第三次实训程序报告

学号: 2310764 姓名: 王亦辉

1 问题重述

本实验旨在通过机器学习或深度学习方法，构建一个能够自动识别垃圾短信的分类模型。垃圾短信是指未经用户同意发送的广告或违法短信，随着手机使用的普及，垃圾短信的数量呈爆发式增长，已经对人们的生活和社会秩序造成了严重影响。实验提供了基础的数据读取、模型训练代码，要求在此基础上自行完成核心模型的设计与优化，使得模型在10秒内完成一次推理，并尽可能提高识别准确率。通过在海量短信数据中挖掘有用特征，有效区分正常短信与垃圾短信，是本次实验的核心目标。

2 设计思想

2.1 所采用的方法 ：

首先使用 `TfidfVectorizer` 替换原来的 `CountVectorizer` 对文本进行向量化，然后加上 `MaxAbsScaler` 对特征进行归一化，最后使用 `ComplementNB` 分类器替换原来的 `MultinomialNB`。

2.2 原来方法的局限性以及优化方向 ：

1. `MultinomialNB` 是一个多项式朴素贝叶斯分类器，但是我们的垃圾短信数据集是不平衡的，正常短信：垃圾短信= 8.938 ： 1，这可能会导致模型偏向认为一条短信是正常短信，即对垃圾短信的识别能力更弱。我们使用 `ComplementNB` 以适应这种不平衡数据集。
2. `CountVectorizer` 只计算词频，但是在垃圾短信的识别中，每个词的重要性是不一样的，如果把每个词认为是一样的，模型可能会过度关注那些在正常短信和垃圾短信中都出现但对判别垃圾短信没用的常用词，这回对模型产生干扰。我们可以使用 `TfidfVectorizer` 进行文本向量化，它不仅考虑词频，还考虑词的重要性（TF-IDF 值），在文本分类中表现更好。
3. 原方法没有进行特征归一化，可能会导致模型训练时某些特征的影响被放大，尤其是在稀疏矩阵的情况下。我们可以使用 `MaxAbsScaler` 进行特征归一化。`MaxAbsScaler` 将每个特征缩放到 `[-1, 1]` 范围，适合稀疏数据，能提高模型的训练效果，减少训练时的偏差。

2.3 `TfidfVectorizer` 中使用的超参数：

4. `ngram_range`：默认只考虑单个词，尝试 `(1, 2)`，同时考虑单个词和词对，捕获更多上下文信息。
5. `max_df` =0.9：过滤掉在 90%以上文档中出现的词，因为这些词在所有文档中都频繁出现，对文本分类帮助不大。
6. `min_df` =3：过滤掉在少于 3 个文档中出现的词，这些词可能没有代表性，噪声较大。

3 代码内容

（能体现解题思路的主要代码，有多个文件或模块可用多个隔开，必填）

```
# 读取停用词
def read_stopwords(stopwords_path):
    stopwords = []
    with open(stopwords_path, 'r', encoding='utf-8') as f:
        for line in f:
            word = line.strip()
            if word:
                stopwords.append(word)
    return stopwords

stopwords_path = r'scu_stopwords.txt'
stopwords = read_stopwords(stopwords_path)
```

```
# pipeline
pipeline = ImbPipeline([
    ('tfidf', TfidfVectorizer(
        token_pattern=r"(?u)\b\w+\b",
        stop_words=stopwords,
        ngram_range=(1, 2),
        max_df=0.9,
        min_df=3
    )),
    ('scaler', MaxAbsScaler()),
    ('classifier', ComplementNB())
])
```

4 实验结果

（实验结果，必填）

原本的 pipeline 得到的模型的实验结果如下：

经过优化之后的实验结果如下

```
测试集混淆矩阵：
[[69344  1470]
 [   38  7809]]
测试集分类报告：
              precision    recall  f1-score   support

         非垃圾       1.00      0.98      0.99     70814
          垃圾       0.84      1.00      0.91      7847

    accuracy                           0.98     78661
   macro avg       0.92      0.99      0.95     78661
weighted avg       0.98      0.98      0.98     78661

测试集 F1 分数（标签 1）：
0.9119467476351746
测试集 ROC-AUC 分数：
0.999327797997521
```

5 总结

归一化可以提升性能，是由于使得各个数据在模型训练时具有相同的权重，从而防止某些特征由于其较大的数值范围对模型造成过大的影响。

尝试过采样来平衡数据集但是起到了反效果，可能是因为过采样导致了过拟合，特别是在数据本身已经很复杂的情况下。且过多的重复样本可能会降低模型的泛化能力，导致反效果。

采用了超参数搜索，结果找到的参数还不如默认参数，这是由于训练数据较大，而我为了减少训练时间，对训练数据进行采样，导致找到超参数可能并非在全部数据上的最优结果。