

第 3 章



模型评估

扫一扫



自测题

本章聚焦于机器学习模型的评估方法与性能度量。模型评估是衡量和提升模型泛化能力的重要手段,也是模型选择与优化的关键依据。随着数据科学的发展,构建高性能模型已不再是唯一目标,如何科学、全面地评估模型在未知数据上的表现,成为模型部署与应用的核心环节。

本章内容涵盖分类与回归任务中的常用评估指标,包括准确率、精确率、召回率、F1 分数、均方误差(MSE)、平均绝对误差(MAE)和决定系数(R^2)等,并进一步介绍模型性能评估的核心方法,如训练集-测试集划分、留出法、交叉验证(特别是 K 折交叉验证)及其实际应用。此外,本章还涉及模型调优中的关键策略,包括网格搜索、随机搜索和贝叶斯优化等超参数调优方法,帮助提升模型性能。

学习目标

- 理解模型评估在提升泛化能力、性能稳定性与可解释性中的核心作用。
- 掌握分类任务中准确率、精确率、召回率、F1 分数,以及回归任务中 MSE、MAE、 R^2 等常用评估指标及其适用场景。
- 熟练使用 K 折交叉验证评估模型表现,并正确解读评估结果。
- 掌握网格搜索、随机搜索和贝叶斯优化等超参数调优方法,能够在不同模型间进行科学比较与选择。

3.1 分类问题评估指标

石榴同学在学习机器学习时遇到了分类问题,她希望通过几个具体的评估指标来衡量她的模型性能。以下是石榴同学在学习过程中遇到的关键评估指标,包括这些指标的基本定义和结合案例的解释说明。

3.1.1 基本术语定义

在机器学习和统计学中,混淆矩阵(Confusion Matrix)是一种特别有用的工具,用于评估分类模型的性能。它通过展示模型预测结果与实际标签之间的关系,帮助理解模型在不同类别上的表现。以下是对混淆矩阵中 4 个关键指标的详细解释和说明。

(1) 真正例(True Positives, TP): 实际为正类,且被预测为正类的样本数。这是模型预

测正确的情况之一,表明模型成功地识别出了正类样本。在医学诊断中,真正例可以类比为正确诊断出患者患有某种疾病的情况。

(2) 真反例(True Negatives, TN): 实际为负类,且被预测为负类的样本数。这同样是模型预测正确的情况,表明模型成功地排除了负类样本。在安全系统中,真反例可以类比为正确识别出非威胁行为的情况。

(3) 假正例(False Positives, FP): 实际为负类,但被错误地预测为正类的样本数。这是模型预测错误的情况之一,可能会导致不必要的警报或资源浪费。在法律领域,假正例可能类比为错误地指控某人有罪的情况。

(4) 假反例(False Negatives, FN): 实际为正类,但被错误地预测为负类的样本数。这是另一种模型预测错误的情况,可能会导致严重的后果,如错过重要的诊断或干预机会。在金融欺诈检测中,假反例可能类比为未能识别出欺诈交易的情况。

通过上面的 4 个指标,就可以得到下面的混淆矩阵:

	预测为正类	预测为负类
真实为正类	TP(真正率)	FN(假反率)
真实为负类	FP(假正率)	TN(真反率)

同样通过这 4 个指标,可以计算出多种评估分类模型性能的指标,如准确率(Accuracy)、精确率(Precision)、召回率(Recall)和 F1 分数(F1 Score)等。这些指标有助于从不同角度理解模型的优缺点,从而进行相应的优化和调整。

3.1.2 准确率

准确率是所有正确预测的样本数占总样本数的比例,其定义如下:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (3.1)$$

石榴同学使用她的模型对 100 张图片进行了分类,其中 80 张图片被正确分类(40 张正类被正确识别为 TP,40 张负类被正确识别为 TN)。因此,石榴同学的模型准确率为

$$\text{Accuracy} = \frac{40 + 40}{100} = 0.8 \quad (\text{或 } 80\%) \quad (3.2)$$

3.1.3 精确率与召回率

精确率定义为在所有被预测为正类的样本中,真正为正类的比例,公式如下:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (3.3)$$

召回率定义为在所有真正为正类的样本中,被预测为正类的比例(也称为真正率 TPR),公式如下:

$$\text{Recall} = \text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3.4)$$

石榴同学的模型在预测为正类的 50 张图片中,有 45 张确实是正类图片(TP=45,FP=5)。同时,在所有真正的正类图片中(共 50 张),模型预测出了 45 张(FN=5)。因此,精确率和召回率分别为

$$\begin{aligned} \text{Precision} &= \frac{45}{50} = 0.9 \quad (\text{或 } 90\%) \\ \text{Recall} &= \frac{45}{50} = 0.9 \quad (\text{或 } 90\%) \end{aligned} \quad (3.5)$$

3.1.4 F1 分数

F1 分数是精确率和召回率的调和平均数,用于综合评估模型的性能,公式如下:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3.6)$$

结合精确率和召回率,石榴同学的模型 F1 分数为

$$\text{F1 Score} = 2 \times \frac{0.9 \times 0.9}{0.9 + 0.9} = 0.9 \quad (\text{或 } 90\%) \quad (3.7)$$

3.1.5 混淆矩阵

利用 3.1.2 节~3.1.4 节得到的数据,可以使用混淆矩阵描述分类模型的性能展示 TP、TN、FP、FN 的数量。例如,下面为一个案例矩阵:

	预测为正类	预测为负类
真实为正类	45	5
真实为负类	5	45

混淆矩阵直观地展示了石榴同学模型在各个分类上的表现。通过这些评估指标和案例说明,石榴同学能够更全面地了解她的模型性能,并据此进行进一步的优化和改进。

3.2 回归问题评估指标

在机器学习中,回归问题是指预测连续值输出的问题。为了评估回归模型的性能,通常使用一些特定的指标。本节将介绍 4 个常用的回归评估指标,并结合案例进行详细说明。

3.2.1 均方误差

均方误差 (Mean Squared Error, MSE) 是预测值与实际值之差的平方的平均值,公式如下:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (3.8)$$

其中, n 是样本数量, y_i 是第 i 个样本的实际值, $\hat{y}_i$ 是第 i 个样本的预测值。

假设石榴同学有一组房价数据,包含 5 个样本,预测值和实际值如表 3.1 所示。

表 3.1 房价预测案例数据集

样 本 编 号	实际值(y_i)	预测值($\hat{y}_i$)
1	200	220
2	300	280
3	150	160
4	400	380
5	250	245

根据式(3.8),MSE 的计算过程如下:

$$\begin{aligned} \text{MSE} &= \frac{1}{5} [(200 - 220)^2 + (300 - 280)^2 + (150 - 160)^2 + (400 - 380)^2 + (250 - 245)^2] \\ &= \frac{1}{5} [400 + 400 + 100 + 400 + 25] = \frac{1325}{5} = 265 \end{aligned} \quad (3.9)$$

3.2.2 均方根误差

均方根误差(Root Mean Squared Error, RMSE)是均方误差的平方根,它与 MSE 有相同的量纲,更容易解释,公式如下:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (3.10)$$

使用上述案例中的数据,可以计算 RMSE。

$$\begin{aligned} \text{RMSE} &= \sqrt{\frac{1}{5} [(200 - 220)^2 + (300 - 280)^2 + (150 - 160)^2 + (400 - 380)^2 + (250 - 245)^2]} \\ &= \sqrt{\frac{1325}{5}} = \sqrt{265} \approx 16.28 \end{aligned} \quad (3.11)$$

3.2.3 平均绝对误差

平均绝对误差(Mean Absolute Error, MAE)是预测值与实际值之差的绝对值的平均值,公式如下:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (3.12)$$

使用上述案例中的数据,可以计算 MAE。

$$\begin{aligned} \text{MAE} &= \frac{1}{5} [|200 - 220| + |300 - 280| + |150 - 160| + |400 - 380| + |250 - 245|] \\ &= \frac{1}{5} [20 + 20 + 10 + 20 + 5] = \frac{75}{5} = 15 \end{aligned} \quad (3.13)$$

3.2.4 R^2

R^2 (R-Squared)是衡量模型解释变异性的一个指标,其值介于 0~1。 R^2 越接近 1,表示模型的解释能力越强,公式如下:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (3.14)$$

其中, $\bar{y}$ 是所有实际值的平均值。使用 3.2.1 节案例中的数据,首先计算 $\bar{y}$:

$$\bar{y} = \frac{1}{5} (200 + 300 + 150 + 400 + 250) = \frac{1300}{5} = 260 \quad (3.15)$$

然后计算 R^2 :

$$R^2 = 1 - \frac{1325}{\sum_{i=1}^5 (y_i - 260)^2}$$

$$\begin{aligned} \text{其中,} \sum_{i=1}^5 (y_i - 260)^2 &= (200 - 260)^2 + (300 - 260)^2 + (150 - 260)^2 + (400 - 260)^2 + (250 - 260)^2 \\ &= 3600 + 1600 + 1600 + 19600 + 100 = 23500 \end{aligned}$$

$$\text{所以,} R^2 = 1 - \frac{1325}{23500} \approx 1 - 0.0564 = 0.9436 \quad (3.16)$$

通过上述计算可以看到,石榴同学的模型在该案例中具有较高的解释能力。

3.3 性能度量方法

在机器学习中,交叉评估是一种重要的模型验证方法,它通过将数据集分成多个子集,反复训练和验证模型,以评估模型的泛化能力。本节将介绍交叉评估的基本概念、常见方法及如何在实际案例中应用。

交叉评估(Cross-validation)是一种统计分析方法,用于验证分类器的性能。它的基本思想是将原始数据集分为训练集和验证集,使用训练集对分类器进行训练,然后利用验证集来测试训练得到的模型,以此来评价分类器的性能指标。

交叉评估的核心原则是确保模型在未见过的数据上也能有良好的表现,即评估模型的泛化能力。在实际应用中,交叉评估可以减少过拟合的风险,并从有限的数据中获取尽可能多的有效信息。

下面介绍交叉评估的具体方法。

3.3.1 留出法

留出法(Hold-out Method)是最简单的交叉验证方法。其将数据集随机分为两部分:训练集和测试集。通常,测试集占总数据集的 20%~30%,其余作为训练集。这种方法操作简单,但可能由于数据分布不均匀导致模型评估不准确。

3.3.2 K 折交叉验证

K 折交叉验证(K-Fold CV)是将数据集分为 K 个子集,每次选择其中一个子集作为测试集,其余 $K-1$ 个子集作为训练集。这个过程重复 K 次,每次选择不同的子集作为测试集,最终得到 K 个模型性能评估结果,取平均值作为最终评估结果。这种方法可以更全面地评估模型性能,但计算成本较高。

3.3.3 留一法交叉验证

留一法交叉验证(Leave-One-Out Cross-validation)是 K 折交叉验证的特例,其中 K 等于数据集中的样本数。每次只留下一个样本作为测试集,其余所有样本作为训练集。这种方法可以最大限度地利用数据,但计算成本极高,通常不推荐在大数据集上使用。

3.3.4 自助法

自助法交叉验证是一种基于重采样的方法。通过对原始数据集进行多次有放回的抽样,每次抽样得到一个新的数据集,然后使用这些数据集进行模型训练和评估。这种方法适用于小样本数据集,可以提供更稳健的模型性能估计。

3.3.5 交叉验证示例

本节通过一个具体的案例来说明如何应用交叉评估。假设石榴同学正在使用线性回归模型来预测房价,数据集包含 100 个样本。

为了更直观地理解交叉评估的过程,可以使用流程图来表示。图 3.1 所示为 5 折交叉验证的流程图。

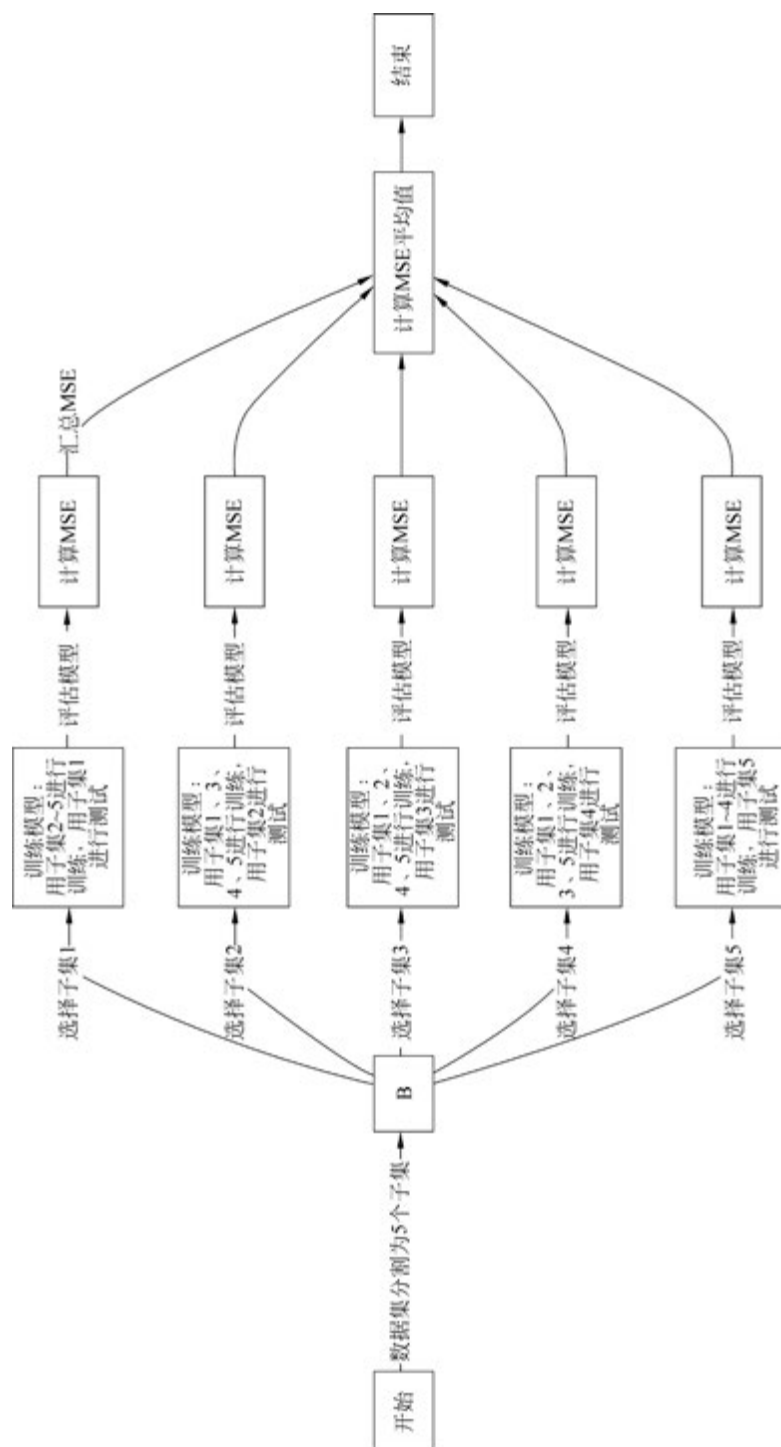


图 3.1 5 折交叉验证的流程图

石榴同学首先将数据集分为特征矩阵 $\mathbf{X}$ 和目标向量 $\mathbf{y}$, 其中 $\mathbf{X}$ 包含房屋的大小、位置等特征, $\mathbf{y}$ 是房价。然后选择使用 5 折交叉验证来评估模型。具体步骤如下:

- (1) 将 100 个样本随机分为 5 个子集, 每个子集包含 20 个样本。
- (2) 进行 5 次训练和验证。
 - ① 使用第 1~4 个子集(80 个样本)训练模型, 第 5 个子集(20 个样本)作为测试集。
 - ② 在测试集上计算模型的性能指标, 如均方误差(MSE)。
 - ③ 重复上述步骤, 每次选择不同的子集作为测试集。
- (3) 计算 5 次验证的 MSE 平均值, 作为模型的最终评估结果。

通过上述步骤, 石榴同学得到了一个稳定且可靠的模型性能评估结果。这个过程不仅评估了模型的准确性, 还提高了模型的泛化能力。

3.3.6 超参数调优方法

在机器学习的探索之旅中, 石榴同学作为一位对数据科学充满热情的学生, 正面临一个挑战: 如何优化她的模型的超参数。石榴同学知道, 超参数是在学习过程开始之前设置的参数, 对模型的性能有着决定性的影响。与模型参数不同, 超参数是通过训练数据学习得到的。合适的超参数设置可以显著提升模型的性能。超参数调优是模型训练过程中的重要环节, 它直接影响模型的泛化能力和训练效率。石榴同学探索了如下 3 种流行的超参数调优方法: 网格搜索(Grid Search)、随机搜索(Random Search)和贝叶斯优化(Bayesian Optimization)。

1. 网格搜索

网格搜索是一种穷举搜索方法, 它通过遍历超参数的所有可能组合来寻找最优超参数。这种方法的优点是全面, 但计算成本高, 特别是当超参数数量和候选值较多时。

2. 随机搜索

随机搜索是一种随机化的搜索方法, 它通过随机采样超参数的组合来寻找最优超参数。与网格搜索相比, 随机搜索的计算量较小, 能够在有限的时间内快速找到较好的超参数组合。

3. 贝叶斯优化

贝叶斯优化是一种基于概率模型的全局优化方法, 它通过构建目标函数的概率模型来寻找最优超参数。贝叶斯优化利用已有的观测数据训练一个高斯过程模型, 然后利用该模型预测未观测点的目标函数值和不确定性, 从而找到最优超参数。图 3.2 所示为贝叶斯优化流程图。

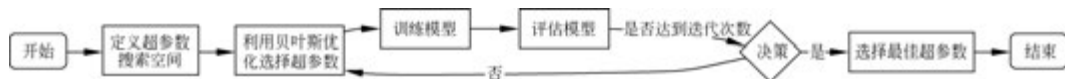


图 3.2 贝叶斯优化流程图

3.3.7 超参数调优示例

石榴同学有一组包含 100 个样本的数据集, 每个样本都有 10 个特征。她决定使用这 3 种方法来调优 SVM 模型。

1. 网格搜索

石榴同学定义了一个超参数网格, 包括不同的 C (正则化参数) 值和 Gamma (RBF 核的参数) 值, 如表 3.2 所示。

表 3.2 支持向量机优化参数

C 值	Gamma 值
1	0.1
1	1
10	0.1
10	1

对于每组超参数,石榴同学都训练了 SVM 模型,并使用交叉验证来评估模型的性能。她计算了每个模型的准确率,并记录了结果。

2. 随机搜索

石榴同学在相同的超参数空间内随机选取了 10 组超参数组合进行评估。这种方法比网格搜索更快,但也更具随机性。

3. 贝叶斯优化

石榴同学使用贝叶斯优化来寻找最优的超参数组合。她首先定义了超参数的搜索空间,然后利用贝叶斯优化库进行优化。贝叶斯优化的过程如下:

$$\text{Best Parameters} = \operatorname{argmax}_{\text{params}} (\text{Accuracy}(\text{params}))$$

(3.17)

在尝试了所有方法后,石榴同学发现贝叶斯优化在她的案例中表现最好,因为它在较少的迭代次数内找到了最优的超参数组合。

3.3.8 模型对比与选择

在机器学习的探索之旅中,石榴同学正面临另一个挑战:从多个模型中选择最佳模型,选择合适的模型对于解决特定的问题至关重要,该问题也被称为模型对比。模型对比是机器学习中的一个重要步骤,它涉及评估不同模型的性能,以便选择最适合特定问题需求的模型。模型选择直接影响解决方案的准确性、效率和可扩展性。常用的模型对比方法有如下 3 种:交叉验证、误差分析和模型融合。

1. 交叉验证

交叉验证是一种评估模型泛化能力的方法,它通过将数据集分成多个子集,反复训练和验证模型,以评估模型的性能。

2. 误差分析

误差分析是一种分析模型预测错误的类型和原因的方法,它有助于理解模型在哪些方面表现不佳,从而指导模型的改进。

3. 模型融合

模型融合是一种结合多个模型预测结果的方法,它可以通过提高模型的稳定性和准确性来提高整体性能。

为了清晰且专业地展示模型对比的意义,石榴同学选用了包含 100 个样本的数据集,其中,每个样本都具备 10 个特征。她计划采用决策树、随机森林和支持向量机 3 种方法来系统地比较不同模型的性能表现。石榴同学首先使用 5 折交叉验证来评估每个模型的性能。她计算了每个模型的准确率,并记录了结果。

表 3.3 所示为 3 个模型的性能对比。可以看出,随机森林模型具有最高的准确率,但训练时间较长。决策树模型的训练和测试时间最短,但准确率较低。

表 3.3 模型性能对比

模 型	准 确 率	训练时间(s)	测试时间(s)
决策树	85%	0.5	0.1
随机森林	90%	1.5	0.2
支持向量机	88%	2.0	0.3

其次,石榴同学进行了误差分析,以了解每个模型在哪些类型的样本上表现不佳。她发现决策树和支持向量机在处理边缘样本时容易出错,而随机森林在处理噪声数据时表现较好。

再次,基于上述分析,石榴同学决定尝试模型融合,将 3 个模型的预测结果结合起来,以提高整体性能。她使用简单的平均规则来融合预测结果,并发现融合后的模型在测试集上的准确率提高到 92%。

最后,综合考虑模型的性能、训练时间和应用场景,石榴同学最终选择了随机森林模型作为解决方案。虽然它的训练时间较长,但其高准确率和鲁棒性使其成为最佳选择。

扫一扫



视频讲解

3.4 实践练习

3.4.1 示例 1: 评估指标计算

在本次实践活动中,将通过动手实验深入理解并熟练掌握机器学习中重要的 4 类评估指标——准确率、AUC、F1 分数与混淆矩阵。借助这些指标,能够帮助人们全面、客观地衡量模型的性能与效果。为了顺利开展实验,需提前完成以下准备:首先,安装 Anaconda 发行版(或其他支持 Python 的 IDE);接着,启动 Jupyter Notebook;最后,如果没有安装 scikit-learn 库,可在终端或命令行中执行 `pip install scikit-learn` 命令完成 scikit-learn 库的安装。

1. 准确率(Accuracy)

准确率是最直观的评估指标,表示模型预测正确的样本数占总样本数的比例。

(1) 导入必要的库。

```
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.datasets import load_iris
```

(2) 加载数据集并划分训练集和测试集。

```
iris = load_iris()
X, y = iris.data, iris.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

(3) 训练模型并计算准确率。

```
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")
```

2. AUC(Area Under the Curve)

AUC 是评估二分类问题中模型性能的一个重要指标,表示模型对样本的正负类别区分能力的度量,可以使用以下程序进行计算:

```
from sklearn.metrics import roc_auc_score
y_pred_proba = model.predict_proba(X_test)[:, 1]
auc = roc_auc_score(y_test, y_pred_proba)
print(f"AUC: {auc:.2f}")
```

3. F1 分数(F1 Score)

F1 分数是精确率和召回率的调和平均数,用于评估模型在二分类问题中的平衡性能,具体计算方式如下:

```
from sklearn.metrics import f1_score
f1 = f1_score(y_test, y_pred)
print(f"F1 Score: {f1:.2f}")
```

4. 混淆矩阵(Confusion Matrix)

混淆矩阵是一个表格,用于描述分类模型的性能。它显示了实际类别和模型预测类别之间的关系,具体计算方法如下:

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:")
print(cm)
```

通过本实践活动,了解了准确率、AUC、F1 分数和混淆矩阵这 4 个重要的评估指标。这些指标有助于全面评价模型的性能,特别是在分类问题中。掌握这些评估指标的计算方法对于构建和优化机器学习模型至关重要。

3.4.2 示例 2: 模型调优

通过本实践活动,学习并掌握模型调优的流程,包括超参数选择、交叉验证等关键步骤,以及如何使用 scikit-learn 库中的相关工具进行模型调优。

为了顺利开展实验,需提前完成以下准备:首先,安装 Anaconda 发行版(或其他支持 Python 的 IDE);接着,启动 Jupyter Notebook;最后,如果没有安装 scikit-learn 库,可在终端或命令行中执行 `pip install scikit-learn` 命令完成 scikit-learn 库的安装。

1. 导入必要的库

导入进行模型调优所需的库,代码如下:

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
```

2. 加载数据集并划分训练集和测试集

这里使用 iris 数据集作为示例,并将其划分为训练集和测试集,代码如下:

```
iris = load_iris()
X, y = iris.data, iris.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

3. 定义模型和参数网格

定义一个随机森林分类器,并设置一个参数网格,用于后续的网格搜索,代码如下:

```
model = RandomForestClassifier(random_state=42)
param_grid = {
    'n_estimators': [10, 50, 100],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 5, 10]
}
```

4. 网格搜索

使用 GridSearchCV 进行网格搜索,找到最佳的参数组合,代码如下:

扫一扫



视频讲解

```
grid_search = GridSearchCV(estimator=model, param_grid=param_grid, cv=5, n_jobs=-1)
grid_search.fit(X_train, y_train)
```

5. 评估最佳模型

使用找到的最佳参数组合,评估模型在测试集上的性能,代码如下:

```
best_model = grid_search.best_estimator_
y_pred = best_model.predict(X_test)
print(classification_report(y_test, y_pred))
```

6. 分析结果

分析网格搜索的结果,包括最佳参数组合、最佳模型的分数等,代码如下:

```
print(f"Best parameters: {grid_search.best_params_}")
print(f"Best cross-validation score: {grid_search.best_score_}")
```