


```

# 使用RFE进行特征选择
selector = RFE(estimator, n_features_to_select=10)
selector = selector.fit(X_train, y_train)

# 输出选择的特征
selected_features = selector.support_
print("Selected features:", np.where(selected_features)[0])

# 使用选择的特征训练模型
X_train_selected = selector.transform(X_train)
X_test_selected = selector.transform(X_test)

# 训练模型
estimator.fit(X_train_selected, y_train)

# 评估模型
score = estimator.score(X_test_selected, y_test)
print("Model accuracy:", score)

```

在这个例子中，我们首先生成了一个包含20个特征的分类数据集，其中5个特征是信息性的，另外5个是冗余的。然后，我们使用递归特征消除（RFE）方法，结合一个多层感知器分类器（MLPClassifier），来选择最重要的10个特征。最后，我们使用这些特征来训练模型，并评估其在测试集上的准确性。

3. 特征选择的基本方法

特征选择方法可以分为三类：过滤式（Filter）、包裹式（Wrapper）和嵌入式（Embedded）。

- 过滤式方法：基于特征与目标变量之间的统计关系来选择特征，如相关系数、卡方检验等。这种方法不考虑特征在模型中的作用，因此计算效率较高。
- 包裹式方法：将特征选择视为一个搜索问题，通过评估不同特征组合在模型上的表现来选择特征。这种方法通常需要较长的计算时间，但可以找到更优的特征组合。
- 嵌入式方法：在模型训练过程中同时进行特征选择，如正则化方法（Lasso、Ridge等）。这种方法结合了过滤式和包裹式方法的优点，既考虑了特征与目标变量的关系，又考虑了特征在模型中的作用。

3.1 示例：使用Lasso回归进行特征选择

Lasso回归是一种嵌入式特征选择方法，它通过添加L1正则化项来惩罚模型的系数，从而将一些特征的系数压缩至零，实现特征选择。以下是一个使用Python和scikit-learn库进行Lasso特征选择的示例：

```

import numpy as np
from sklearn.linear_model import Lasso
from sklearn.datasets import load_boston

```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/138027012024006111>