◀ **特征提取** (Feature Extraction)：将原始数据转换为可用于机器学习算法的特征向量。注意，这个特征向量不是特征值分解中的特征向量。
- ◀ **特征转换** (Feature Transformation)：对原始特征进行数值变换，使其更符合算法的假设。例如，在回归问题中，可以对数据进行对数转换或指数转换等。
- ◀ **特征选择** (Feature Selection)：选择最具有代表性和影响力的特征。例如，可以使用相关性分析、PCA 等方法选择最相关或最重要的特征。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger：<https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：[jiang.visualize.ml@gmail.com](mailto:jiang.visualize.ml@gmail.com)

- ◀ **特征创造** (Feature Creation): 根据原始特征创造新的特征。例如, 在房价预测问题中, 可以根据房屋面积和房龄创建新的特征。
- ◀ **特征缩放** (Feature Scaling): 将特征缩放到相同的尺度或范围内, 避免某些特征对模型训练的影响过大。例如, 在神经网络中, 可以使用标准化或归一化等方法对数据进行缩放。

特征工程在机器学习中扮演着至关重要的角色, 它可以提高模型的精度、泛化能力和效率。在实际应用中, 需要根据具体问题选择合适的特征工程方法, 并不断尝试和改进以达到最佳效果。

相信大家都听过“**垃圾进, 垃圾出** (garbage in, garbage out, GIGO)”。这句话的含义很简单, 将错误的、无意义的输入数据输入计算机系统, 计算机自然也一定会输出错误、无意义的结果。在数据科学、机器学习领域, 很多时候数据扮演核心角色。以至于在数据分析建模时, 大部分的精力都花在了处理数据上。

有关特征工程, 大家可以参考这本开源专著:

<http://www.feat.engineering/>

Scikit-learn 也有大量特征工程工具, 请大家参考:

[https://scikit-learn.org/stable/modules/feature\\_selection.html](https://scikit-learn.org/stable/modules/feature_selection.html)

## 1.7 下一步学什么?

本书前文提到过《机器学习》这本书仅仅选取机器学习中 24 个话题, 分为四类——回归、分类、降维、聚类。每类算法不多不少, 仅仅分配 6 个话题。而机器学习是一个非常庞大系统, 《机器学习》限于篇幅不可能涉及所有话题。本章最后推荐一些“课后读物”, 供大家日后探索学习。

读完这本书, 大家可以学习如下资源, 了解如何在不同模型中做出选择。

[https://scikit-learn.org/stable/model\\_selection.html](https://scikit-learn.org/stable/model_selection.html)

有关深度学习, 推荐大家学习 *Dive into Deep Learning*, 英文开源图书地址为。

<https://d2l.ai/>

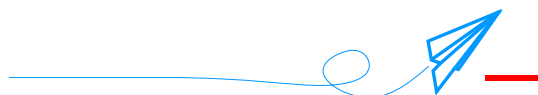
这本书也有开源中文版本。

<https://zh.d2l.ai/>

可以用来做自然语言处理的 Python 库有很多, 对于初学者大家可以从 NLTK 开始学起。NLTK 还提供如下学习手册, 很容易入门。

<https://www.nltk.org/book/>

此外, 本书最后还会给出一些供大家深入阅读的图书; 这些图书也是鸢尾花书的核心参考文献。



大家特别需要注意根据数据有无标签可以把机器学习分成两个大类——有监督学习、无监督学习。而有监督学习又可以细分为回归、分类。无监督学习则进一步分为降维、聚类。本章又聊了聊机器学习的一般流程以及特征工程。

下面开始本书 24 个话题的探索。

本 PDF 文件为作者草稿, 发布目的为方便读者在移动终端学习, 终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有, 请勿商用, 引用请注明出处。

代码及 PDF 文件下载: <https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger: <https://space.bilibili.com/513194466>

欢迎大家批评指教, 本书专属邮箱: [jiang.visualize.ml@gmail.com](mailto:jiang.visualize.ml@gmail.com)