

机器学习

讲师名称：李川

CanWay 嘉为数字咨询

数 字 化 人 才 培 养 先 行 者



目录

1

机器学习概述

2

机器学习基础理论

3

线性回归
(Linear Regression)

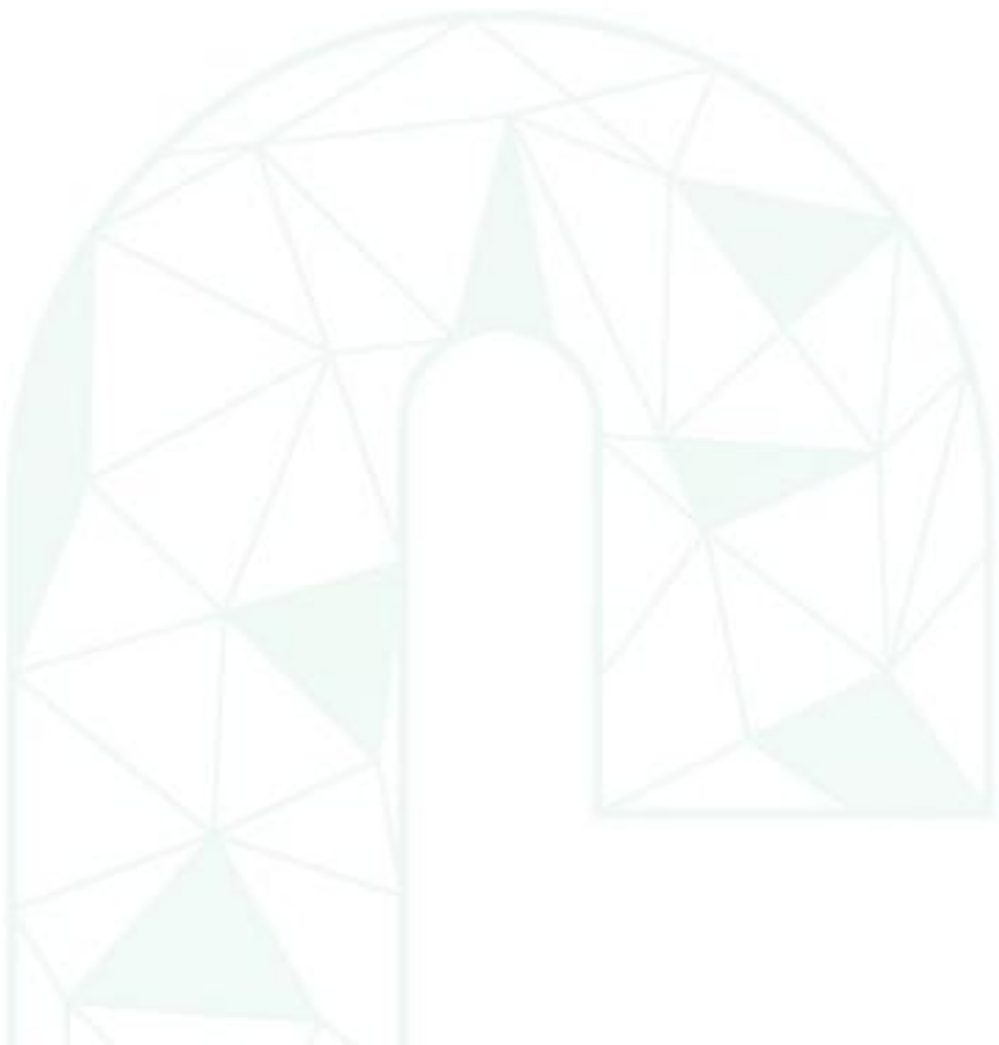
4

逻辑回归
(Logistic Regression)

5

决策树

CONTENTS



/01

机器学习概述

1.1 机器学习简介

机器学习 (Machine Learning) 是人工智能 (AI) 的一个分支, 它使计算机系统能够利用数据和算法自动学习和改进其性能。

机器学习是一个不断发展的领域, 它正在改变我们与技术的互动方式, 并为解决复杂问题提供了新的工具和方法。

机器学习是让计算机通过数据进行学习的一种技术, 广泛应用于各行各业。

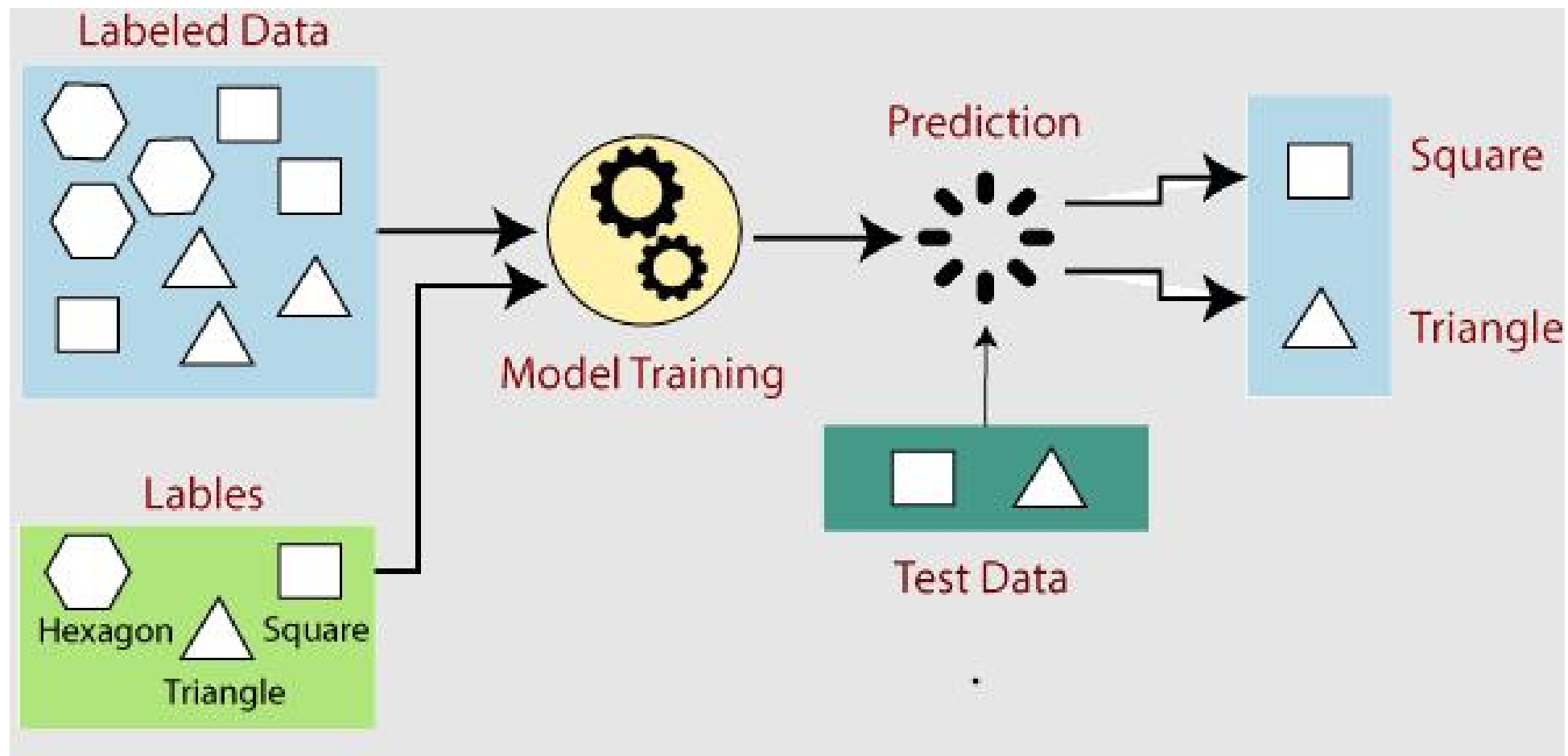
机器学习通过让计算机从大量数据中学习模式和规律来做出决策和预测。

首先, 收集并准备数据, 然后选择一个合适的算法来训练模型。

然后, 模型通过不断优化参数, 最小化预测错误, 直到能准确地对新数据进行预测。

最后, 模型部署到实际应用中, 实时做出预测或决策, 并根据新的数据进行更新。

1.2 机器学习的基本流程



1.2 机器学习的基本流程

1. Labeled Data (标记数据)：图中蓝色区域显示了标记数据，这些数据包括了不同的几何形状（如六边形、正方形、三角形）。
2. Model Training (模型训练)：在这个阶段，机器学习算法分析数据的特征，并学习如何根据这些特征来预测标签。
3. Test Data (测试数据)：图中深绿色区域显示了测试数据，包括一个正方形和一个三角形。
4. Prediction (预测)：模型使用从训练数据中学到的规则来预测测试数据的标签。在图中，模型预测了测试数据中的正方形和三角形。
5. Evaluation (评估)：预测结果与测试数据的真实标签进行比较，以评估模型的准确性。

1.3 机器学习完整工作流程

1. 数据收集

收集数据：这是机器学习项目的第一步，涉及收集相关数据。数据可以来自数据库、文件、网络或实时数据流。

数据类型：可以是结构化数据（如表格数据）或非结构化数据（如文本、图像、视频）。

2. 数据预处理

清洗数据：处理缺失值、异常值、错误和重复数据。

特征工程：选择有助于模型学习的最相关特征，可能包括创建新特征或转换现有特征。

数据标准化/归一化：调整数据的尺度，使其在同一范围内，有助于某些算法的性能。

3. 选择模型

确定问题类型：根据问题的性质（分类、回归、聚类等）选择合适的机器学习模型。

选择算法：基于问题类型和数据特性，选择一个或多个算法进行实验。

1.3 机器学习完整工作流程

4. 训练模型

划分数据集：将数据分为训练集、验证集和测试集。

训练：使用训练集上的数据来训练模型，调整模型参数以最小化损失函数。

验证：使用验证集来调整模型参数，防止过拟合。

5. 评估模型

性能指标：使用测试集来评估模型的性能，常用的指标包括准确率、召回率、F1分数等。

交叉验证：一种评估模型泛化能力的技术，通过将数据分成多个子集进行训练和验证。

1.3 机器学习完整工作流程

6. 模型优化

调整超参数：超参数是学习过程之前设置的参数，如学习率、树的深度等，可以通过网格搜索、随机搜索或贝叶斯优化等方法来调整。

特征选择：可能需要重新评估和选择特征，以提高模型性能。

7. 部署模型

集成到应用：将训练好的模型集成到实际应用中，如网站、移动应用或软件中。

监控和维护：持续监控模型的性能，并根据新数据更新模型。

8. 反馈循环

持续学习：机器学习模型可以设计为随着时间的推移自动从新数据中学习，以适应变化。

1.4 机器学习类型

1. 监督学习 (Supervised Learning)
2. 无监督学习 (Unsupervised Learning)
3. 强化学习 (Reinforcement Learning)

1.4.1 监督学习 (Supervised Learning)

有监督学习是指利用带标签的训练数据（输入 + 对应输出）训练模型，让模型学习 “输入到输出的映射规律”，最终实现对新数据的预测（输出标签）。

核心特点

数据要求：训练数据必须包含 “输入特征” 和 “对应标签”（如 “图片 + 猫 / 狗标签” “房屋特征 + 房价标签”）。

学习目标：学习输入与输出的明确对应关系，最终实现 “给定新输入，输出预测标签”。

评价方式：通过预测标签与真实标签的差异（如准确率、均方误差）评估模型效果。

1.4.1 监督学习 (Supervised Learning)

典型算法

根据输出标签的类型，分为两类：

分类任务（标签是离散值，如 “是 / 否” “类别 A/B/C” ）：

逻辑回归、支持向量机 (SVM)、决策树、随机森林、神经网络（如用于图像分类的 CNN）。

回归任务（标签是连续值，如 “价格” “温度” ）：

线性回归、多项式回归、梯度提升树 (GBDT)、神经网络（如用于预测房价的 MLP）。

应用场景：

分类场景：垃圾邮件识别、图像识别、疾病诊断。

回归场景：房价预测、销量预测、气温预测。

1.4.2 无监督学习 (Unsupervised Learning)

无监督学习是指仅利用无标签的训练数据（只有输入，无输出）训练模型，让模型自主发现数据中隐藏的规律（如结构、分布、聚类）。

核心特点

数据要求：训练数据仅包含“输入特征”，无标签（如“一堆用户行为数据”“一批未分类的图片”）。

学习目标：挖掘数据的内在结构（如聚类、降维、异常检测），不直接预测标签。

评价方式：难以用“准确率”衡量，通常通过“聚类紧凑性”“降维后信息保留度”等间接指标评估。

1.4.2 无监督学习 (Unsupervised Learning)

典型算法

聚类算法：将相似数据聚为一类（无预设类别），如 K-Means（指定聚类数量）、DBSCAN（基于密度聚类）。

降维算法：在保留关键信息的前提下，减少数据维度（方便可视化或简化计算），如 PCA（主成分分析）、t-SNE（适合高维数据可视化）。

异常检测：识别与多数数据模式不符的“异常值”，如孤立森林、自编码器。

应用场景

聚类场景：用户分群、商品分类。

降维场景：高维数据可视化、数据压缩。

异常检测场景：欺诈识别、设备故障预警。

1.4.3 强化学习 (Reinforcement Learning)



强化学习通过与环境互动，智能体在试错中学习最佳策略，以最大化长期回报。每次行动后，系统会收到奖励或惩罚，来指导行为的改进。

核心特点

从与环境的互动中 “试错学习”

1.4.3 强化学习 (Reinforcement Learning)

典型算法

基于价值 (Value-Based) 的方法。

基于策略 (Policy-Based) 的方法。

Actor-Critic 方法 (结合 “价值方法” 和 “策略方法” 的优势)。

应用场景

游戏领域: AI 玩电子游戏 (如 Atari、《星际争霸》)、棋盘游戏 (围棋、国际象棋)。

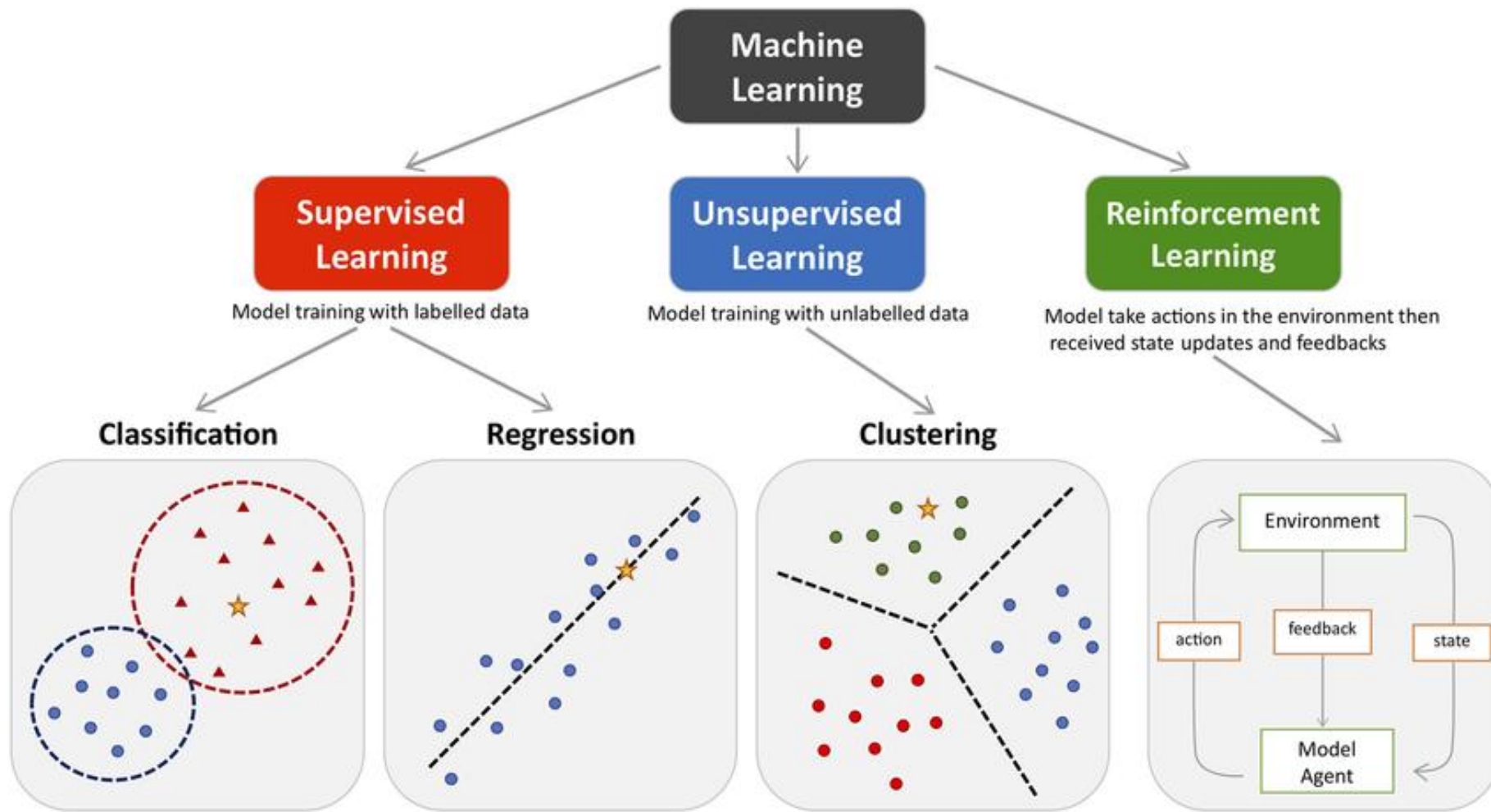
机器人控制: 机械臂抓取物体、无人机自主导航、人形机器人行走。

推荐系统: 动态调整推荐策略, 最大化用户长期点击率 (如视频平台推荐)。

自动驾驶: 决策车辆加速、刹车、转向, 平衡安全与效率。

资源调度: 数据中心算力分配、电网负荷调节, 优化长期资源利用率。

1.4.4 三种机器学习模型图示





/02

机器学习基础理论

2.1 训练集、测试集和验证集

训练集 (Training Set) : 训练集是用于训练机器学习模型的数据集，它包含输入特征和对应的标签（在监督学习中）。模型通过学习训练集中的数据来调整参数，逐步提高预测的准确性。

测试集 (Test Set) : 测试集用于评估训练好的模型的性能。测试集中的数据不参与模型的训练，模型使用它来进行预测，并与真实标签进行比较，帮助我们了解模型在未见过的数据上的表现。

验证集 (Validation Set) : 验证集用于在训练过程中调整模型的超参数（如学习率、正则化参数等）。它通常被用于模型调优，帮助选择最佳的模型参数，避免过拟合。验证集的作用是对模型进行监控和调试。

2.2 特征 (Features) 和标签 (Labels)

特征 (Features)：特征是输入数据的不同属性，模型使用这些特征来做出预测或分类。例如，在房价预测中，特征可能包括房子的面积、地理位置、卧室数量等。

标签 (Labels)：标签是机器学习任务中的目标变量，模型要预测的结果。对于监督学习任务，标签通常是已知的。例如，在房价预测中，标签就是房子的实际价格。

2.3 模型 (Model) 与算法 (Algorithm)

模型 (Model) : 模型是通过学习数据中的模式而构建的数学结构。它接受输入特征, 经过一系列计算和转化, 输出一个预测结果。常见的模型有线性回归、决策树、神经网络等。

算法 (Algorithm) : 算法是实现机器学习的步骤或规则, 它定义了模型如何从数据中学习。常见的算法有梯度下降法、随机森林、K近邻算法等。算法帮助模型调整其参数以最小化预测误差。

2.4 过拟合与欠拟合

过拟合 (Overfitting)：过拟合是指模型在训练数据上表现非常好，但在测试数据上表现很差。这通常发生在模型复杂度过高、参数过多，导致模型"记住"了训练数据中的噪声或偶然性，而不具备泛化能力。过拟合的模型无法有效应对新数据。

欠拟合 (Underfitting)：欠拟合是指模型在训练数据上和测试数据上都表现不佳，通常是因为模型过于简单，无法捕捉数据中的复杂模式。欠拟合的模型无法从数据中学习到有用的规律。

解决方法：

过拟合：可以通过简化模型、增加训练数据或使用正则化等方法来缓解。

欠拟合：可以通过增加模型复杂度或使用更复杂的算法来改进。

2.5 训练与测试误差

训练误差 (Training Error) : 训练误差是模型在训练数据上的表现, 反映了模型是否能够很好地适应训练数据。如果训练误差很大, 可能说明模型不够复杂, 欠拟合; 如果训练误差很小, 可能说明模型太复杂, 容易过拟合。

测试误差 (Test Error) : 测试误差是模型在未见过的数据上的表现, 反映了模型的泛化能力。测试误差应当与训练误差相匹配, 若测试误差远高于训练误差, 通常是过拟合。

2.6 评估指标

根据任务的不同，机器学习模型的评估指标也不同。以下是常用的一些评估指标：

准确率（Accuracy）：分类任务中，正确分类的样本占总样本的比例。

精确率（Precision）和召回率（Recall）：主要用于分类任务中处理不平衡数据集，精确率衡量的是被模型预测为正类的样本中，有多少是真正的正类；召回率衡量的是所有实际正类中，有多少被模型正确识别为正类。

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{Recall} = \frac{TP}{TP+FN}$$

F1 分数：精确率与召回率的调和平均数，用于综合考虑模型的表现。

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

均方误差（MSE）：回归任务中，预测值与真实值之间差异的平方的平均值。



/03

线性回归 (Linear Regression)

3.1 线性回归简介

线性回归 (Linear Regression) 是机器学习中最基础且广泛应用的算法之一。

线性回归 (Linear Regression) 是一种用于预测连续值的最基本的机器学习算法，它假设目标变量 y 和特征变量 x 之间存在线性关系，并试图找到一条最佳拟合直线来描述这种关系。

$$y = w * x + b$$

其中：

y 是预测值

x 是特征变量

w 是权重 (斜率)

b 是偏置 (截距)

3.1 线性回归简介

线性回归的目标是找到最佳的 w 和 b ，使得预测值 y 与真实值之间的误差最小。常用的误差函数是均方误差 (MSE):

$$\text{MSE} = 1/n * \sum (y_i - y_{\text{pred}_i})^2$$

其中:

y_i 是实际值。

y_{pred_i} 是预测值。

n 是数据点的数量。

目标是通过调整 w 和 b ，使得 MSE 最小化。

3.2 求解线性回归

最小二乘法

最小二乘法是一种常用的求解线性回归的方法，最小二乘法的目标：最小化平方误差。

定义损失函数（平方误差和）为：

$$L(w, b) = \sum_{i=1}^n (\hat{y}_i - y_i)^2 = \sum_{i=1}^n (w \cdot x_i + b - y_i)^2$$

最小二乘法的目标是找到 w 和 b ，使 $L(w, b)$ 最小。

3.2 求解线性回归

求解最优参数（数学推导）

通过求导并令导数为 0，可得到 w 和 b 的解析解：

计算偏置 b

对 $L(w, b)$ 关于 b 求导并令其为 0，化简后得： $b = \bar{y} - w \cdot \bar{x}$

其中 $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ 是 x 的均值， $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$ 是 y 的均值。

计算权重 w

对 $L(w, b)$ 关于 w 求导并令其为 0，代入 b 的表达式后得：

$$w = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

3.3 线性回归——最小二乘法实例

例5-1最小二乘法求解线性回归参数

```
import numpy as np
import matplotlib.pyplot as plt
# 全局设置字体（适用于所有图表）
plt.rcParams["font.family"] = ["SimHei"] # 支持中文的字体
plt.rcParams["axes.unicode_minus"] = False # 解决负号显示问题
# 1. 生成样本数据
np.random.seed(42)
x = np.linspace(0, 10, 50) # 输入特征
y = 2 * x + 5 + np.random.normal(0, 1, 50) # 真实关系: y=2x+5, 添加噪声
# 2. 手动实现最小二乘法
n = len(x)
x_mean = np.mean(x)
y_mean = np.mean(y)
# 计算w和b
numerator = np.sum((x - x_mean) * (y - y_mean)) # 分子: 协方差之和
denominator = np.sum((x - x_mean) ** 2) # 分母: x的方差之和
w = numerator / denominator
b = y_mean - w * x_mean
print(f"手动计算的参数: w={w:.2f}, b={b:.2f}") # 接近真实值w=2, b=5
```

3.3 线性回归——最小二乘法实例

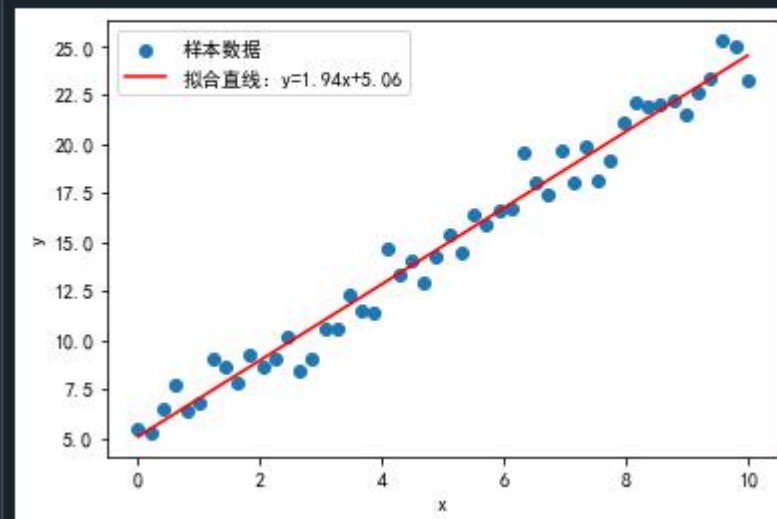
例5-1最小二乘法求解线性回归参数

```
# 3. 用sklearn验证（库函数实现）
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(x.reshape(-1, 1), y) # 输入需为二维数组
print(f"sklearn计算的参数: w={model.coef_[0]:.2f},
      b={model.intercept_:.2f}")
```

4. 可视化拟合结果

```
y_pred = w * x + b # 预测值
plt.scatter(x, y, label="样本数据")
plt.plot(x, y_pred, 'r-', label=f"拟合直线:
y={w:.2f}x+{b:.2f}")
plt.xlabel("x")
plt.ylabel("y")
plt.legend()
plt.show()
```

手动计算的参数: $w=1.94$, $b=5.06$
sklearn计算的参数: $w=1.94$, $b=5.06$



3.3 线性回归评估模型性能

在线性回归模型中，`model.score(X, y)`计算的指标是决定系数（Coefficient of Determination），通常记为 R^2 （R-squared）。它是衡量线性回归模型拟合效果的核心指标之一，其计算公式：

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

理想情况下， R^2 越接近 1，说明模型拟合效果越好（残差越小，解释能力越强）。

若 $R^2 = 0$ ，表示模型的预测效果等同于直接用均值 预测（无解释力）。

极端情况下， R^2 可能为负数（当模型拟合效果差于直接用均值预测时，常见于非线性数据强行用线性模型拟合）。

计算模型性能程序如下：

```
# 计算模型得分
score = model.score(x.reshape(-1,1), y)
print("模型得分:", score)
```




/04

逻辑回归 (Logistic Regression)

4.1 逻辑回归简介

逻辑回归 (Logistic Regression) 是一种广泛应用于分类问题的统计学习方法, 尽管名字中带有"回归", 但它实际上是一种用于二分类或多分类问题的算法。

逻辑回归通过使用逻辑函数 (也称为 Sigmoid 函数) 将线性回归的输出映射到 0 和 1 之间, 从而预测某个事件发生的概率。

逻辑回归广泛应用于各种分类问题, 例如:

垃圾邮件检测 (是垃圾邮件/不是垃圾邮件)

疾病预测 (患病/不患病)

客户流失预测 (流失/不流失)

4.2 逻辑回归实例

鸢尾花数据集 (Iris Dataset) , 每个类别样本数: 各 50 个, 分布均衡, 无严重类别不平衡问题

特征名称	描述	取值范围 (大致)
花萼长度 (sepal length)	花萼 (花瓣外的绿色部分) 的长度	4.3–7.9 cm
花萼宽度 (sepal width)	花萼的宽度	2.0–4.4 cm
花瓣长度 (petal length)	花瓣 (彩色部分) 的长度	1.0–6.9 cm
花瓣宽度 (petal width)	花瓣的宽度	0.1–2.5 cm
类别 (species)	鸢尾花的种类	setosa / versicolor / virginic

4.2 逻辑回归实例

例5-2 鸢尾花分类

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
# 加载数据集
iris = load_iris()
X = iris.data[:, :2] # 只使用前两个特征
y = (iris.target != 0) * 1 # 将目标转化为二分类问题
print(X.shape)
# 划分训练集和测试集
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# 创建逻辑回归模型
model = LogisticRegression()

# 训练模型
model.fit(X_train, y_train)
```

4.2 逻辑回归实例

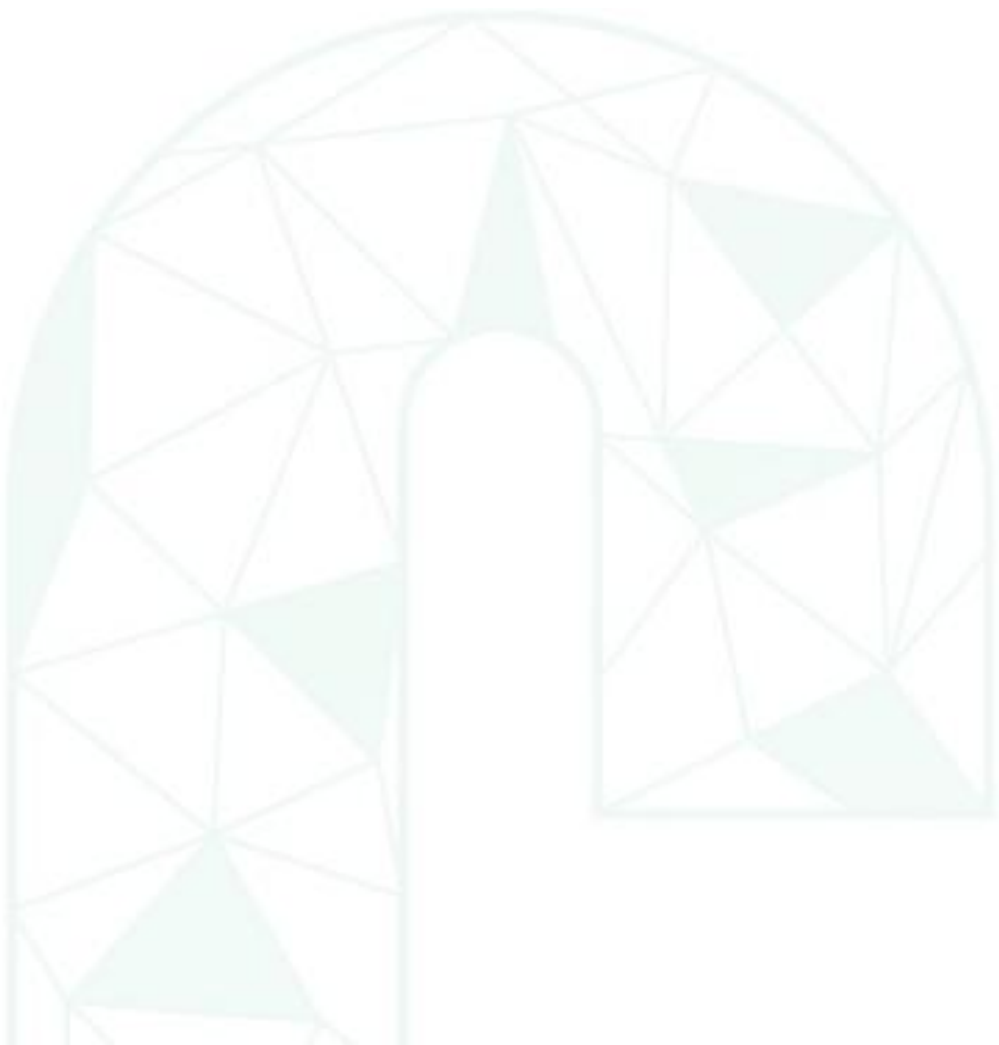
例5-2 鸢尾花分类

```
# 预测测试集
y_pred = model.predict(X_test)

# 计算准确率
accuracy = accuracy_score(y_test, y_pred)
print(f"模型准确率: {accuracy:.2f}")

# 混淆矩阵
conf_matrix = confusion_matrix(y_test, y_pred)
print("混淆矩阵:")
print(conf_matrix)

# 分类报告
class_report = classification_report(y_test, y_pred)
print("分类报告:")
print(class_report)
```



/05

决策树

5.1 决策树简介

决策树是一种常用的机器学习算法，既可以用于分类任务，也能用于回归任务。

核心原理

通过对数据特征的不断分割，构建出一棵包含根节点、内部节点、叶节点的树。

根节点：代表整个数据集。

内部节点：表示根据某个特征进行的分割判断。

叶节点：对应最终的决策结果（分类的类别或回归的数值）。

分割过程遵循“让同一子集中的数据尽可能相似”的原则，常用信息增益、基尼指数等指标来选择最优分割特征和分割点。

5.2 决策树算法流程

1. 训练阶段（构建树）

数据准备：收集带标签的数据集（特征 + 目标类别），确保数据格式符合模型输入要求（如数值型或编码后的类别型）。

特征选择：从所有特征中选择最优分割特征，目标是让分割后的子数据集“纯度”更高（同类样本更集中）。

常用指标：

信息增益（基于信息熵，ID3 算法）；

信息增益比（C4.5 算法，解决信息增益偏向多值特征的问题）；

基尼指数（CART 算法，计算更高效）。

5.2 决策树算法流程

1. 训练阶段（构建树）

递归分割：

以当前数据集为根节点，用最优特征分割为多个子节点（每个子节点对应该特征的一个取值或区间）；

对每个子节点重复特征选择和分割过程，直到满足停止条件（如子节点样本全为同一类、样本数小于阈值、无更多特征可分）。

剪枝：避免过拟合（树过于复杂，对训练数据拟合过好但泛化能力差）。

方式：

预剪枝：在树生长过程中提前停止（如限制树的深度、叶子节点最小样本数）；

后剪枝：先构建完整树，再移除对泛化能力无增益的分支（如通过验证集评估分支是否保留）。

5.2 决策树算法流程

2. 预测阶段

对于新样本，从根节点开始，根据样本的特征值逐层进入对应的子节点，最终到达叶节点，叶节点的类别（或多数样本类别）即为预测结果。

5.3 决策树实例

例5-3 鸢尾花分类

```
# 导入库
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score

# 1. 准备数据（以鸢尾花数据集为例）
data = load_iris()
X = data.data # 特征（花萼长度、宽度等）
y = data.target # 目标类别（3种鸢尾花）
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 2. 构建决策树模型（CART算法，默认基尼指数）
clf = DecisionTreeClassifier(
    criterion='gini', # 特征选择指标（'gini'或'entropy'）
    max_depth=5, # 预剪枝：限制树的最大深度
    min_samples_leaf=5 # 预剪枝：叶子节点最小样本数
)
```

5.3 决策树实例

例5-3 鸢尾花分类

```
# 3. 训练模型
clf.fit(X_train, y_train)
# 4. 预测与评估
y_pred = clf.predict(X_test)
print("准确率: ", accuracy_score(y_test, y_pred)) # 输出预测准确率

# 5. 可视化决策树
from sklearn.tree import plot_tree
import matplotlib.pyplot as plt

plt.figure(figsize=(12, 8))
plot_tree(
    clf,
    feature_names=data.feature_names,
    class_names=data.target_names,
    filled=True, # 按类别填充颜色
    rounded=True # 圆角边框
)
plt.show()
```

5.4 综合实例

例5-4 使用线性回归和决策树构建核能源应用发展预测模型，以历史数据预测未来核能源发电量.

数据说明

假设数据集包含以下特征（模拟数据）：

year: 年份（1990-2020）

nuclear_plants: 核电站数量

investment: 年度投资（十亿美元）

policy_support: 政策支持力度（0-10 分）

carbon_price: 碳价（美元 / 吨）

目标变量: nuclear_generation（核能源发电量，太瓦时）

5.3 决策树实例

例5-4 使用线性回归和决策树构建核能源应用发展预测模型，以历史数据预测未来核能源发电量。

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_squared_error, r2_score

# -----
# 1. 生成模拟数据
# -----
np.random.seed(42)
years = np.arange(1990, 2021)
n = len(years)
```

5.3 决策树实例

例5-4 使用线性回归和决策树构建核能源应用发展预测模型，以历史数据预测未来核能源发电量。

```
data = pd.DataFrame({
    'year': years,
    'nuclear_plants': np.linspace(100, 150, n).astype(int) + np.random.randint(-5, 5, n),
    'investment': np.linspace(20, 50, n) + np.random.normal(0, 3, n),
    'policy_support': np.linspace(3, 7, n) + np.random.normal(0, 1, n),
    'carbon_price': np.linspace(10, 40, n) + np.random.normal(0, 5, n)
})

# 模拟目标变量（核发电量）：与特征正相关
data['nuclear_generation'] = (
    0.5 * data['nuclear_plants'] +
    1.2 * data['investment'] +
    3.0 * data['policy_support'] +
    0.8 * data['carbon_price'] +
    0.02 * (data['year'] - 1990) +
    np.random.normal(0, 5, n) # 噪声
)
```

5.3 决策树实例

例5-4 使用线性回归和决策树构建核能源应用发展预测模型，以历史数据预测未来核能源发电量.

```
# -----  
# 2. 数据拆分  
# -----  
X = data.drop(['year', 'nuclear_generation'], axis=1) # 特征（排除年份，可用作时间轴）  
y = data['nuclear_generation'] # 目标变量  
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)  
  
# -----  
# 3. 模型训练与评估  
# -----  
# 线性回归  
lr = LinearRegression()  
lr.fit(X_train, y_train)  
y_pred_lr = lr.predict(X_test)  
  
# 决策树回归  
dt = DecisionTreeRegressor(max_depth=5, random_state=42) # 限制深度避免过拟合  
dt.fit(X_train, y_train)  
y_pred_dt = dt.predict(X_test)
```


5.3 决策树实例

例5-4 使用线性回归和决策树构建核能源应用发展预测模型，以历史数据预测未来核能源发电量。

```
# 评估指标
print("线性回归：")
print(f"均方误差 (MSE)：{mean_squared_error(y_test, y_pred_lr):.2f}")
print(f"决定系数 (R²)：{r2_score(y_test, y_pred_lr):.2f}\n")
print("决策树回归：")
print(f"均方误差 (MSE)：{mean_squared_error(y_test, y_pred_dt):.2f}")
print(f"决定系数 (R²)：{r2_score(y_test, y_pred_dt):.2f}")
# -----
# 4. 预测未来（示例：2021-2030年）
# -----
future_years = np.arange(2021, 2031)
future_data = pd.DataFrame({
    'nuclear_plants': np.linspace(150, 180, 10).astype(int), # 假设核电站数量增长
    'investment': np.linspace(50, 80, 10), # 投资增加
    'policy_support': np.linspace(7, 9, 10), # 政策支持加强
    'carbon_price': np.linspace(40, 60, 10) # 碳价上涨
})
# 预测未来发电量
future_pred_lr = lr.predict(future_data)
future_pred_dt = dt.predict(future_data)
```

5.3 决策树实例

例5-4 使用线性回归和决策树构建核能源应用发展预测模型，以历史数据预测未来核能源发电量。

```
# -----  
# 5. 可视化结果  
# -----  
plt.figure(figsize=(12, 6))  
plt.plot(data['year'], data['nuclear_generation'], 'b-', label='历史数据')  
plt.plot(future_years, future_pred_lr, 'r--', label='线性回归预测')  
plt.plot(future_years, future_pred_dt, 'g--', label='决策树预测')  
plt.xlabel('年份')  
plt.ylabel('核能源发电量（太瓦时）')  
plt.title('核能源应用发展预测')  
plt.legend()  
plt.grid()  
plt.show()
```

The logo features a central green diamond containing the text. To the left of the diamond is a dark green chevron pointing right. To the right is a light green chevron pointing left. A horizontal line passes through the center of the composition. Several small diamonds in dark green, light green, and grey are positioned around the main elements.

Canllay 嘉为数字咨询

数字化人才培养
先行者