**垃圾短信识别程序报告**

学号：2212452 姓名：孟启轩

1. **问题重述**

使用基于 Python 的 Pandas、Numpy、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，实现垃圾短信的识别。

根据一段中文文本（ 200 个中文字符以内），预测这段文本是否为垃圾短信。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**

在读取数据部分中，读取了全部label值为1的数据并随机排序，读取label值为0的数据并随机排序，并只选取与label值为1的数据同样多的数据，将二者合并为一份新的数据集。主要原因是当label值为0的数据过多时，容易造成训练过拟合的情况，导致训练模型分类能力变差。

其次读取停用词库，并将其进行按行分割。构建训练集（msg\_new）和测试集（label）。构建pipeline，将文本向量化方法 TfidfVectorizer，常用于文本分类的朴素贝叶斯方法 MultinomialNB 传入其中，进行训练和预测。

在构建pipeline时，选用了**TF-IDF算法（词频-逆向文件频率，**一种用于信息检索与文本挖掘的常用**加权技术），其是一种统计方法，**用以评估一字词对于一个文件集或一个语料库中的其中一份文件的重要程度。**字词的重要性随着它在文件中出现的次数成正比增加，但同时会随着它在语料库中出现的频率成反比下降。**如果某个单词在一篇文章中出现的频率TF高，并且在其他文章中很少出现，则认为此词或者短语具有很好的类别区分能力，适合用来分类。

1. **代码内容**

# 导入相关的包

import warnings

import joblib

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.naive\_bayes import MultinomialNB

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

from sklearn import metrics

#from sklearn.externals import joblib

# 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')

sms\_pos = sms[(sms['label'] == 1)]

sms\_neg = sms[(sms['label'] == 0)].sample(frac=1.0)[: len(sms\_pos)]

sms = pd.concat([sms\_pos, sms\_neg], axis=0).sample(frac=1.0)

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

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

y = np.array(sms.label)

pipeline = Pipeline([

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

('clf', MultinomialNB())

])

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

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

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

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

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

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

print("在测试集上的 f1-score ：")

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

print('在测试集上的准确率：')

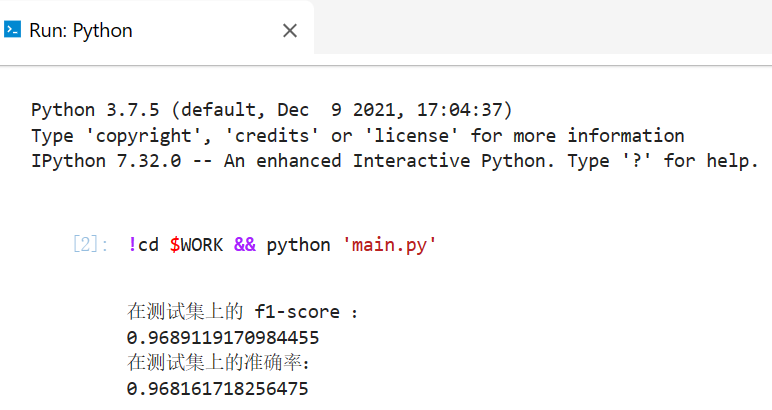
print(metrics.accuracy\_score(y\_test, y\_pred))

1. **实验结果**

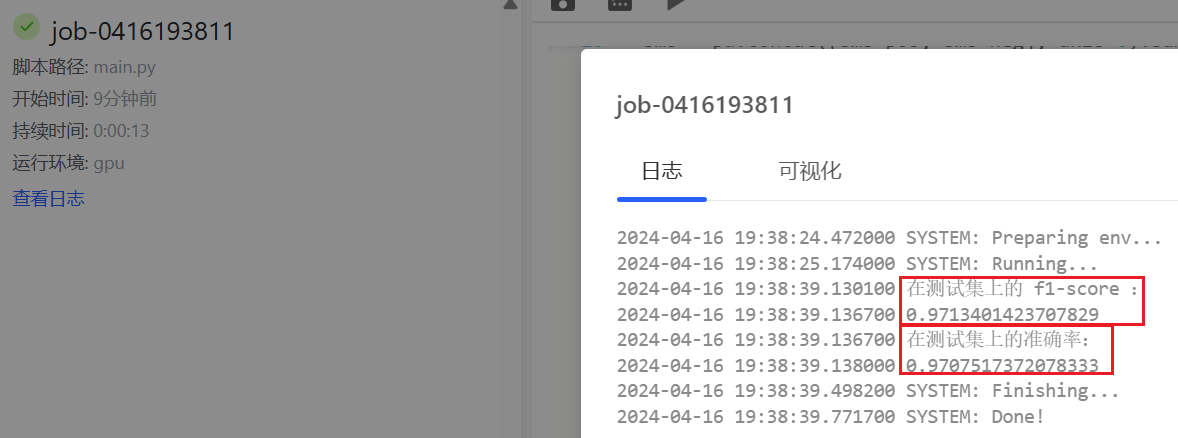
通过训练和测试，实现了对垃圾短信的识别，有较高的识别成功率。



NoteBook控制台结果



官方GPU机器



1. **总结**

本次实验完成了利用 Sklearn 框架训练分类器，实现了垃圾短信的识别。在后续实验中可以适当调节文本向量化中的各种参数设定，选用更好的停用词库，调节分类器的参数，选用更好的统计分类算法来进行优化。

TfidfVectorizer相当于先后调用CountVectorizer和TfidTransformer两种方法。CountVectorizer用于将文本从标量转换为向量，TfidfTransformer将向量文本转换为tf-idf矩阵，从而让我们的预测更加准确。