程序报告

学号：22375080　　姓名：杨佳宇轩

一、问题重述

（简单描述对问题的理解，从问题中抓住主干，必填）

==========

实验旨在通过机器学习的方法实现中文垃圾短信的自动分类，并通过设置不同的模型以及优化方法来探索最优方法。具体而言，给定标注数据集，构建文本分类模型，完成线性回归、正则化优化线性模型、逻辑回归、探索 sklearn 库。最终目标是在私有测试机上达到最优的分类效果

二、设计思想

（所采用的方法，有无对方法加以改进，该方法有哪些优化方向（参数调整，框架调整，或者指出方法的局限性和常见问题），伪代码，理论结果验证等… **非必填**）

==========

采用的方法

1. 使用卡方检验选择了 1000 个最具有区分度的特征，降维向量矩阵
2. 在实验中使用正则化优化线性模型，控制模型复杂度

优化方向

1. 可以尝试使用 TF-IDF 代替词频统计
2. 实验不同的正则化方法

三、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）

==========

计算正则损失：

```
reg_loss = 0.0
for param in model.parameters():
    # 偏置项排除
    if 'weight' in param.name:
    if reg_type == 'l1':
        reg_loss += lambda_ * paddle.sum(paddle.abs(param))
    else:
        reg_loss += lambda_ * paddle.sum(paddle.square(param))
```

==========

卡方检验，降维向量矩阵

```
from sklearn.feature_selection import SelectKBest, chi2
selector = SelectKBest(chi2, k=1000)
X_train_selected = selector.fit_transform(X_train_dtm, y_train)
X_test_selected = selector.transform(X_test_dtm)
```

```
X_private_selected = selector.transform(X_private_dtm)
=====================
数据集向量化
from sklearn.feature_extraction.text import CountVectorizer
#  设置匹配的正则表达式和停用词
vect = CountVectorizer(token_pattern=r"(?u)\b\w+\b", stop_words=stopwords)
# vect = TfidfVectorizer(token_pattern=r"(?u)\b\w+\b", stop_words=stopwords)
X_train_dtm = vect.fit_transform(X_train)
X_test_dtm = vect.transform(X_test)
X_private_dtm = vect.transform(X_private)
```

四、实验结果

（实验结果，必填）

==========

300epoch　F1-score
线性回归 0.8003491271820449
正则优化 0.8428428927680797
逻辑回归 0.8626943005181348
支持向量机 0.8857142857142858
决策树 0.7704280155642023
随机森林 0.7967257844474762
梯度提升树 0.7878787878787878

五、总结

（自评分析（是否达到目标预期，可能改进的方向，实现过程中遇到的困难，从哪些方面可以提升性能，模型的超参数和框架搜索是否合理等），必填）

==========

成果分析
1. Sklearn-支持向量机的表现最佳，F1-score 为 0.8857142857142858
2. 正则化优化没有达到预期效果，可能和 λ 值有关

优化方向
1. 可以尝试使用 BERT 等预训练模型
2. 加入自注意力机制，提升对关联垃圾信息的提取
3. 增加标注数据量
4. 调整数据类别占比不平衡的问题

收获
1. 特征选择对于高维文本数据至关重要，需要寻找更优秀的文本转向量方法
2. 成熟的机器学习库往往在任务上相较于手工更加高效