**对Scikit-learn库的学习和分析（一）**

* 1. **为什么选择Scikit-learn**

**选择Scikit-learn作为机器学习工具的原因有很多：**

**易用性：Scikit-learn提供了简洁且一致的API，使得即使是初学者也能快速上手。它的文档详尽，示例丰富，有助于用户理解和应用各种算法。**

**功能全面：Scikit-learn涵盖了广泛的机器学习算法和工具，包括数据预处理、特征工程、模型选择、评估等，满足了大多数机器学习任务的需求。**

**高效性：Scikit-learn的算法实现经过了优化，能够高效地处理大规模数据集，适合在实际项目中使用。**

**社区支持：Scikit-learn拥有一个活跃的开发者社区，不断有新的功能和改进被贡献出来，用户可以从中获得帮助和支持。**

**集成性：Scikit-learn与其他Python科学计算库（如NumPy、Pandas）无缝集成，方便用户在数据分析和机器学习之间进行切换。**

**3. 数据预处理**

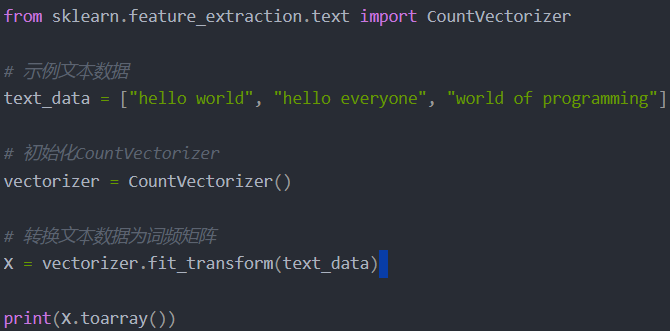
**3.1 数据清洗**

数据清洗是数据预处理的第一步，涉及处理缺失值、重复数据、异常值等。以下是一个简单的代码案例，展示如何使用Pandas进行数据清洗：



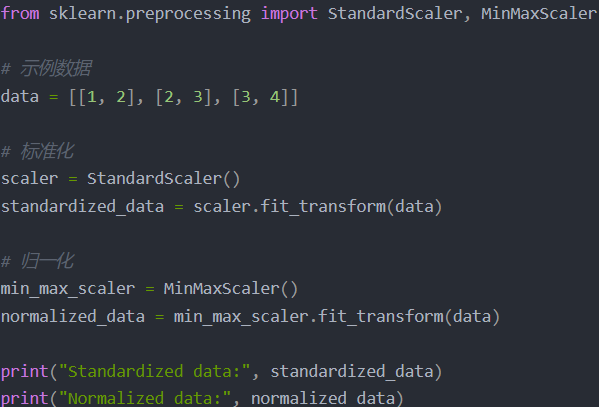
**3.2 特征提取与转换**

特征提取是将原始数据转换为更适合机器学习模型的特征表示。Scikit-learn提供了多种特征提取工具，如DictVectorizer用于处理字典数据，CountVectorizer用于文本数据的词频统计。以下是一个文本特征提取的示例：



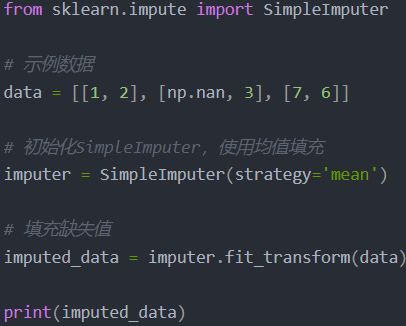
**3.3 标准化与归一化**

标准化和归一化是调整特征尺度的重要步骤，有助于提高某些算法的性能。以下是使用Scikit-learn进行标准化和归一化的示例：



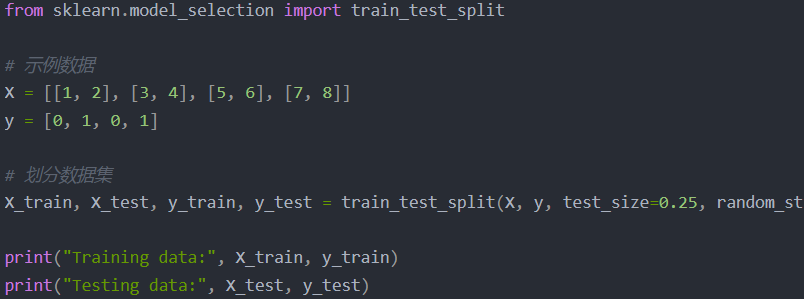
**3.4 缺失值处理**

处理缺失值是数据预处理中的常见任务。Scikit-learn提供了SimpleImputer来填充缺失值。以下是一个示例：



**3.5 数据集划分**

在机器学习中，通常将数据集划分为训练集和测试集，以评估模型的泛化能力。以下是使用Scikit-learn进行数据集划分的示例：

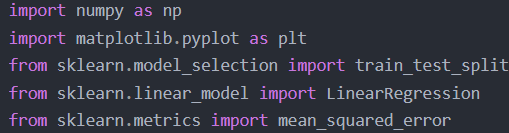


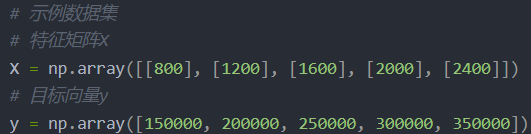
4. 监督学习

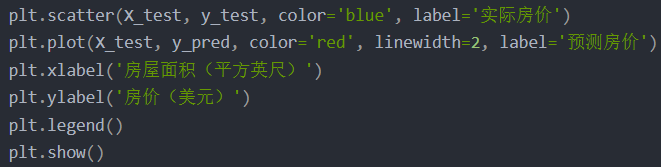
监督学习是机器学习的一个重要分支，它涉及使用标记数据来训练模型，以便对新数据进行预测。Scikit-learn提供了多种监督学习算法，以下是一些常见的监督学习方法及其代码案例。

4.1 线性模型

线性模型是监督学习中最基础的模型之一，它假设特征之间的关系可以用一条直线（对于二元分类）或超平面（对于多类分类）来表示。线性模型主要包括线性回归（用于连续目标变量）和逻辑回归（用于分类目标变量）。下面，我将通过一个简单的例子来讲解如何使用Scikit-learn进行线性模型的训练和预测。示例：使用线性回归进行房价预测假设我们有一个关于房屋的数据集，它包含了房屋的各种特征（比如面积、房间数、位置等），以及一个目标变量——房价（这是一个连续的数值型变量）首先，我们需要导入所需的库和模块：

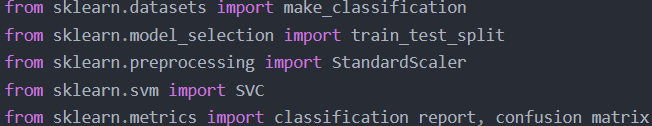




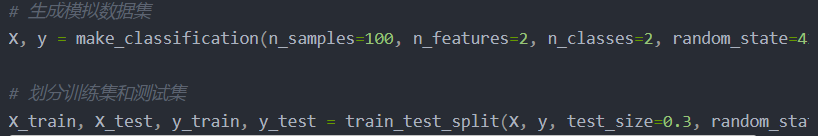


4.2 支持向量机

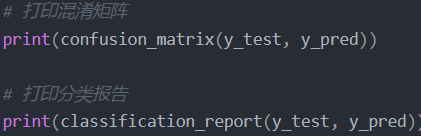
支持向量机（Support Vector Machine, SVM）是一种强大的监督学习算法，适用于分类和回归分析。在分类问题中，SVM旨在找到一个超平面，该超平面能够最大化不同类别之间的边际。Scikit-learn提供了多种SVM实现，包括线性SVM和核SVM。线性SVM适用于线性可分数据，而核SVM通过使用核技巧，可以处理非线性可分数据。下面，我将通过一个简单的例子来讲解如何使用Scikit-learn实现线性可分数据的支持向量机。示例：使用线性SVM进行二分类假设我们有一个关于红蓝花的数据集，其中花的大小是一个特征，花的颜色是另一个特征。我们的目标是根据这些特征将红花和蓝花区分开来。首先，我们需要导入所需的库和模块：



接下来，我们生成一个模拟的数据集：

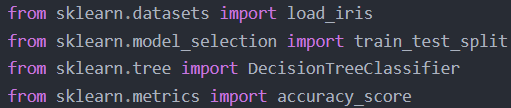


最后，我们评估模型的性能。我们可以使用混淆矩阵和分类报告来查看模型的准确率、召回率和F1分数：

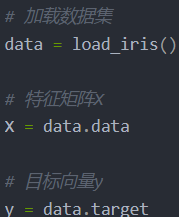


4.3 决策树与随机森林

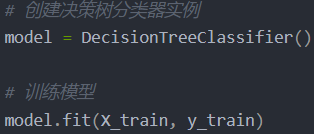
监督学习中的决策树（Decision Tree）和随机森林（Random Forest）是两种流行的分类和回归算法。决策树通过一系列的判断规则对数据进行分类，而随机森林是决策树的集成学习方法，通过构建多个决策树并进行投票来提高预测的准确性。下面，我将通过一个简单的例子来讲解如何使用Scikit-learn实现决策树和随机森林。示例：使用决策树进行分类假设我们有一个关于是否签署贷款的分类问题，特征包括年龄、收入、信用评分等。首先，我们需要导入所需的库和模块：



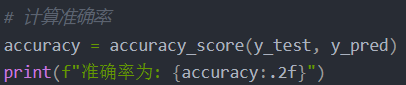
接下来，我们加载一个模拟的数据集（例如，Iris数据集）：



现在我们创建一个决策树分类器实例，并在训练集上进行训练：

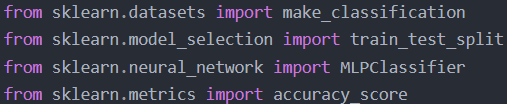


最后，我们评估模型的性能。我们可以使用准确率来评估分类的正确性：

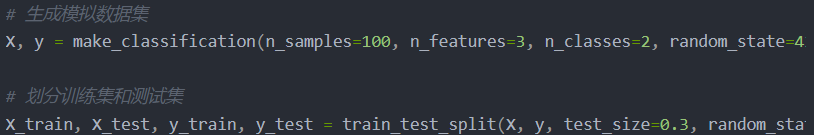


4.4 神经网络基础

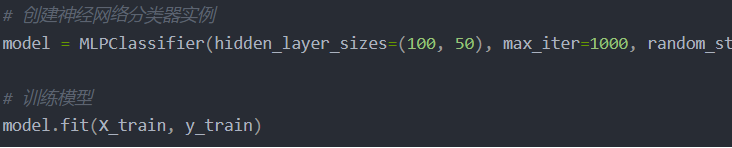
监督学习中的神经网络是模仿人脑神经元连接的方式进行信息处理的一种算法。Scikit-learn提供了一个模块sklearn.neural\_network，它基于Keras库实现了一些基本的神经网络功能。下面，我将通过一个简单的例子来讲解如何使用Scikit-learn实现神经网络。示例：使用神经网络进行分类假设我们有一个关于是否签署贷款的分类问题，特征包括年龄、收入、信用评分等。首先，我们需要导入所需的库和模块：



接下来，我们生成一个模拟的数据集：



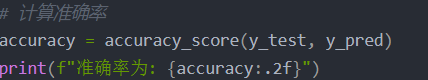
现在我们创建一个神经网络分类器实例，并在训练集上进行训练：



训练完成后，我们可以使用predict方法在测试集上进行预测：



最后，我们评估模型的性能。我们可以使用准确率来评估分类的正确性：



在这个例子中，我们看到了如何使用Scikit-learn中的神经网络进行分类问题的解决。在实际应用中，我们可能需要调整神经网络的结构（如隐藏层的大小、数量、激活函数等），以及训练参数（如学习率、批量大小、优化器等）来优化模型性能。此外，我们可能还需要进行数据预处理和特征工程来提高模型的预测能力。