交叉验证（Cross Validation）

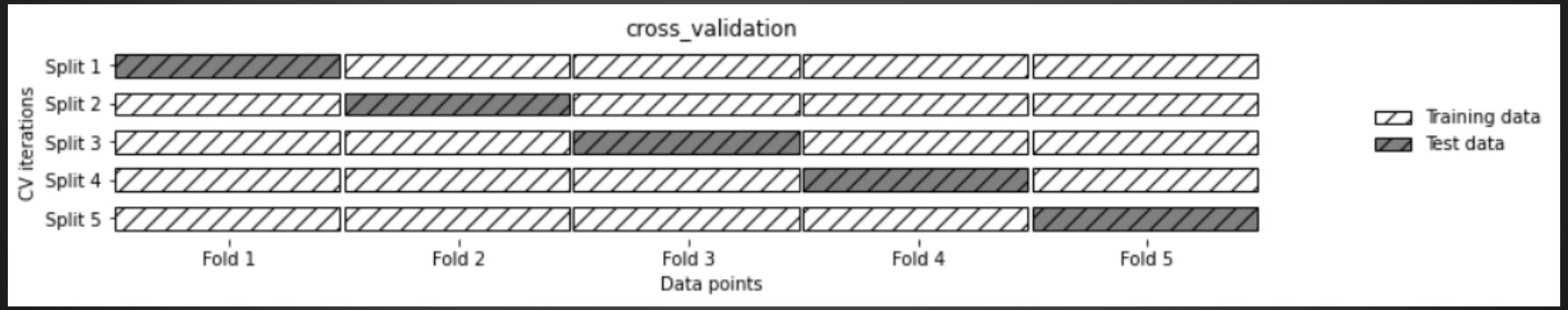
数据集的划分

- 数据集划分：将数据集划分为两部分，一部分用于模型训练，剩余一部分用于模型评估
- 单次划分：存在偶然性，例如训练集包括较多的容易分类的样本，测试集中的样本比较难以分类，这样会导致训练集的精度很高，而测试集的精度会很低！相反亦如此！

- 多次划分：将数据集多次划分，对于每一次划分都计算一个泛化精度，最后取一个平均精度，减小数据集划分带来的偶然性。缺点是增加计算量。
- 我们将多次划分取平均精度这种方式称为交叉验证（ Cross Validation ）。

交叉验证（Cross Validation）

- 交叉验证是一种评估泛化性能的统计学方法，在交叉验证中，数据被多次划分，并且需要训练多个模型
- 比单次划分训练集和测试集的方法更加稳定、全面。



简单交叉验证

sklearn中提供一个交叉验证的接口：cross_val_score

```
from sklearn.model_selection import cross_val_score
```

```
scores=cross_val_score(model,features.data,target.data,cv=5)
```

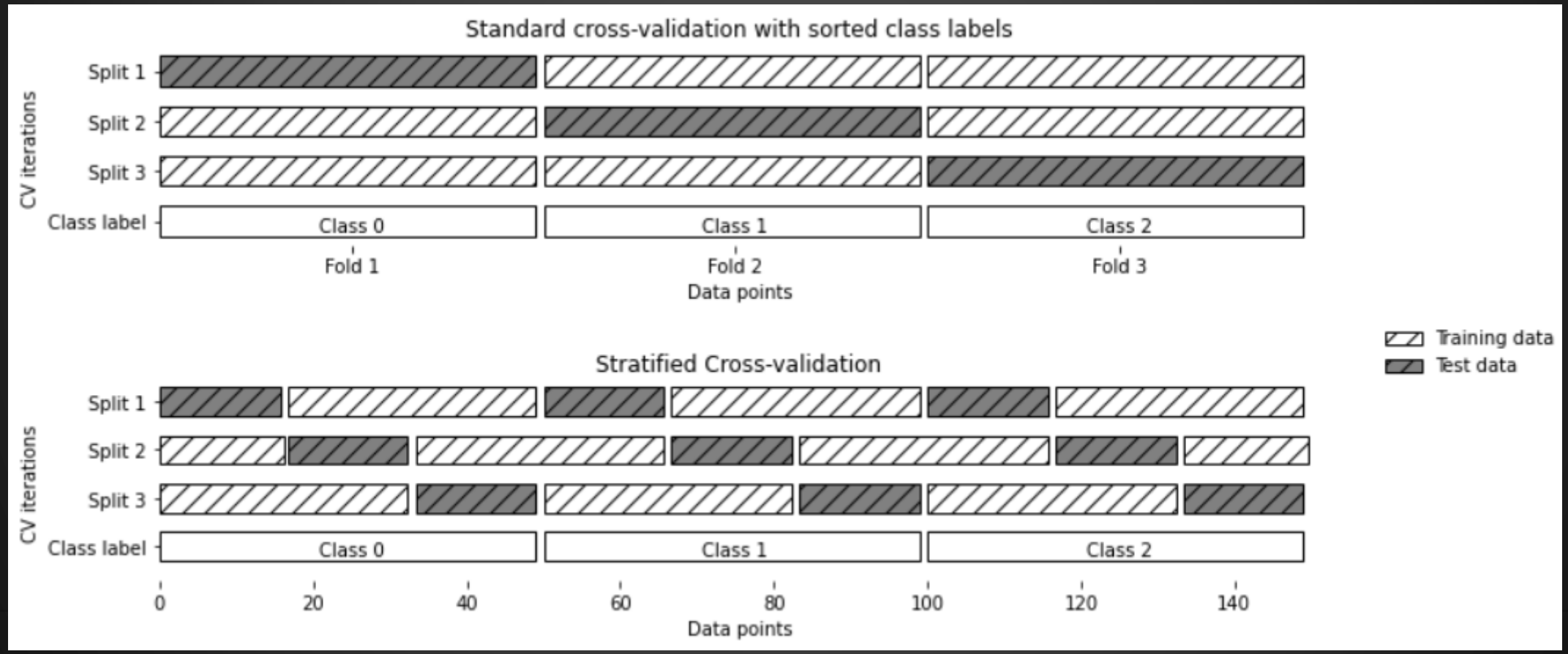
说明：参数cv指定进行几折交叉验证。

输出：

```
Cross-validation scores:[0.83268482 0.81656476 0.82768205 0.83759733 0.83759733]
```

简单交叉验证的问题

- 对于按照样本标签排序的数据，简单的交叉验证可能会失效，如下图所示。



交叉验证分离器：打乱数据

- sklearn中提供了一个交叉验证分离器，可以对数据划分进行更多的控制，例如打乱数据，使结果可重现等。
- 用法如下：

```
from sklearn.model_selection import KFold

kfolder=KFold(n_splits=5,shuffle=True,random_state=0)

scores=cross_val_score(LR_Model,X,y,cv=kfolder)
```

- 说明：

1. shuffle=True：划分前，将样本数据打乱
2. random_state=0：设置随机数种子，使结果可重现

交叉验证案例：员工离职预测

- 简单K折策略
- 交叉验证分离器