**程序报告**

学号：2311061 姓名：马淏怡

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）

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

1.1 实验背景

垃圾短信是指未经过用户同意向用户发送不愿接收的商业广告或者不

符合法律规范的短信。随着手机的普及，垃圾短信在日常生活日益泛滥，已经严重的影响到了人们的正常生活娱乐，乃至社会的稳定。 据 360 公司 2020 年第一季度有关手机安全的报告提到，360 手机卫士在第一季度共拦截各类垃圾短信约 34.4 亿条，平均每日拦截垃圾短信约 3784.7 万条。 大数据时代的到来使得大量个人信息数据得以沉淀和积累，但是庞大的数据量缺乏有效的整理规范；在面对量级如此巨大的短信数据时，为了保证更良好的用户体验，如何从数据中挖掘出更多有意义的信息为人们免受垃圾短信骚扰成为当前亟待解决的问题。

1.2 实验要求

1. 任务提供包括数据读取、基础模型、模型训练等基本代码

2. 参赛选手需完成核心模型构建代码，并尽可能将模型调到最佳状态

3. 模型单次推理时间不超过 10 秒

1.3 实验环境

可以使用基于python 的Numpy、Pandas、SKlearn 等库进行相关特征处理，使用SKlearn

框架训练分类器，也可编写深度学习模型，使用过程中请注意python 包（库）的版

本。

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**）

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

1. 初始设置停用词
2. 、导入停用词词库
3. 、读入停用词，返回停用词列表
4. Pipeline构建

（1）、CountVectorizer: 文本特征提取

token\_pattern=r"(?u)\b\w+\b": 匹配所有Unicode单词

stop\_words=stopwords: 使用之前加载的停用词表

（2）、MultinomialNB: 多项式朴素贝叶斯分类器

1. 数据准备
2. 、从CSV文件加载数据集
3. 、提取短信文本(msg\_new)作为特征X
4. 、提取标签(label)作为目标y
5. 、将数据划分为训练集(90%)和测试集(10%)
6. 模型训练与评估
7. 、在训练集上训练模型
8. 、在测试集上进行预测
9. 、输出模型在测试集上性能指标：  
    混淆矩阵：显示真正例、假正例、真负例、假负例的数量

分类报告：包括精确度、召回率、F1分数等

F1分数：精确度和召回率的调和平均

1. 保存最终模型
2. 、使用全部数据重新训练模型
3. 、将训练好的模型保存到文件中
4. **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）

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

1. 模型代码

import os

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

# ---------- 停用词库路径，若有变化请修改 -------------

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 = []

# ----------- 请完成读取停用词的代码 ------------

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

stopwords = f.read()

stopwords = stopwords.splitlines()

return stopwords

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

return stopwords

# 读取停用词

stopwords = read\_stopwords(stopwords\_path)

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

from sklearn.pipeline import Pipeline

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.naive\_bayes import BernoulliNB

from sklearn.naive\_bayes import MultinomialNB

from sklearn.naive\_bayes import ComplementNB

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

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

('classifier', MultinomialNB())

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

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

]

# 搭建 pipeline

pipeline = Pipeline(pipeline\_list)

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')

# 显示前 5 条数据

sms.head()

# 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)

# 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))

# 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. 主函数

import os

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

# ---------- 停用词库路径，若有变化请修改 -------------

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 = []

# ----------- 请完成读取停用词的代码 ------------

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

stopwords = f.read()

stopwords = stopwords.splitlines()

return stopwords

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

return stopwords

# 读取停用词

stopwords = read\_stopwords(stopwords\_path)

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

from sklearn.pipeline import Pipeline

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.naive\_bayes import BernoulliNB

from sklearn.naive\_bayes import MultinomialNB

from sklearn.naive\_bayes import ComplementNB

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

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

('classifier', MultinomialNB())

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

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

]

# 加载训练好的模型

from sklearn.externals import joblib

# ------- pipeline 保存的路径，若有变化请修改 --------

pipeline\_path = 'results/pipeline.model'

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

pipeline = joblib.load(pipeline\_path)

def predict(message):

"""

预测短信短信的类别和每个类别的概率

param: message: 经过jieba分词的短信，如"医生 拿 着 我 的 报告单 说 ： 幸亏 你 来 的 早 啊"

return: label: 整数类型，短信的类别，0 代表正常，1 代表恶意

proba: 列表类型，短信属于每个类别的概率，如[0.3, 0.7]，认为短信属于 0 的概率为 0.3，属于 1 的概率为 0.7

"""

label = pipeline.predict([message])[0]

proba = list(pipeline.predict\_proba([message])[0])

return label, proba

if \_\_name\_\_ == '\_\_main\_\_':

import pandas as pd

import numpy as np

data\_eval = pd.read\_csv('sms\_pub.csv', encoding='utf8')

y\_eval = np.array(data\_eval['label'])

X\_eval = np.array(data\_eval['msg\_new'])

total = y\_eval.shape[0]

count = 0

for x, y in zip(X\_eval, y\_eval):

y\_pred, \_ = predict(x)

if y\_pred == y:

count += 1

print('{} / {}'.format(count, total))

# 测试用例

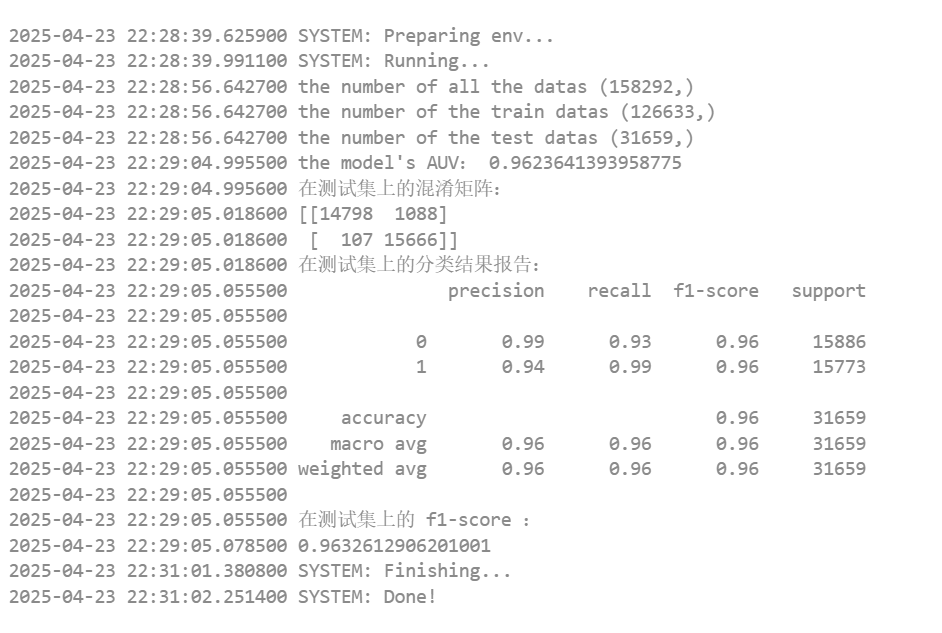
label, proba = predict('医生 拿 着 我 的 报告单 说 ： 幸亏 你 来 的 早 啊')

print(label, proba)

1. **实验结果**

（实验结果，必填）

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



生成模型时的日志



系统测试结果

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、本实验通过采用朴素贝叶斯分类器和 TF-IDF 特征提取方法，实现了对垃圾短信的识别。根据实验结果，该方法在测试集上的 F1-score 和 AUC 上达到了预期目标。

2、可以通过调整参数来实现模型的优化。比如调整训练集/测试集比例：增大训练集比例，模型能利用更多数据训练可能提高泛化能力，但评估结果可能方差较大；减小训练集比例，测试集更大评估更可靠，但训练数据变少模型可能欠拟合使正确率下降。再比如调整分类器参数，MultinomialNB 可调整 alpha，目前alpha值为0.5，可尝试0.1~1.0

3、可以改进文本预处理，调整 CountVectorizer。

3、也可以选择其他的机器学习、深度学习算法模型来提升与优化模型。