程序报告

学号：2112514　　姓名：辛浩然

1　问题重述

垃圾短信识别的实质是对一条短信进行自动分类，判断其是否是垃圾短信。这个问题可以被视为一个二分类问题，即判断一条短信是正常短信还是垃圾短信。分类的过程可以通过机器学习的方法来实现，例如使用支持向量机、朴素贝叶斯等分类器进行训练和预测。在这个过程中，需要将短信文本进行预处理，提取出有用的特征，例如词频、文本长度等，然后将这些特征输入到分类器中进行训练和预测。通过不断地优化和更新模型，可以提高垃圾短信的识别率和准确度。

2　设计思想

垃圾短信识别的基本步骤包括：

1. 数据收集和预处理：对短信文本进行预处理，包括去除噪声、分词、去除停用词、词干提取、标点符号等。其中停用词是指一些常见的无实际意义的词语，如“的”、“了”、“是”等，这些词语在文本分类中通常没有太大的作用，需要进行去除。

2. 特征工程：将预处理后的文本转化为特征向量，一般采用文本向量化的方式。文本向量化是将文本转换成数值向量的过程，常用的方法有词袋模型和 TF-IDF 模型。词袋模型是将文本看作一个袋子，将所有的词语作为特征，每个词语的计数作为该词语在该文本中的特征值。TF-IDF 模型是一种根据词语在文本中出现的频率以及在整个语料库中的频率来计算每个词语的权重，然后将所有词语的权重作为该文本的特征向量。TF 表示词语在文本中出现的频率，IDF 表示逆文档频率。逆文档频率 (IDF) 是根据一个词在语料库中的出现次数来计算的，出现次数越多，IDF 越小，表示该词的重要性越低。

3. 划分训练集和测试集：将数据集划分成训练集和测试集。训练集用于训练模型，测试集用于评估模型的性能。

4. 模型搭建和训练：使用训练集训练模型，可以使用各种分类器模型，例如支持向量机、朴素贝叶斯等。朴素贝叶斯是一种基于贝叶斯公式的监督学习算法，其基本思想是：对于给定的待分类样本，通过已知类别的样本学习得到每个类别下各个特征的概率分布，并将待分类样本的特征向量代入各个类别的概率分布中计算其属于每个类别的概率，最终选择概率最大的类别作为预测结果。在模型搭建时，可以构建 PipleLine 将数据处理和数据分类结合在一起，这样输入原始的数据就可以得到分类的结果，方便直接对原始数据进行预测。

5. 模型评估和优化：使用测试集对训练好的模型进行评估，计算模型的准确率等指标。根据评估结果进行模型优化，例如调整模型参数、增加特征等。

3 代码内容

3.1 读取停用词

```
import os
os.environ["HDF5_USE_FILE_LOCKING"] = "FALSE"
# 停用词库路径
stopwords_path = r'scu_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 = []
    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)
```

读取一个停用词文件，并将文件内容转换成一个停用词列表，以供后续的文本处理使用。具体步骤如下：

首先，创建一个空列表 stopwords，以存储最终的停用词。

接着，使用 with open() 语句打开指定路径下的文件，文件模式为只读模式（'r'），文件编码为 UTF-8（'utf-8'）。使用 read() 读取文件内容，并将其存储在字符串 stopwords 中。

最后，使用 splitlines() 将字符串 stopwords 按行分割，得到一个包含每行文本的字符串列表，赋值给列表 stopwords。返回 stopwords。

3.2 实现 pipline_list

```
# 导入相关的库
from sklearn.pipeline import Pipeline
from sklearn.naive_bayes import MultinomialNB
from sklearn.naive_bayes import ComplementNB
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.preprocessing import MaxAbsScaler
from sklearn import preprocessing

# pipline_list用于传给Pipline作为参数
pipeline_list = [
```

```
    ('cv', TfidfVectorizer(token_pattern=r"(?u)\b\w+\b", decode_error =
    'ignore',stop_words=stopwords,ngram_range=(1,2))),
    ('MaxAbsScaler', preprocessing.MaxAbsScaler()),
    ('classifier', MultinomialNB())
]
```

pipeline 实现将文本数据经过预处理和特征工程后输入到分类器中进行预测。由三个步骤组成：文本特征提取（TfidfVectorizer）、数据归一化（MaxAbsScaler）和分类器（MultinomialNB）。

3.2.1 文本特征提取

文本特征提取使用 TfidfVectorizer 方法。TfidfVectorizer 是一种文本特征提取方法，它将原始文本转换为一组基于单词频率和文档频率的特征。TF-IDF 代表“词频-逆文档频率”，是一种常用于信息检索和文本挖掘的技术。在 TF-IDF 中，每个单词的重要性由两个因素决定：词频和逆文档频率。词频指单词在文档中出现的次数，逆文档频率指在整个语料库中出现单词的文档数量的倒数。TfidfVectorizer 的工作方式是先将文本分词，然后计算每个词语在文本中的词频和在整个语料库中的逆文档频率。最后，将这些特征向量标准化并组合成特征矩阵，以便进行机器学习模型的训练。

其中的参数含义如下：

token_pattern：正则表达式，用于提取文本中的单词或字符序列，默认值为 r”(?u) \ b \ w+ \ b”，表示提取至少包含一个字母或数字的单词。

decode_error：编码错误处理方式，默认为 ignore ，表示忽略编码错误。

stop_words：停用词表，是之前读取的停用词表。

ngram_range：n-gram 特征的范围。设为 (1,2)，表示提取 1-gram（单个词）和 2-gram（两个连续词）的词组合特征。

3.2.2 数据归一化

preprocessing 是 scikit-learn 库中的一个模块，用于数据预处理；MaxAbsScaler 是该模块中用于数据归一化的一个类。该步骤会对经过 TfidfVectorizer 转换后的特征进行缩放，将特征缩放到 [-1, 1] 之间的范围。

3.2.3 分类器

分类器选择使用 MultinomialNB，它是朴素贝叶斯分类器（Naive Bayes Classifier）的一种，常用于文本分类任务。它是一种基于概率的分类器，根据贝叶斯定理，将样本分类为具有最高后验概率的类别。

4 实验结果

在测试集中的表现结果：

```
在测试集上的混淆矩阵:
[[70104   710]
 [  175  7672]]
在测试集上的分类结果报告:
              precision    recall  f1-score   support

           0       1.00      0.99      0.99     70814
           1       0.92      0.98      0.95      7847

    accuracy                           0.99     78661
   macro avg       0.96      0.98      0.97     78661
weighted avg       0.99      0.99      0.99     78661

在测试集上的 f1-score :
0.9454679894016884
```

提交测试模型，通过测试，训练的分类器具备检测恶意短信的能力，分类正确比例为 9/10。

测试详情

测试点	状态	时长	结果
测试模型预测结果	✓	34s	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:9/10
测试读取停用词库函数结果	✓	43s	read_stopwords 函数返回的类型正确

5 总结

训练的分类器具备检测恶意短信的能力，分类正确比例为 9/10，基本达到了实现了垃圾短信识别的目的。

可能优化的方向：

1. 调节文本向量化的参数，达到更优表现；
2. 更换更好的停用词库；
3. 适当调节分类器的参数，提高模型的表现。
4. 更换其他分类器模型，如支持向量机等。