**程序报告**

学号：2113662 姓名：张丛

1. **问题重述**

垃圾短信 (Spam Messages，SM) 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。

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. **设计思想**

设计思想主要是基于机器学习和自然语言处理技术。

1. 数据收集：收集大量的垃圾短信数据，包括各种类型的垃圾信息；实验中已经提供数据集。

1. 数据处理：对收集到的数据进行处理，包括去重、分词、去除停用词、词干化等操作，而后进行特征提取：从处理后的数据中提取出有效的特征，例如词频、tf-idf权重、句法结构、情感极性等，将文本转化为特征向量的形式。此次实验中涉及到停用词、向量化等。

3. 模型训练：使用机器学习算法，例如支持向量机、朴素贝叶斯、决策树、神经网络等，对提取出来的特征进行训练，从而得到垃圾短信分类模型。实验需要对数据进行划分训练集、测试集。

1. 模型评估：使用测试数据集对模型进行评估，包括准确率、召回率、F1值等，调整模型参数以达到最佳效果。
2. **代码内容**

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

读取停用词：

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 = [line.strip() for line in f]

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

return stopwords

导入库：

stopwords = read\_stopwords(stopwords\_path)

import warnings

warnings.filterwarnings('ignore')

import pandas as pd

import numpy as np

from sklearn.pipeline import Pipeline

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.model\_selection import train\_test\_split

from sklearn import metrics

from sklearn.externals import joblib

from sklearn.preprocessing import MaxAbsScaler

from sklearn.naive\_bayes import MultinomialNB

from sklearn.metrics import roc\_auc\_score

from sklearn.naive\_bayes import ComplementNB

from sklearn.neighbors import KNeighborsClassifier

from sklearn.model\_selection import GridSearchCV

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn import svm

from sklearn.naive\_bayes import MultinomialNB

数据集处理：

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

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

data\_1 = data[(data['label'] == 1)].sample(frac=1.0)

data\_0 = data[(data['label'] == 0)].sample(frac=1.0)[: int(1\*len(data\_1))]

data\_new = pd.concat([data\_1, data\_0], axis=0).sample(frac=1.0)

# 构建训练集和测试集

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

y = np.array(data\_new.label)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, random\_state=9,

test\_size=0.2)

# 构建分类器

# 朴素贝叶斯

estimator = MultinomialNB()

#estimator = ComplementNB()

# # 决策树算法

#estimator = DecisionTreeClassifier(criterion="entropy")

# # 随机森林算法

# # 1.随即森林预估器

##estimator =RandomForestClassifier()

# # 2.参数准备

#param\_dict = {"n\_estimators": [90,100,110], "max\_depth":[100,200,300]}

#estimator = GridSearchCV(estimator, param\_dict, cv=3)

# # 支持向量机

estimator=svm.SVC(kernel='rbf',C=1000, probability=True)

# 2.KNN算法预估器

estimator = KNeighborsClassifier()

pipeline = Pipeline([

('tfidf', TfidfVectorizer(token\_pattern=r"(?u)\b\w+\b",

stop\_words=stopwords, ngram\_range=(1,2))),

('MaxAbsScaler', MaxAbsScaler()),

('classifier', estimator)

])

# 模型训练

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

# 模型评价

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

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))

print("在测试集上的 AUC为:")

print(roc\_auc\_score(y\_test, y\_pred))

print('在测试集上的准确率：')

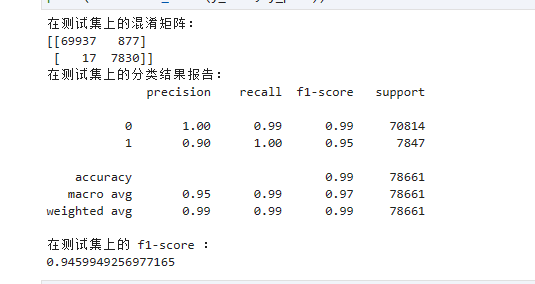
print(metrics.accuracy\_score(y\_test, y\_pred))

# 保存模型

pipeline.fit(X, y)

joblib.dump(pipeline, 'results/pipeline.model')

1. **实验结果**





1. **总结**

与前几次实验相比，这次实验需要我们亲自训练模型，优化参数，提高模型的能力。

此次实验基本达成了目标。训练了多个模型，可以看到参数、代码的调整对模型能力的影响。

通过此次实验，对机器学习的理解更进一步，同时对模型优化的步骤有了一定的体会。