**异常值检测报告**

# 关于数据

数据自1965年至2005年，共计22943站次，2596104记录数,89.92MB。

数据格式：

1 1955032918 9.883 138.617 1 1 0

数据标识，时间YYYYMMDDHH，纬度，经度，仪器代码，温度是否存在，盐度是否存在

2 0.0 28.100 1 999.999 9

数据标识，深度，温度值，温度质控符，盐度值，盐度质控符

质控符（1正确、2可能正确、3可能错误、4错误、9缺测）【缺测值999.999】

# 数据预处理

1. 对于.OSF的数据进行转换，变成pandas能够处理的.csv
2. 标签标注（根据温度和盐度的错误数值，对每个训练样本进行标签标注）
3. 数据存在的问题
4. 数据样本的不平衡问题（正常样本个数22620, 异常样本个数282）
5. 数据的深度变长问题，，每一年的探测深度不同，导致序列变长
6. 数据缺失问题，大部分温度和盐度数据的缺失问题
7. 提取特征，采用某标准进行数据的统一化尺寸
8. 数据清洗，去掉空列
9. 提取特征，根据可能引起异常的原因，分析了以下特征：

|  |  |
| --- | --- |
| 特征分析 | 特征描述 |
| 时间 | 是一个特征， 因为早上和下午时间的不同，可能会影响水温，进而可能会导致出现异常 |
| 经度和纬度 | 是特征，因为不同的经纬度代表着地点不同， 可能会产生异常值 |
| 仪器代码 | 是一个特征，不同的仪器可能会产生异常 |
| 水的深度 | 是一个特征， 不同水深可能发生异常 |
| 温度和盐度 | 这两个是检测指标，异常不异常，也是特征 |

PS：前面三个，时间，经纬度和仪器代码，都是单独成列，可以直接提取作为特征列，但是后面的水深和温度盐度，由于深度不同，温度和盐度样本间又存在大量的缺失，所以需要进一步处理

1. 筛选出异常样本，以这些出问题的特征作为基准，进行特征统一（这样得到的特征是15740，特征太多，所以进一步筛选特征）
2. 在上面找到的特征基础上，筛选出引起异常的那个特征作为基准特征（这样最后得到的特征数量2929）
3. 构造统一化数据集

把所有的数据进行特征统一，构造成标准尺寸的数据集（22902行，2929列）

1. 基于标准化的数据集进行数据预处理
2. 缺失值的填充，由于缺失值太多，并且样本之间的差异，采用零填充
3. 数据归一化
4. 自编码压缩特征（由于特征还是太多，针对深度的那些特征，进行了自编码压缩至5维）
5. 最后得到的数据大小（22902, 9）

经过上述的处理，就可以用普通的机器学习模型进行分类预测。

# 训练集和测试集的划分

由于正负样本的数据严重不平衡，这里采用分层次抽样的方式，数据集分为训练集和测试集，比例是4:1

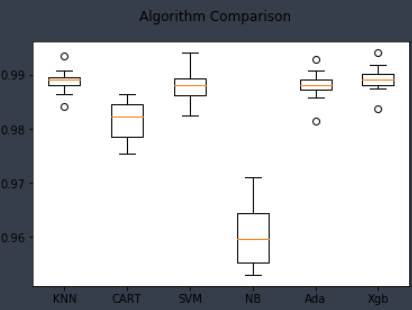
* 训练集中包括（18321个样本，其中18095个正常样本，226个异常样本）
* 测试集中包括（4581个样本，其中4525个正样本，56个异常样本）

# 模型创建与训练

模型的训练部分，采用K折交叉验证的方式，由于是分类的问题，尝试了六种分类模型，分别对比他们的训练效果的准确度的均值和方差，结果如下：

|  |  |  |
| --- | --- | --- |
| 模型 | 准确度均值 | 准确度方差 |
| KNN | 0.989 | 0.002 |
| CART | 0.982 | 0.004 |
| SVM | 0.988 | 0.003 |
| NaiveBayes | 0.961 | 0.006 |
| AdaBoost | 0.988 | 0.003 |
| XGBoost | 0.989 | 0.003 |

可视化如下：

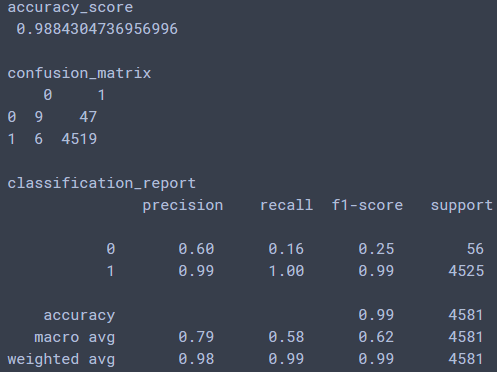


由训练结果，可以发现KNN和XGBoost的效果较好，所以后面用这两种模型进行预测

# 模型的测试

模型的测试部分，将KNN和XGBoost运用于测试集进行分类，结果如下：

1. **KNN**

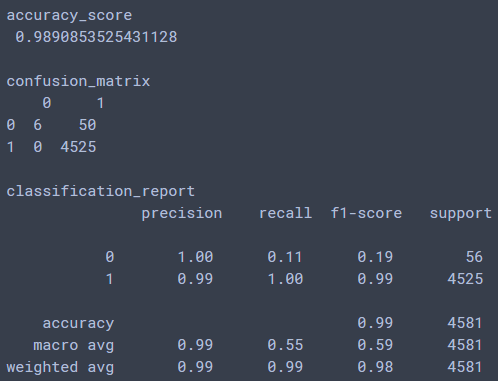


准确率达到：0.9884

对于正确样本：有4519个识别正确，6个分类错误

对于错误样本：有9个识别正确，47个分类错误

1. **XGBoost**



准确率达到：0.989

对于正确样本：全部分类正确

对于错误样本：有6个识别正确，50个分类错误

# 存在的问题与初步解决方法

对于KNN和XGBoost，普遍存在的问题就是对于异常的样本不敏感，没办法捕获到异常样本的特征，所以分析了一下可能的原因如下：

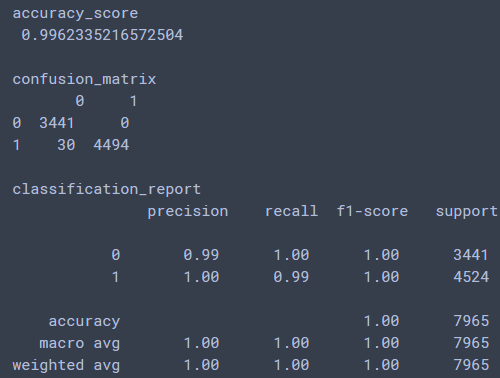
1. 由于样本的数量存在严重的不平衡问题，异常样本太少，所以模型学习的时候没办法学到太多异常样本的有用特征
2. 自编码压缩的时候可能压缩的太严重

**初步解决办法：**

* 对于异常样本数量太少的情况，比较好的解决办法是增加异常样本的数量
* 自编码压缩的这一个，需要反复试验，找到一个合适的压缩范围

# 初步的改进结果

对于样本数量不平衡的问题，进行了改进，增加了异常样本数量，采用异常样本分组随机复制的方式，把异常样本的个数增加到了17202个，这时候，正样本是22620个，相差不多，采用同样的方式进行训练测试，最后得到KNN的效果比较好，结果如下：



KNN准确率达到：0.996

对于正确样本：4494个识别正确，30个识别错误

对于错误样本：全部分类正确

增加异常样本的数量之后，KNN的效果显著提升，异常样本的识别率大大增加。

# 后期的改进思路

增加了异常样本的数量之后，KNN的效果比较好了，能够正确的分类出异常样本，但是感觉仍然可以进一步改进：

1. 异常样本的增加方式方面： 现在采用的单纯的复制原来的异常样本，或许可以用别的方式
2. 自编码压缩特征方面：目前采用的5维的压缩效果，特征可能会损失，并且训练次数太少，导致压缩的损失太大，后期可以寻找一个合理的压缩范围，更换更长的迭代次数
3. 由于数据存在严重的不平衡问题，是不是可以采用无监督的异常检测思想，计算样本出现的概率，通过概率的大小筛选异常样本。