程序报告

学号：2211312　　姓名：贾景顺

一、问题重述

垃圾短信在日常生活中的泛滥严重影响了人们正常的生活娱乐，因此在面对现如今的海量数据下，如何精确识别垃圾短信来保证更良好的用户体验至关重要。本实验基于基于 Python 的 Pandas、Numpy、Sklearn 等库进行相关特征处理，使用 Sklearn 框架训练分类器来完成对应垃圾短信的特征识别。实验过程中采用了多种方法，例如更换（扩充）停用词库，更改文本向量化方法，进行数据归一化，调整模型参数，来优化模型——以 f1_score 表示。

二、设计思想

代码整体首先导入对应的包以及训练用数据库（dataset），并利用 skleran 包中的 train_test_split 进行训练_验证集的划分，确定 random_state=42，训练集：验证集=9：1。

编写 read_stopwords 函数来完成对停用词库的读取，停用词库尝试了讲义中给出的四川大学停用词库，最终训练效果 f1_score=0.9193，随后上网搜索另一较大的停用词库（最全中文停用词表（可直接复制） 停用词库-CSDN 博客），最终训练效果 f1_score=0.9191,更换后效果不佳，故仍使用四川大学停用词库。

随后搭建 pipeline 便于后续模型的训练，在其中使用讲义默认的文本向量化方法 CountVectorizer 且不进行数据归一化，至此实验的代码已完善，能顺利通过接口测试和用例测试，后续进入模型调优部分。

尝试更换文本向量化方法：文本向量化方法更换为 TfidfVectorizer 后，f1 降至 0.9190，故不进行更换，仍使用 CountVectorizer。

使用 MaxAbsScaler 对其进行归一化，f1_score 提升至 0.9464.

分类器模型采用朴素贝叶斯模型（MultinomialNB），其中的平滑参数 alpha 根据设置数值不同，一定程度影响模型的训练效果。参数调整过程中尝试 alpha=0.8，1.0，1.5，2.0，其中当 alpha=1.5 时 f1_score=0.9475，在该组别中最高。

调整 ngram_range 参数，尝试（1，1），（1，2），（1，3），其中 f1_score 逐渐提高但模型训练时间也逐渐变长，故使用（1，3）作为参数传入。

在进行其他测试后，最终确定模型：CountVectorizer，MaxAbsScaler，MultinomialNB。相关参数 alpha=1.5，ngram_range=（1，3），优化后 f1_score=0.9573。

三、代码内容

```
#读取停用词代码
def read_stopwords(stopwords_path):
    """
    读取停用词库
    :param stopwords_path:  停用词库的路径
    :return:  停用词列表，如 ['嘿', '很', '乎', '会', '或']
    """
    stopwords = []
    # ----------  请完成读取停用词的代码  ----------
    """
```

```
    读取停用词库
    :param stopwords_path:  停用词库的路径
    :return:  停用词列表
    """
    with open(stopwords_path, 'r', encoding='utf-8') as f:
        stopwords = f.read()
    stopwords = stopwords.splitlines()
    return stopwords

    #---------------------------------------------

    return stopwords

# 读取停用词
stopwords = read_stopwords(stopwords_path)
# ----------------- 导入相关的库 -----------------
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.preprocessing import MaxAbsScaler
from sklearn.naive_bayes import BernoulliNB
from sklearn.naive_bayes import MultinomialNB
from sklearn.naive_bayes import ComplementNB

# pipline_list 用于传给 Pipline 作为参数
pipeline_list = [
    # -------------------------- 需要完成的代码 ------------------------------

    #    ============================    以 下 代 码 仅 供 参 考 ==============================
    ('cv',                                          CountVectorizer(token_pattern=r"(?u)\b\w+\b",
stop_words=stopwords,ngram_range=(1,3))),
    ('scaler', MaxAbsScaler()),
    ('classifier', MultinomialNB(
        alpha=1.5,              # 平滑参数（默认 1.0）
        fit_prior=True,        # 是否学习先验概率（默认 True）
        class_prior=None       # 手动指定先验概率（默认 None）
    ))
    #
================================================================================
==
    ]
#更换停用词库前 f1=0.9193433261955747
#更换停用词库后 f1=0.9191798784122064
```

#更换为 TfidfVectorizer 后 f1=0.9190590615909954
#使用 MaxAbsScaler()进行归一化后 f1=0.9464444727133953
#调整 alaph=1.5 f1=0.9475035297137724
#调整 alaph=2.0 f1=0.9461628057460852
#调整 alaph=0.8 f1=0.9450883526505796
#调整 ngram_range=(1,2) f1=0.9566198595787362
#调整 ngram_range=(1,3) f1=0.9573786956248823

四、实验结果

能够顺利通过接口测试和用例测试，预测结果正确率 7/10，模型未调优前正确率为 8/10，但 f1_score 相对较低，考虑到正确率表现较高的可能原因是测试点过少，故仍提交优化后的模型。

五、总结

在本次实验中，通过对 Pandas、Numpy、Sklearn 等库的使用，进一步了解了机器学习的实现方式。通过对参数的逐步调整与优化，我意识到对于不同要求的问题和不同类型的数据，对于模型选择适合的方法和参数也是非常重要的，对最终结果影响很大。在本次实验中，我没有重新分配随机数种子和测试集分割比例进行验证，这可能成为后续的优化方向。然而，总体上我已经对不同方法和参数进行了充分选择，也得到了还算不错的结果。