

**项目开发报告**

**题目：基于Bert和NaiveBayes的新闻情感分类模型**

**课程名称： \_\_\_机器学习\_\_\_\_\_\_**

**专业班级： 计科1906\_\_\_\_\_\_**

**学 号： U201914213\_\_\_\_\_**

**姓 名： 吴官骏\_\_\_\_\_\_**

**指导教师： \_\_李\_玉\_华\_\_\_\_**

**报告日期： 2021.11.30\_\_\_\_\_\_**

**计算机科学与技术学院**

# 项目开发报告

## 1.1项目目的

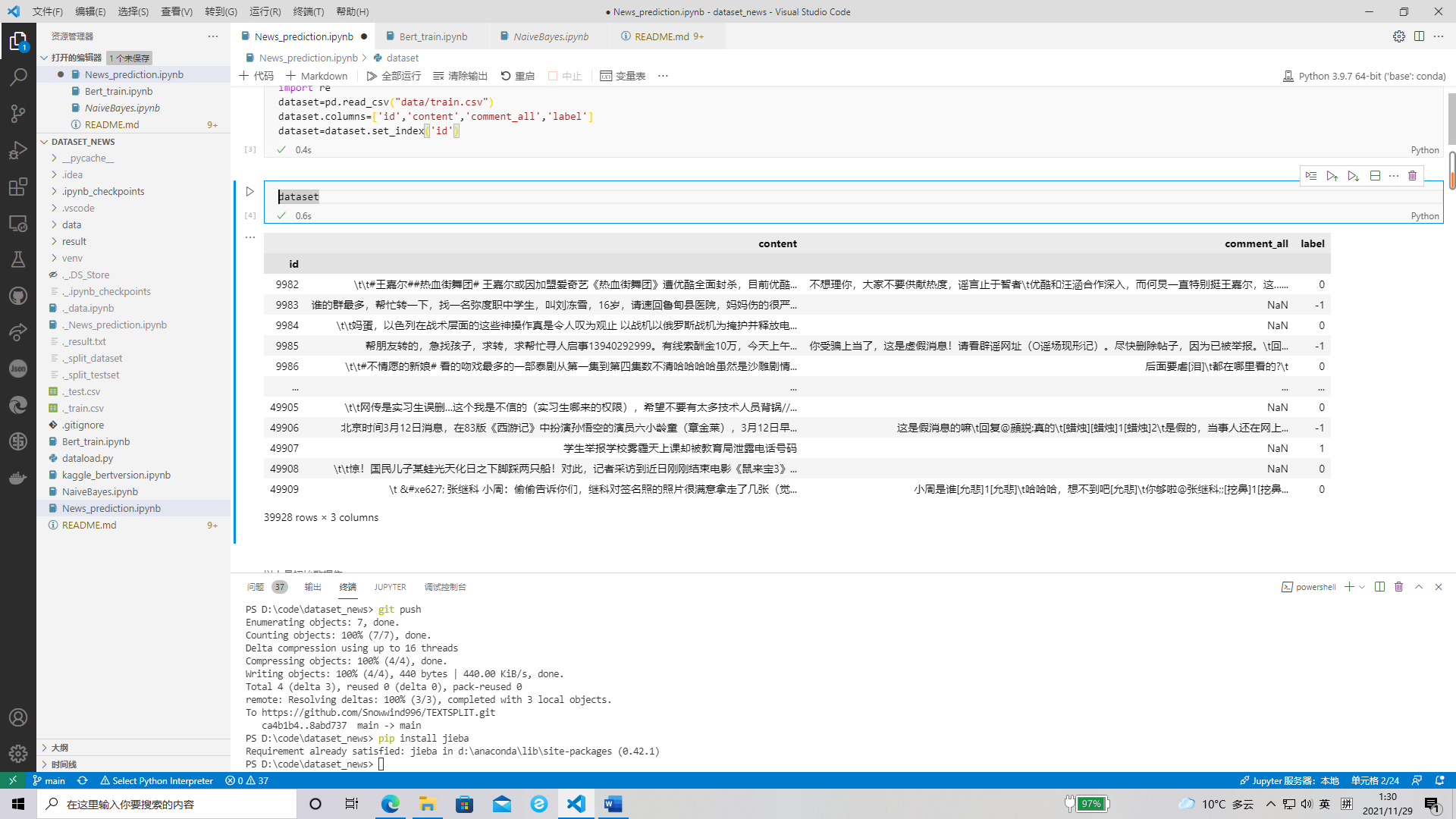
本项目是机器学习课程的课设，任务是实现一个情感分类的模型，给定40000条新闻的信息及其评论，通过实现一个新闻文本分类模型来对其的真假性进行判断。在这个项目中，我使用了两种算法，第一种是词嵌入模型与朴素贝叶斯算法，在测试集上达到了87.4%的准确率、查准率。第二种是基于预训练词向量的Bert[1]微调模型，预训练词向量来自于哈工大讯飞实验室的CN-WWM模型[2]，开发框架来自于复旦大学的fastNLP[3]。

## 1.2问题分析

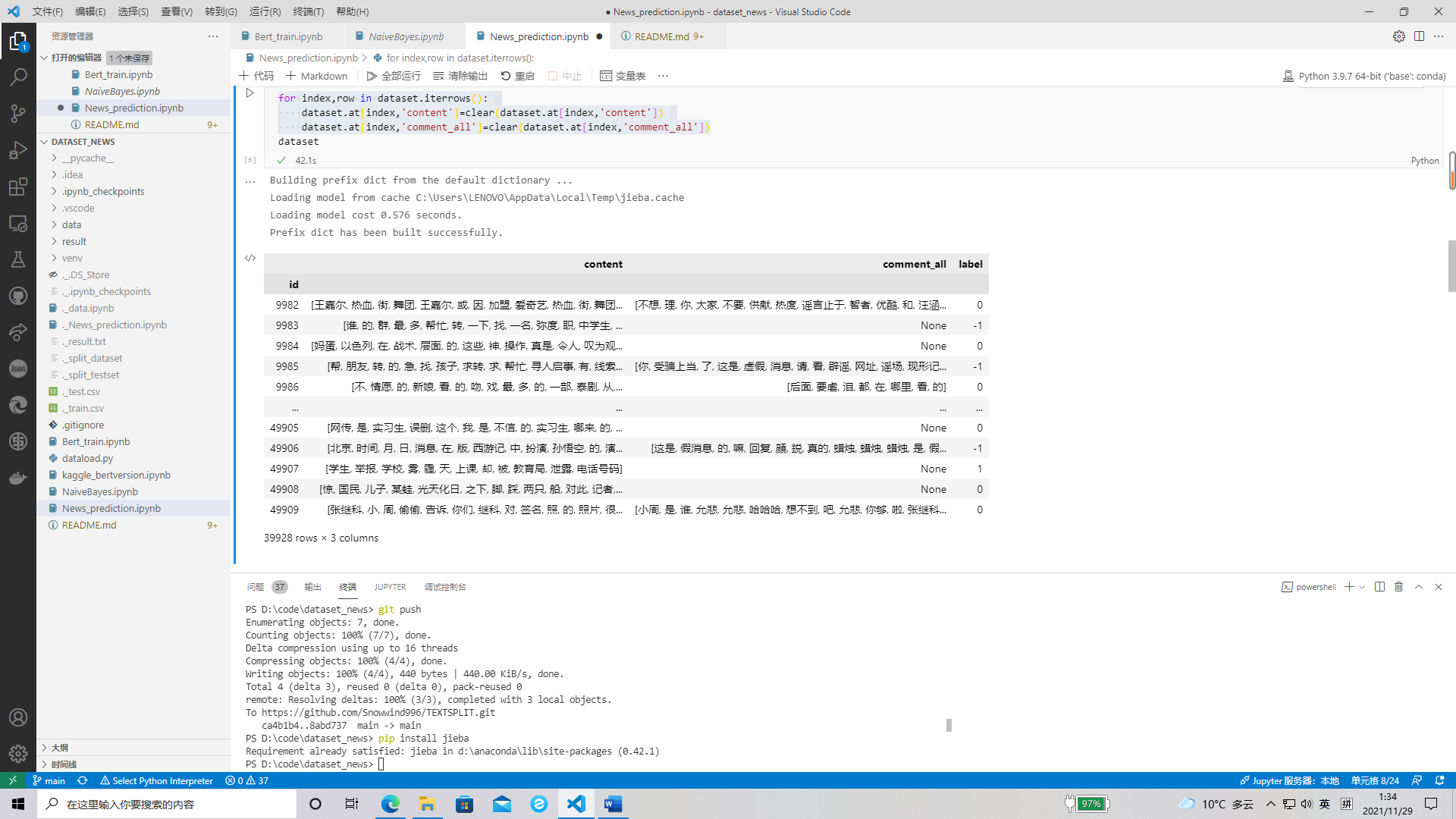
本问题是一个典型的三分类模型，输入为一系列的文本内容。在这里对输入内容做了简化，删去了新闻内容中的非汉字字符，仅仅留下了汉字字符，然后再用分词库对句子进行分词。在朴素贝叶斯实现中，通过TF-IDF词袋模型对每一个词语在各种样本类别的出现概率进行记录，在预测时用朴素贝叶斯算法计算每个句子属于哪一类的最大概率，确定分类。在Bert实现中，分词之后通过加载预训练的词向量模型对汉语文本进行词嵌入，得到合适的句向量编码，最后加载训练完成的Bert参数，在Bert模型的最后一层输出层加上全连接层作为微调，实现预测分类。

## 1.3设计与分析

### 1.3.1 数据分析与预处理

首先通过Pandas库加载数据集，简要查看数据集的情况。如图所示

从中可以看出，文本形式主要为中文，但是还存在很多杂项，比如各种转义符\n\t以及标记#!等符号。这些符号在词嵌入的时候都十分的不方便，虽然某些符号也能表达一定的语义，但是为了简化处理，直接删掉也没关系。因此可以先用Python的正则表达式对数据进行预处理，再用Jieba库对其进行分词，得到结果如下图所示，发现所有content和comment\_all中的词语都已经被分开，而且基本保持了语义的连贯性，比如id为9982中的“王嘉尔”作为单独的人名就被分开，以及id为49909的“张继科”也作为人名被识别出来。虽然分词的粒度有点细，比如“热血街舞团”被分成了“热血”“街”“舞团”，但不影响整体语义的表达，而且要训练一个合理的分词器也要用更深的模型，比如BERT的分类代码。



因为本实验要实现两个模型：BERT和NaiveBayes，因此初步的预处理就到这里为止，接下来分开实现朴素贝叶斯的词嵌入和BERT的词嵌入。但在实现词嵌入之前，首先需要简单介绍一下BERT和朴素贝叶斯的核心思想与流程。

### 1.3.2 算法流程设计

整体流程很简单，预处理后生成文本分词表，再分别通过BERT和朴素贝叶斯专门的预处理流程进行预处理，再代入算法计算。

二者首先通过数据清洗\预处理函数对初始数据进行清洗，数据清洗函数在dataload.py文件的preprocess类中，先调用preprocesingdata\_news函数对数据统一清洗（提词分词），再分别调用processingdata\_bayes和processingbert函数将初步清洗的数据分别转换成训练相应的模型需要的格式。朴素贝叶斯算法的格式主要为列表和字符串，每一个词语之间用空格分开，每一个句子都是单独的列表。标签同样也是列表存储。而BERT模型则用逗号隔开，仍然用pandas库中的DataFrame存储，另外，每一个训练样本都由新闻内容和评论直接拼接而成。

朴素贝叶斯算法流程图如下：



图1 朴素贝叶斯算法流程图

BERT模型处理流程图如下：

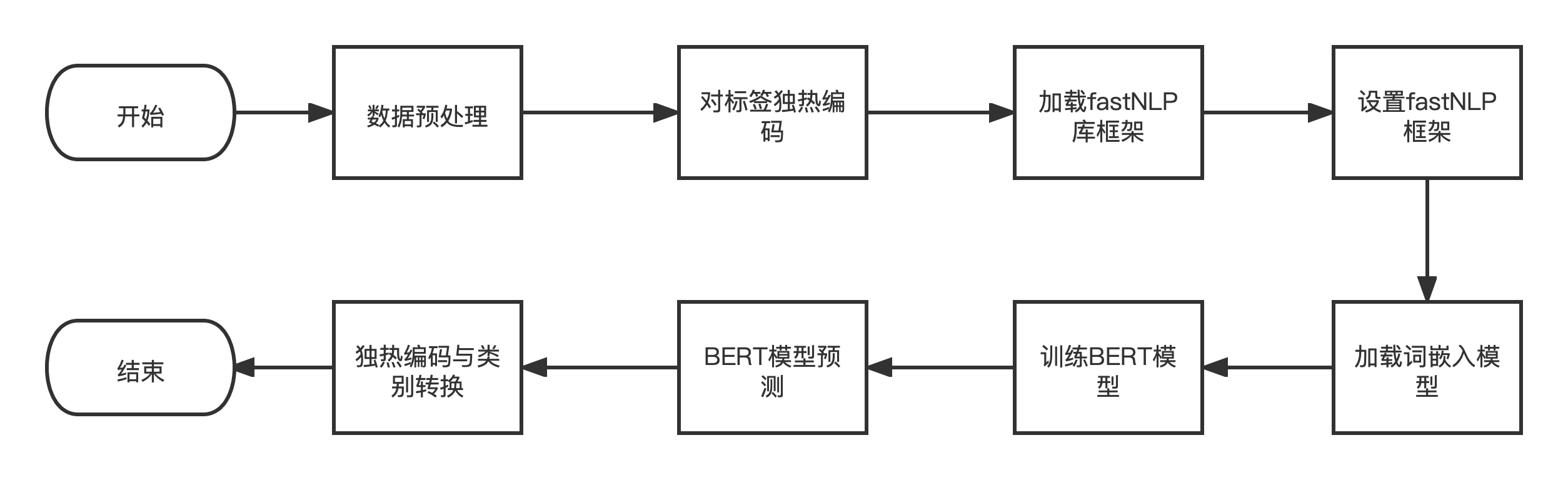


图2 BERT算法流程图

其中，朴素贝叶斯算法的特征提取使用了sklearn的特征提取库feature\_extraction.text。

### 1.3.3 机器学习算法设计

在朴素贝叶斯算法的实现中，作者调用了Sklearn库的多项式朴素贝叶斯分类器MultinomialNB。BERT算法实现中，预训练的词向量为哈工大-讯飞实验室的’cn-wwm’预训练词向量，而Bert模型则为fastNLP中的BertForSequenceClassification函数，其就是简单的在Bert模型之上加了一层全连接层。训练方面，多项式朴素贝叶斯均采取了sklearn的默认参数。而BERT因为训练过于复杂，作者将训练部署至kaggle中，全部可执行代码都在kaggle\_bertversion.ipynb中。可以直接运行。BERT训练采取了Adam优化器，交叉熵损失函数，5次迭代，minibatch设置为8，学习率为0.01。

## 1.4结果分析

1.4.1 朴素贝叶斯模型

朴素贝叶斯的训练与预测在M1芯片的Mac电脑上1分钟内就能运行完成，结果如下：

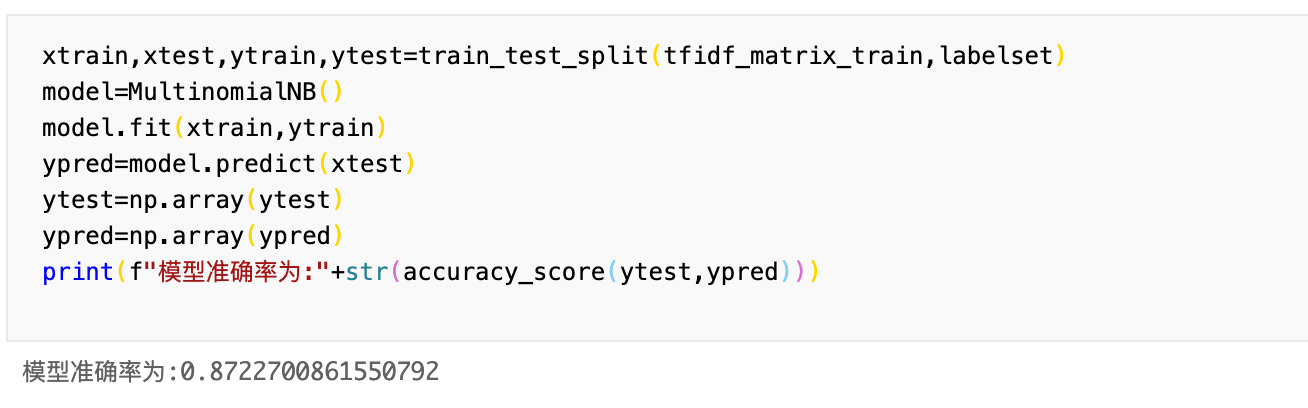


图3 本地运行截图

将结果提交至educoder，结果如下。查准率和查全率都比较高。而且测试集验证集的精度都比较接近。说明模型没有过拟合的现象，泛化能力较强。

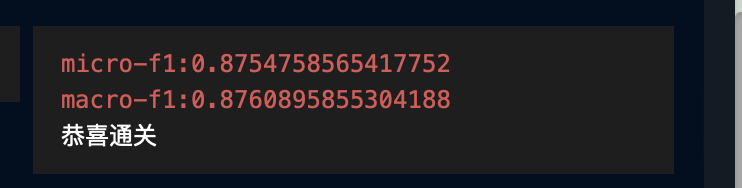


图4 朴素贝叶斯educoder结果截图

结果文档已经保存在文件目录下的result\_bayes文件中。

1.4.2 BERT训练模型

由于BERT在kaggle上训练，作者还是第一次使用kaggle，因此没有将模型保存下来。由于BERT模型过大，作者一共使用了30000个数据作为训练，5000个数据验证。在kaggle上训练了3.5小时候迭代完成，模型的准确率为91.5%。将模型对测试集进行测试，最后将结果提交到educoder中，结果如下：

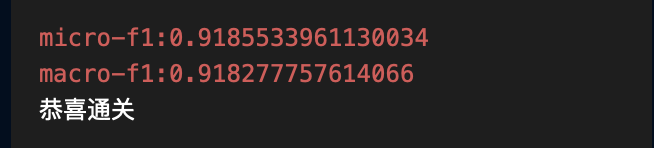


图5 BERTeducoder结果截图

同样地，BERT模型泛化能力也十分强劲，而且在较少的几个epoch中就能达成比较好的准确率。其优势主要来源于经过完全训练的预训练词嵌入向量与BERT模型自身的双向transformer的强大的拟合能力。

## 1.5思考与总结

本次项目是我的第二个机器学习项目，第一个是基于UNet的语义分割，自己复现的UNet论文完成的，NLP这块我本身只有一些基础知识储备，实践项目还没有做过，正好就借着这个机会实现了一下BERT。朴素贝叶斯作为一个基线，同样可以相互比较（多混点分）。感觉还是万事开头难，数据清洗处理以及特征工程部分最考验基本功了，当时有很多同学直接去了content的部分而忽略的comment，这个我是比较反对的。我想了想，可以直接把这两部分连起来就OK了，然后就是做正则表达式提取以及Jieba分词。当然只提取中文会损失一些语义，但是有助于训练，所以默认接受了这部分误差。

在模型训练阶段，由于朴素贝叶斯比较简单，就不多做赘述了，主要讲讲BERT。

因为自己的电脑是MAC电脑，所以Tensorflow比较难装，装了好几天都没装好，就没有办法用transformers库了。找来找去最后还是选了复旦大学的fastNLP，因为这是基于pytorch做的。结果研究了好几天人家的代码，从Dataset库到vocabulary库，以及各种训练Demo，各种出错调错，耗费了大量的时间，甚至看了不少fastNLP的源代码，当然收获也是巨大的。结果后面发现还是MAC电脑的问题，没办法训练BERT。于是就换了windows游戏本，真正跑起来之后才发现BERT果然是名不虚传的难train。而研究了一下华为云，发现搞不来，所以就去白嫖了kaggle，一会就弄好了。用kaggle训练了两次，第一次一个下午，结果回来电脑自动睡眠，导致训练终止。于是晚上把参数调了一下，守在电脑面前看，功夫不负有心人，最后train好了，看着90多的准确率还是十分的欣慰的，也是自己这几天辛勤劳作的结果吧。

因为自己最近做科研也需要用到transformer，而BERT就是transformer的一个魔改版，所以熟悉了模型框架也有助于使用。但是个人感觉FASTNLP似乎没那么好用，可能还是不够开源，封装的太多了，一层一层包起来的。听说hugging的transformers会好一些，自己之后会再去学一学。千里之行始于足下，这个NLP的入门任务算是比较好的完成啦，以后继续加油！

## 1.6 相关文章

[1] https://arxiv.org/pdf/1810.04805.pdf

[2] [ymcui/Chinese-BERT-wwm: Pre-Training with Whole Word Masking for Chinese BERT（中文BERT-wwm系列模型） (github.com)](https://github.com/ymcui/Chinese-BERT-wwm)

[3][fastnlp/fastNLP: fastNLP: A Modularized and Extensible NLP Framework. Currently still in incubation. (github.com)](https://github.com/fastnlp/fastNLP)