**程序报告**

学号：2110508 姓名：杨冰雪

1. **问题重述**

（简单描述对问题的理解，从问题中抓住主干，必填）

====================================================================

垃圾短信分类实验是要求实现一个模型对一段中文文本进行预测，预测其是否是一个垃圾短信。模型的大致实现流程如下：

1. 读取数据集。在程序中导入一个带有标签的短信数据集，其中含垃圾短信和正常短信。
2. 读取停用词。停用词是指在文本数据处理时会过滤掉的词汇。
3. 文本向量化。需将文本数据转化为向量输入到机器学习算法中。
4. 对数据集进行划分，在训练集上训练并在测试集上测试。

构建pineline，将数据处理与数据分类组合，进行数据预测

1. **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… **思考题，非必填**）

====================================================================

总体实现流程如下：

1. 将带有标签的数据集导入，利用sklearn库中的train\_test\_split对数据集进行划分。划分为训练集和测试集，比例为0.9和0.1。每个数据X都有对应的类别标签y，0代表正常短信，1代表垃圾短信。
2. 读取停用词用于后续的数据处理。使用了最全的中文停用词库替代了原有默认的“四川大学停用词表”。
3. 数据预处理分为两步：

使用CountVectorizer将文本进行向量化，参数设置为：

CountVectorizer(token\_pattern=r"(?u)\b\w+\b",stop\_words=stopwords)

①token\_pattern=r"(?u)\b\w+\b"表示提取特征词时可以提取单个词，在默认情况下CountVectorizer不能提取单个词；②stop\_words=stopwords表示使用显现导入的停用词；

使用MaxAbsScaler对数据进行归一化

1. 构造分类器

采用朴素贝叶斯MultinomialNB作为分类器

1. **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）

====================================================================

关键点1：构建预测集和训练集

X = np.array(sms.msg\_new)

y = np.array(sms.label)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, random\_state=42, test\_size=0.2)

将数据集按8:2拆分成训练集和测试集。

====================================================================

关键点2：读入停用词

def read\_stopwords(stopwords\_path):

stopwords = []

with open(stopwords\_path, 'r', encoding='utf-8') as f:

for line in f:

stopwords.append(line.strip())

return stopwords

stopwords = read\_stopwords(stopwords\_path)

进停用词读入stopwords中。

====================================================================

关键点3：构造Pineline

pipeline\_list = [

('cv', CountVectorizer(token\_pattern=r"(?u)\b\w+\b", stop\_words=stopwords)),

('norm',MaxAbsScaler()),

('classifier', MultinomialNB())

]

====================================================================

关键点4：pred

def predict(message):

label = pipeline.predict([message])[0] #分类标签

proba = list(pipeline.predict\_proba([message])[0]) #分为每个类的概率

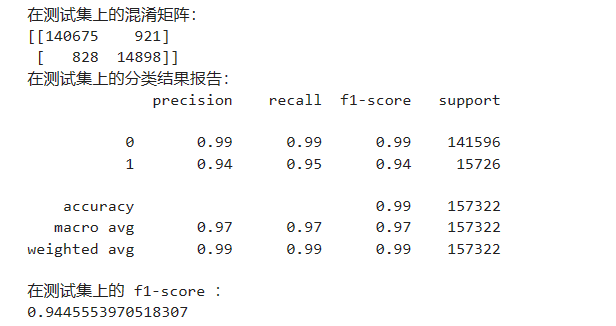
return label, proba

返回分类预测结果。

1. **实验结果**

（实验结果，必填）

====================================================================





1. **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），**思考题，非必填**）

====================================================================

1. 实验结果分析：在本地测试时对于数据集的测试集的预测结果正确率可以达到94%左右，但是提交平台测试分类正确的比例为7/10，说明模型的泛化能力可能还不够强。
2. 可尝试进行改进的方向：

1.适当调节分类器的参数或者使用准确度更高的分类器

2.做更多的数据预处理。如清洗、去重、平衡等处理，以减少噪声和不必要的干扰。