垃圾信息识别程序报告

学号：2111460　　姓名：张洋

一、问题重述

训练一个模型实现垃圾信息识别并进行分类，完成核心模型构建代码，调整模型参数，尽可能将模型调到最佳状态，尽可能多的识别垃圾信息。

二、设计思想

从以下几个方面去优化模型：

1. 数据预处理

因为普通短信的数量为 707464，垃圾短信的数量为 79146，普通短信的数量约为垃圾短信的十倍，所以对普通短信进行欠采样。

2. 文本向量化选择 TfidfVectorizer，适当调节里面的参数。

一些参数的详细解释：

（1）ngram_range: tuple(min_n, max_n)

要提取的 n-gram 的 n-values 的下限和上限范围，在 min_n <= n <= max_n 区间的 n 的全部值

（2）stop_words：string {'english'}, list, or None(default)

如果 english，用于英语内建的停用词列表

如果 list，该列表被假定为包含停用词，列表中的所有词都将从令牌中删除

如果 None，不使用停用词。max_df 可以被设置为范围[0.7, 1.0)的值，基于内部预料词频来自动检测和过滤停用词

（3）token_pattern：string

正则表达式显示了”token“的构成，仅当 analyzer == ‘word’时才被使用。两个或多个字母数字字符的正则表达式（标点符号完全被忽略，始终被视为一个标记分隔符）。

（4）max_df：　float in range [0.0, 1.0] or int, optional, 1.0 by default

当构建词汇表时，严格忽略高于给出阈值的文档频率的词条，语料指定的停用词。如果是浮点值，该参数代表文档的比例，整型绝对计数值，如果词汇表不为 None，此参数被忽略。

（5）min_df：float in range [0.0, 1.0] or int, optional, 1.0 by default

当构建词汇表时，严格忽略低于给出阈值的文档频率的词条，语料指定的停用词。如果是浮点值，该参数代表文档的比例，整型绝对计数值，如果词汇表不为 None，此参数被忽略。

经过调节参数的值，选择出最佳状态，对应的 ngram_range=（1，2），max_df=0.25。

3. 更换更好的停用词库

我对多个停用词库进行了尝试，包括中文停用词表（cn_stopwords）、哈工大停用词表（hit_stopwords）、百度停用词表（baidu_stopwords）、四川机器智能实验室停用词库（scu_stopwords）和以上四个词库合并去重合成的词库（combine_stopwords）

4. 对数据进行归一化，添加 MaxAbsScaler

5. 适当调节分类器的参数，提高模型的表现

选择 ComplementNB 分类器，通过多次调整参数，找到性能最好的一组参数为 alpha=0.8, fit_prior=True, class_prior=None, norm='l2'。

三、代码内容（添加部分标红）

```
import numpy as np
# 构建训练集和测试集
from sklearn.model_selection import train_test_split
X = np.array(sms.msg_new)
y = np.array(sms.label)

x1 = X[np.where(y==1)]
y1 = y[np.where(y==1)]
x0 = np.random.choice(X[np.where(y==0)],len(x1))
y0 = np.zeros(len(x1))
X = np.append(x0,x1)
y = np.append(y0,y1)

X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42, test_size=0.1)
print("总共的数据大小", X.shape)
print("训练集数据大小", X_train.shape)
print("测试集数据大小", X_test.shape)
```

```
import os
os.environ["HDF5_USE_FILE_LOCKING"] = "FALSE"

# ----------  停用词库路径，若有变化请修改  ----------
stopwords_path = r'combine_stopwords.txt'
# -------------------------------------------------

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  ------------
    with open(stopwords_path, 'r', encoding='utf-8') as f:
        stopwords = f.read()
    stopwords = stopwords.splitlines()
    #-------------------------------------------
```

```
    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.naive_bayes import BernoulliNB
from sklearn.naive_bayes import MultinomialNB
from sklearn.naive_bayes import ComplementNB
from sklearn.preprocessing import MaxAbsScaler
from sklearn.feature_extraction.text import TfidfVectorizer
# pipline_list 用于传给 Pipline 作为参数
pipeline_list = [
    # ------------------------- 需要完成的代码 ----------------------------------------------------
    ('cv', TfidfVectorizer(token_pattern=r"(?u)\b\w+\b", stop_words=stopwords, max_df=0.25,
ngram_range=(1,2))),
     #数据归一化
    ('MaxAbsScaler', MaxAbsScaler()),
     #分类器
    ('classifier',ComplementNB(alpha=0.8, fit_prior=True, class_prior=None, norm='l2'))
    # ---------------------------------------------------------------------------------------------
]
```

四、实验结果

```
在测试集上的混淆矩阵:
[[7260  613]
 [  31 7926]]
在测试集上的分类结果报告:
              precision    recall  f1-score   support

         0.0       1.00      0.92      0.96      7873
         1.0       0.93      1.00      0.96      7957

    accuracy                           0.96     15830
   macro avg       0.96      0.96      0.96     15830
weighted avg       0.96      0.96      0.96     15830

在测试集上的 f1-score :
0.9609602327837051
```

测试点	状态	时长	结果
测试读取停用词库函数结果	✓	10s	read_stopwords 函数返回的类型正确
测试模型预测结果	✓	13s	通过测试，训练的分类器具备检测恶意短信的能力，分类正确比例:10/10

测试样例全部正确分类。

五、总结

本次实验达到了预期目标，十条短信全部正确分类。

可能改进的方向：

1. 对于数据集中样本数量不平衡的情况通过减少多数类样本的数量来达到平衡数据集的目的。实验中对普通短信进行欠采样，即从普通短信中随机选择与垃圾短信数量相等的样本，然后将它们和垃圾短信一起构成平衡的数据集。这样可以避免模型偏向多数类，提高分类器对少数类的识别能力。但是在随机采样后可能代表性有所下降，可能导致导致信息的损失和模型性能下降。

2. 可以尝试使用其他的向量化器，如 Word2Vec、Doc2Vec 等。

3. 数据归一化：可以尝试使用不同的归一化方法和参数，比如使用 MinMaxScaler、RobustScaler 等。

4. 调节分类器参数：可以根据实际情况进行调节，如调节正则化参数、决策树深度、学习率等超参数，或者使用 GridSearchCV 进行网格搜索。

实验过程中遇到的困难：

在实现过程中调参是一个需要不断重复的过程，以期获得最好的参数。每次更改一个参数，控制其他参数的值不变，找到该参数的最优值。一个函数会有多个参数，通过在 csdn 上查找每个参数的含义并通过多次尝试才能找到最优值。

通过本次实验我认识到对于一个机器学习模型，调参是一个很重要的环节。在实践中，需要对不同的超参数进行组合，进行实验验证，选择最优的超参数组合。