**程序报告**

学号：2213410 姓名：徐俊智

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）

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

构建一个垃圾短信识别系统，该系统能够自动区分正常短信和垃圾短信。核心任务是从文本数据中提取有效特征，并利用这些特征训练一个分类模型，以准确预测未知短信的类别。

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**）

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

方法：

1.数据预处理：首先对原始短信文本进行清洗，包括去除标点符号、数字、无意义的停用词等，以减少噪声并提高后续处理的效率。

2.中文分词：使用jieba分词工具对短信内容进行分词处理，以便更好地捕捉文本的语义信息。

3.特征提取：通过CountVectorizer或TfidfVectorizer将分词后的文本转换为数值特征向量，为机器学习模型提供输入。

4.模型选择与训练：选择了朴素贝叶斯分类器作为基础模型，并利用Pipeline将数据预处理和模型训练封装在一起，以便批量处理和模型训练。

5.模型评估与优化：在测试集上评估模型的性能，并通过调整模型参数、改进特征提取方法等方式来优化模型。

优化方向：

1.参数调整：通过调整CountVectorizer和TfidfVectorizer的参数（如ngram\_range、max\_df、min\_df等）来优化特征提取过程。

1. 框架调整：尝试不同的文本向量化方法和分类器，如支持向量机（SVM）、逻辑回归等。
2. 局限性：朴素贝叶斯分类器假设特征之间相互独立，这在实际文本数据中可能不成立，可能会影响模型的准确性。
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）

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

# 导入相关的包

import warnings

warnings.filterwarnings('ignore')

import os

os.environ["HDF5\_USE\_FILE\_LOCKING"] = "FALSE"

import pandas as pd

import numpy as np

# 数据集的路径

data\_path = "./datasets/5f9ae242cae5285cd734b91e-momodel/sms\_pub.csv"

# 读取数据

sms = pd.read\_csv(data\_path, encoding='utf-8')

# ---------------------------------------------------

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

"""

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

stopwords = f.read()

stopwords = stopwords.splitlines()

return stopwords

# 停用词库路径

stopwords\_path = r'scu\_stopwords.txt'

# 读取停用词

stopwords = read\_stopwords(stopwords\_path)

# 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)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, random\_state=42, test\_size=0.1)

# ----------------- 导入相关的库 -----------------

from sklearn.pipeline import Pipeline

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.naive\_bayes import MultinomialNB

from sklearn.naive\_bayes import BernoulliNB

from sklearn.preprocessing import MaxAbsScaler

from sklearn.preprocessing import StandardScaler

from sklearn.preprocessing import MinMaxScaler

from sklearn.preprocessing import Normalizer

from sklearn.linear\_model import LogisticRegression

from sklearn.naive\_bayes import ComplementNB

from sklearn import preprocessing

# pipline\_list用于传给Pipline作为参数

pipeline\_list = [

# --------------------------- 需要完成的代码 ------------------------------

# ========================== 以下代码仅供参考 =============================

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

('classifier', MultinomialNB()),

# ========================================================================

# ------------------------------------------------------------------------

]

# 搭建 pipeline

pipeline = Pipeline(pipeline\_list)

# 训练 pipeline

pipeline.fit(X\_train, y\_train)

# 对测试集的数据集进行预测

y\_pred = pipeline.predict(X\_test)

# 在测试集上进行评估

from sklearn import metrics

print("在测试集上的混淆矩阵：")

print(metrics.confusion\_matrix(y\_test, y\_pred))

print("在测试集上的分类结果报告：")

print(metrics.classification\_report(y\_test, y\_pred))

print("在测试集上的 f1-score ：")

print(metrics.f1\_score(y\_test, y\_pred))

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

y = np.array(sms.label)

# 在所有的样本上训练一次，充分利用已有的数据，提高模型的泛化能力

pipeline.fit(X, y)

# 保存训练的模型，请将模型保存在 results 目录下

from sklearn.externals import joblib

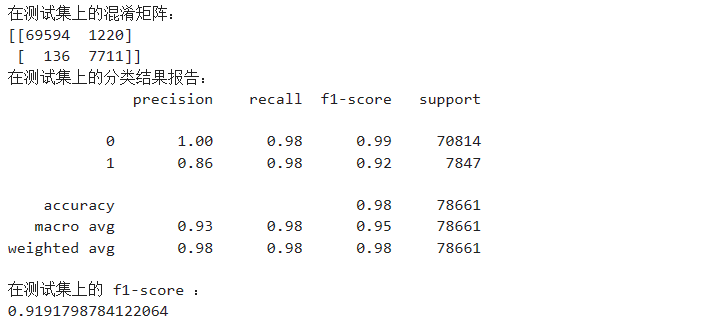
pipeline\_path = 'results/pipeline.model'

joblib.dump(pipeline, pipeline\_path)

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**）

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

在本次实验中，我成功的构建了一个垃圾短信识别系统，该系统在测试集上达到了较高的准确率和F1分数，基本满足了预期目标。

可能改进的方向：进行更细致的超参数调整和模型框架搜索，以找到最优的模型配置。

实现过程中遇到的困难：模型过拟合、特征提取不够充分等。

提升性能：通过数据增强技术扩充训练集，提高模型的泛化能力。