

**python语言程序设计**

**大作业 实验报告**

****

学 院 计算机学院

专 业 计算机科学与技术

学 号 2213583

姓 名 卢艺晗

班 级 2022级计算机科学卓越班

**目录**

**一、问题描述------------------------------ 1**

**二、数据集说明---------------------------- 1**

**三、使用的模型及改进过程------------------ 2**

1、Bi\_LSTM---------------------------------- 2

2、Bert------------------------------------- 6

**四、反思总结------------------------------ 12**

1. **问题描述**：

在信息爆炸的时代，虚假新闻已经成为一个严重的社会问题。虚假新闻不仅误导公众，破坏社会秩序，还可能对个人、企业和国家的安全造成威胁。因此，自动化检测虚假新闻在当今时代格外具有意义和实用价值。

此问题为微信消息中的虚假新闻检测，对给定训练数据集中的信息进行分 析，并预测测试数据集中新闻的真假（label:0表示真，1表示假）。同时 给出Precision,Recall,F1-score,AUC的值。

1. **数据集说明：**

**1、给定数据集**

给定数据集分为训练集（train.news.csv)和测试集(test.feature.csv)。其中训练集中的信息包括：

“Ofiicial Account Name”、“Title”、“News Url”、 “Image Url”、

“Report Content”、“label”。

测试集中没有 “label”列，需要我们自己预测。

训练集数据10587条，只提取其中的“Title”列和“label”列进行分析并训练模型；测试集数据10141条，只提取其中的 “Title”列进行预测。

1. **提交文件说明**

提交的python源代码包括：

1. Bert模型源代码：Bert.ipynb

（2）原Bi-LSTM模型源代码：big\_work\_01.ipynb

1. **使用的模型及改进过程**

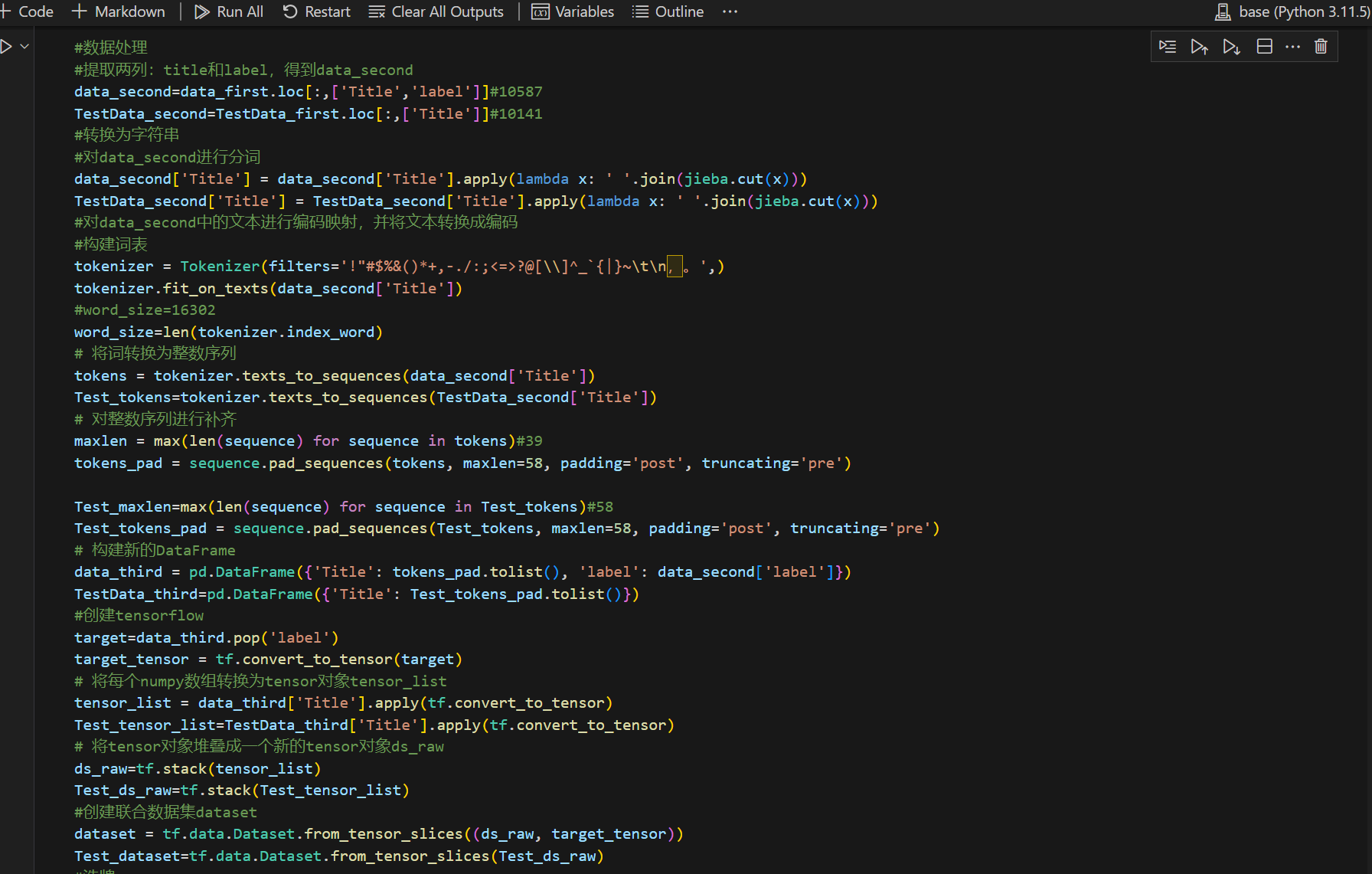
在整个比赛期间，我主要使用了两种模型：Bi\_LSTM和Bert。

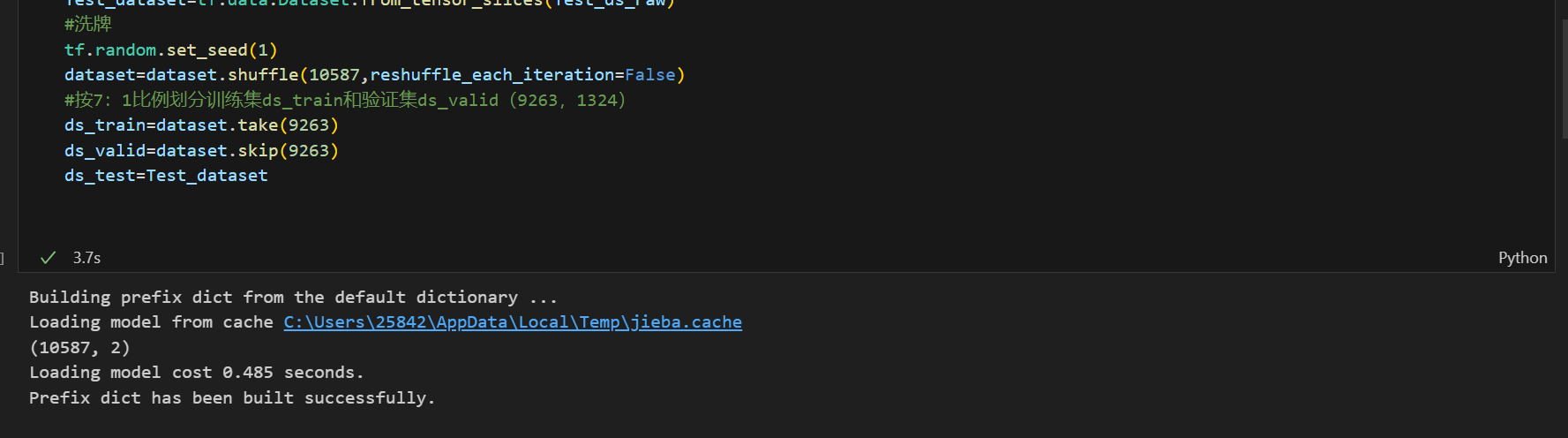
第一阶段使用Bi\_LSTM模型，测试集AUC在0.74到0.79之间，再调节参数、对数据处理进行优化，未能使AUC值提高。

第二阶段使用Bert模型，AUC值在0.84到0.86左右。

1. **Bi\_LSTM模型**
2. **数据处理**

使用jieba分词包对’Title’列进行分词，用tokenizer编码成长度相等的整数序列，并转换成tensorflow的Dataset数据类型（适于输入模型）。对训练集进行洗牌，并按照7：1比例划分成训练集ds\_train和验证集ds\_valid.





【图1 Bi\_LSTM数据处理部分 关键细节及运行截图】

1. **建立模型**

首先建立嵌入层(Embedding)，将输入序列转换为词向量；

其次建立一个双向LSTM层，并设置返回所有时间步的隐藏状态，便于再次建立LSTM层；

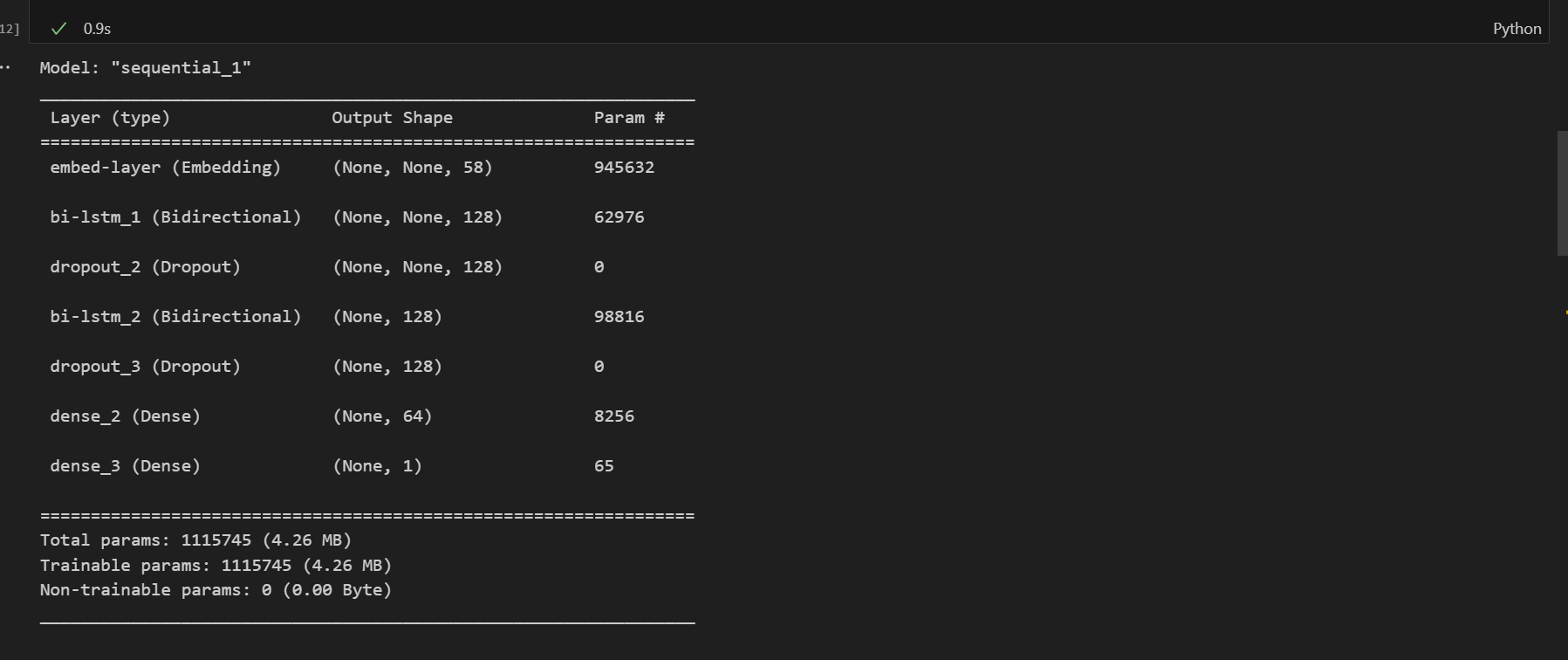
然后是Dropout层，以0.1的比例随机丢弃神经元，防止过拟合；

然后是第二层Bi\_LSTM，加强训练效果。只返回最后一个时间步的隐藏状态，便于得到最终结果； 再一层Dropout;

最后两层Dense层，分别使用激活函数ReLu和Sigmoid,将双向LSTM层的输出先转化为64维，再映射为一维，进行二分类。

此外，设置了L2正则化惩罚项系数，用于解决模型过拟合问题。

****

****

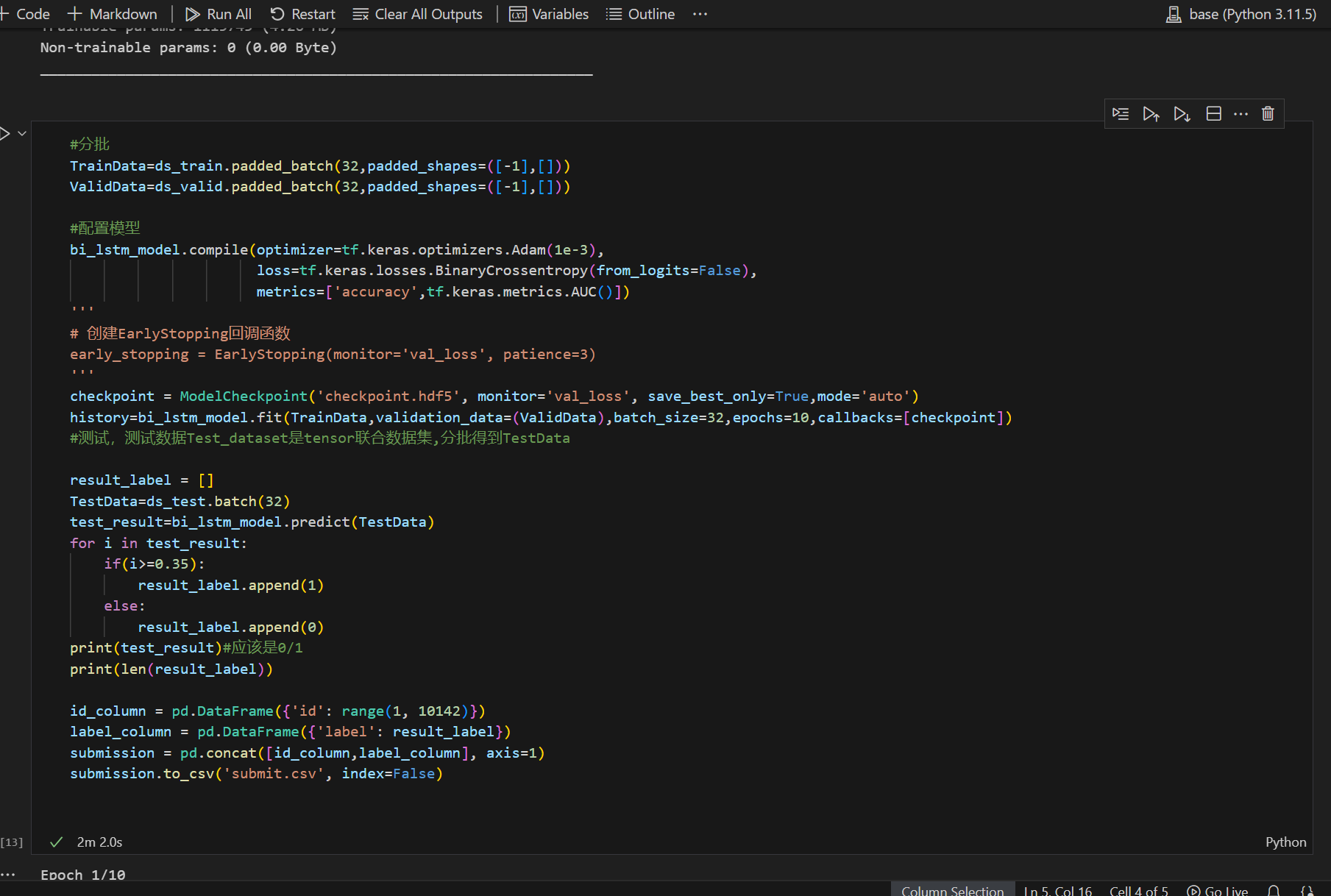
【图2 Bi\_LSTM训练模型部分 代码细节及运行截图】

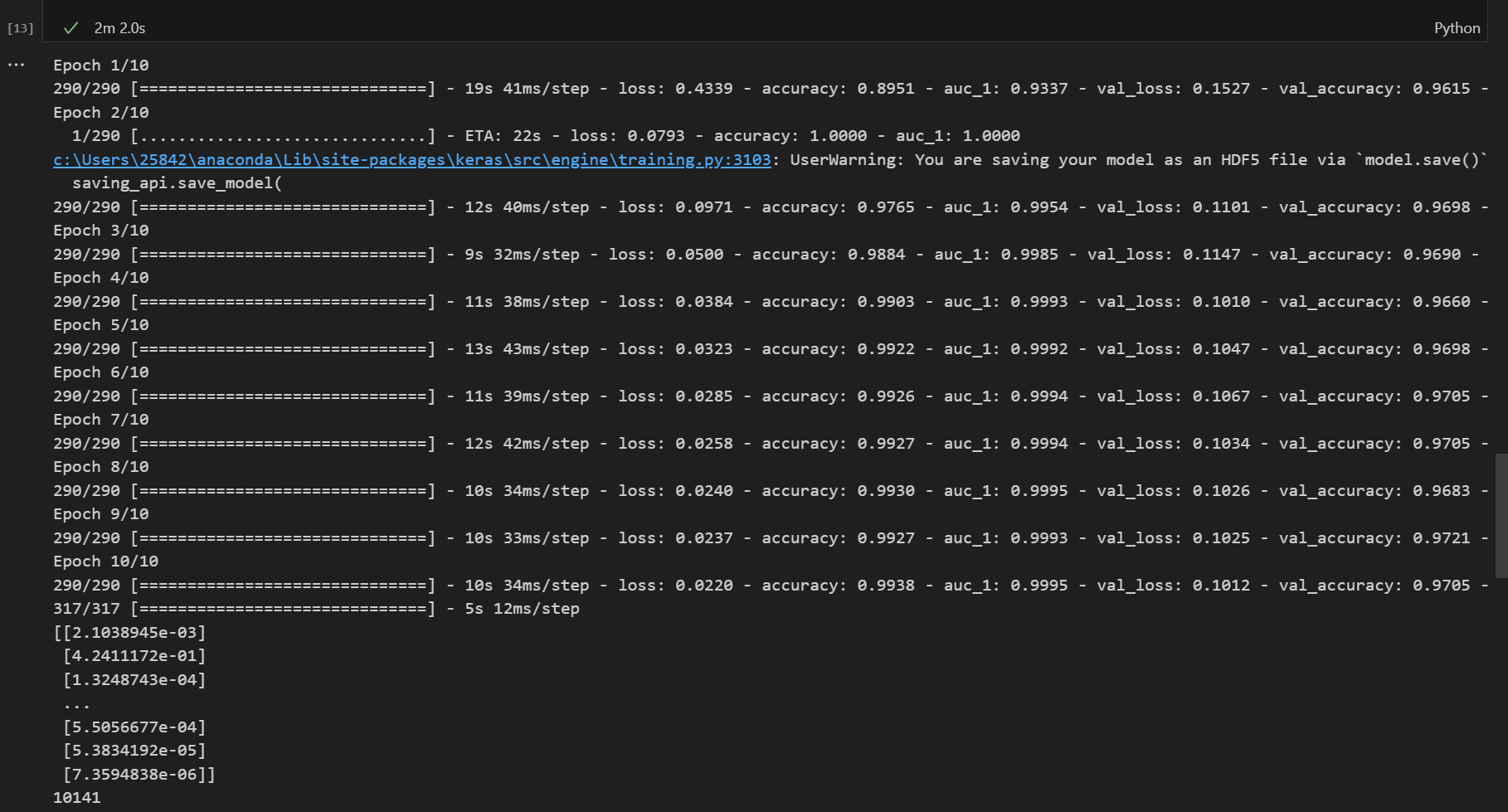
1. **模型编译和预测**

使用Adam优化器动态调整学习率；使用二元交叉熵损失函数度量预测结果准确度；并输出accuracy和auc作为评估指标。

设置批次大小为32，训练10轮，并将预测的概率值以0.35为界转化为二分类标签。最终将预测结果保存在文件’submit.csv’中。

（说明：1、由于训练集数据不均匀，模型倾向于将结果预测为0，因此调整阈值为0.35作为划分0和1的界限，能够达到较好效果；

2、尝试使用early stopping控制训练轮数，但使用结果未达到理想效果。）****

****

【图3 Bi\_LSTM模型编译、预测和分类输出部分 代码及运行截图】

1. 模型改进过程

**有效**（对应的AUC提升）：

（1）增加一层Bi\_LSTM层，提高模型性能**（0.74->0.76)**

（2）丰富’Title’列，（增加了Report Content的内容）

(3)增设L2正则化系数**（0.79)**

**弃用**（效果不显著，未提升AUC）：

（4）改用word2vec

（5）再增加LSTM层

(6)添加early stopping（训练10轮，patience=3)

1. **Bert模型**
2. 数据处理

加载数据集，提取’Title’列和’label’列，并按照7：1比例划分训练集train\_data和验证集valid\_data,同时得到测试集train\_data.



【图4 Bert模型数据处理部分 代码细节】

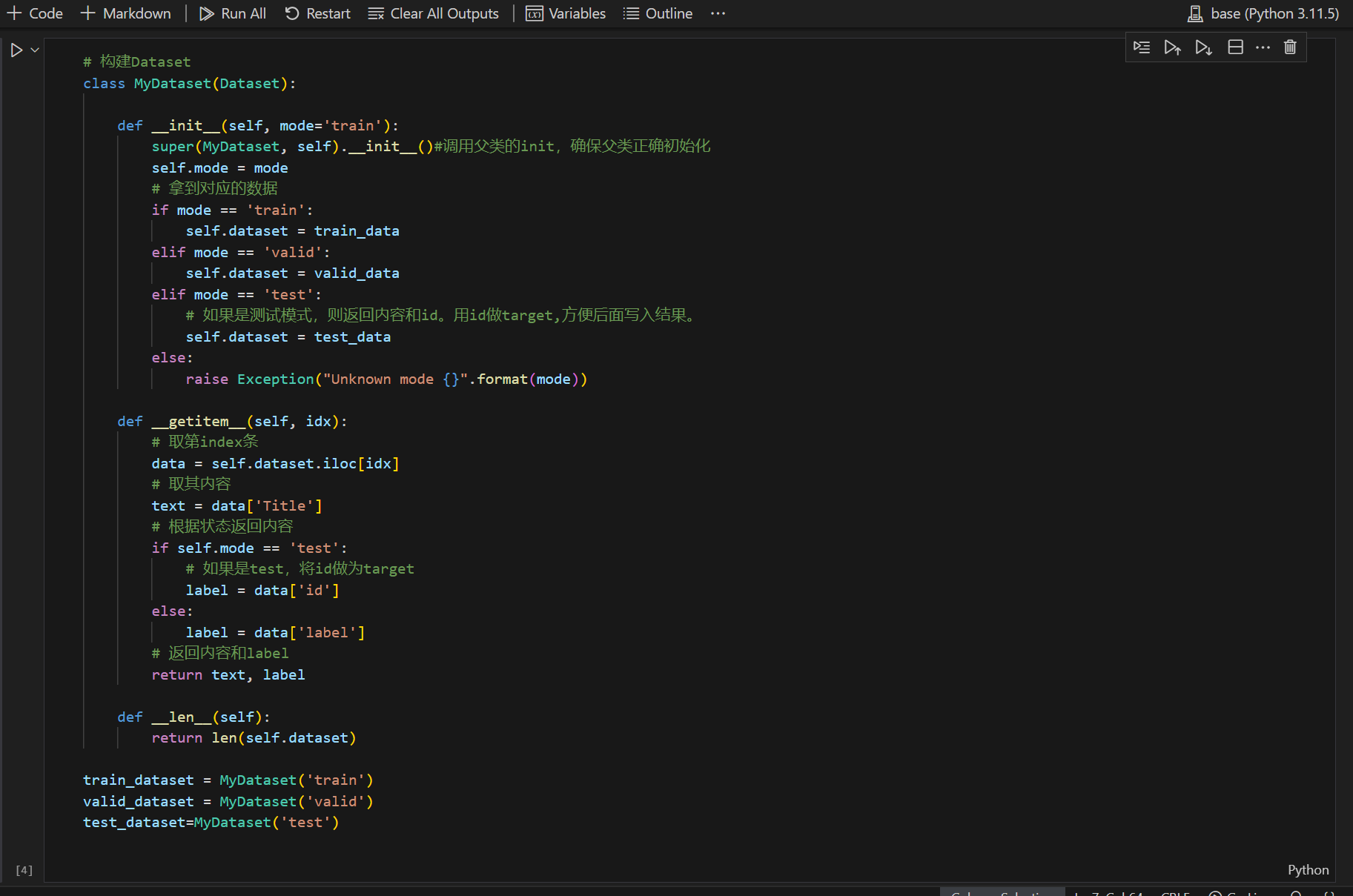
1. 自定义Dataset和Dataloader

**Dataset:**

加载预训练的ERNIE模型，并自定义继承的Dataset类，使得自身数据可以和模型融洽。

\_\_init\_\_函数，通过类实例化时输入的参数确定数据集；

\_getitem\_\_函数，提取数据中的’Title’列和’label’列内容，返回text和label；其中测试集没有’label’列，定义’id’列，便于输出； \_\_len\_\_函数返回数据集长度。



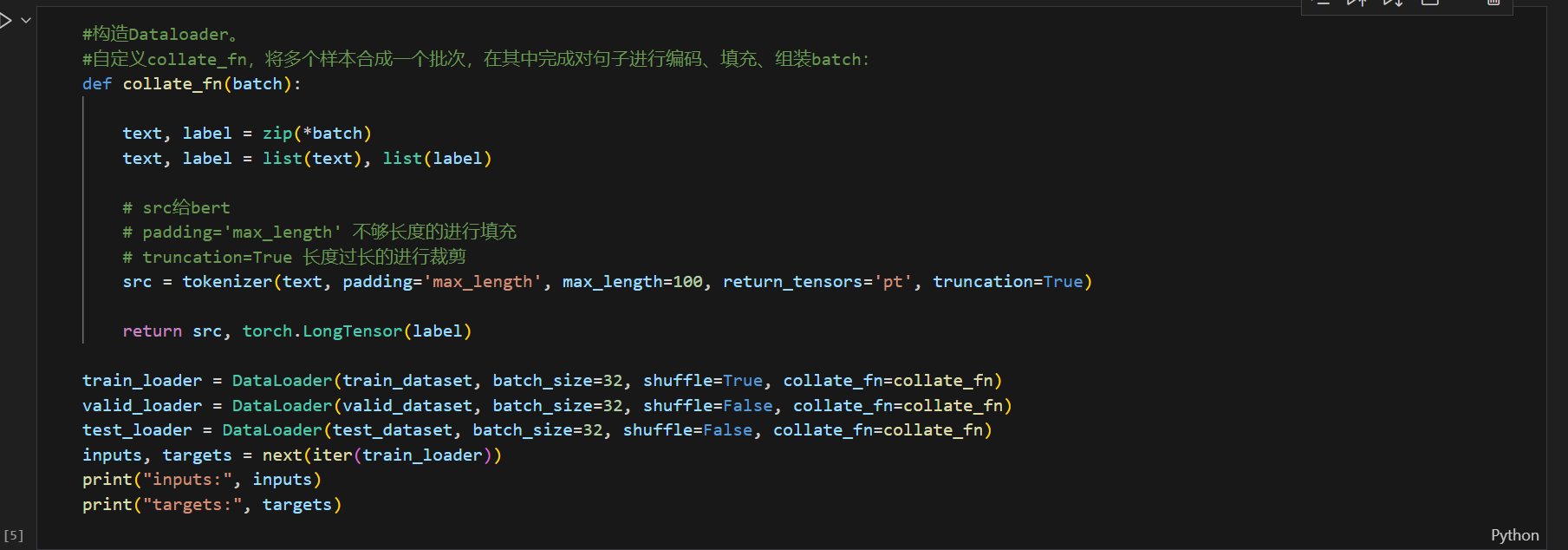
【图5 Bert模型自定义dataset部分 关键代码】

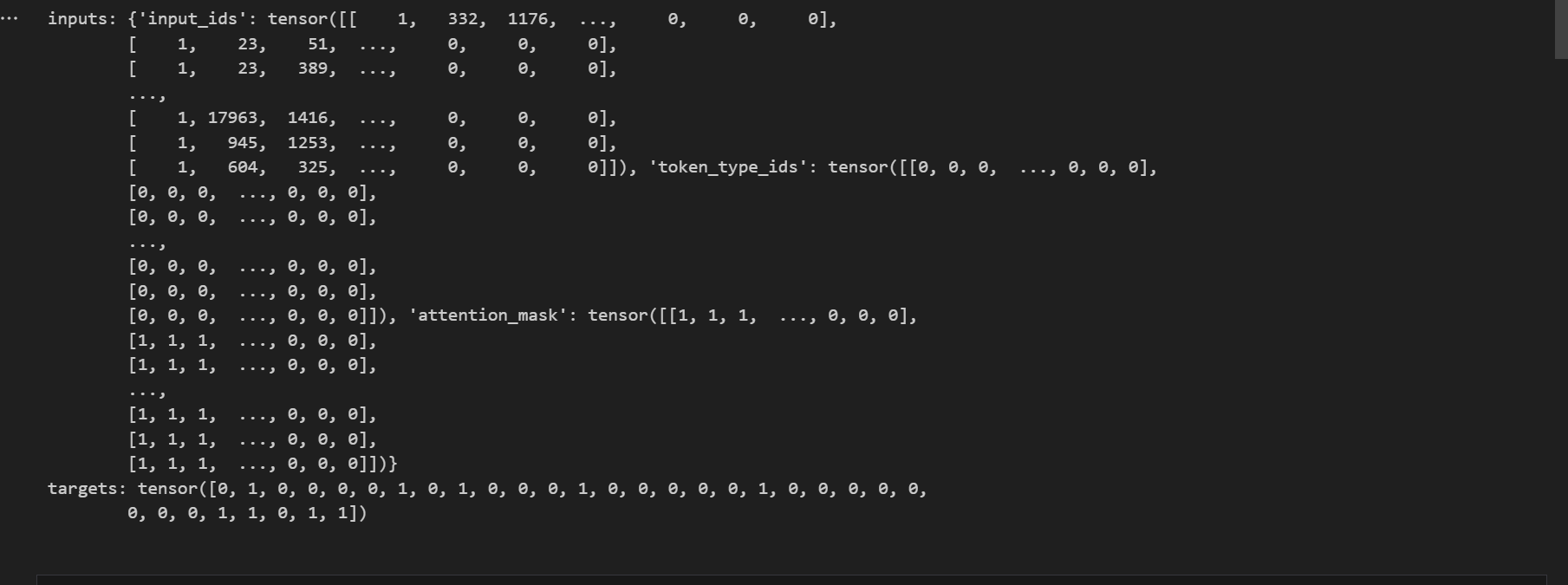
**Dataloader:**

将数据加载入模型，需要自定义Dataloader类的collate\_fn函数，将多个样本合成一个批次，并对句子编码、填充为等长、组装batch。

分别定义train\_loader,valid\_loader和test\_loader对象，设置batch\_size为32并且训练集每次洗牌，能达到较好效果。

形成嵌入层，得到词嵌入、位置嵌入和分割嵌入的相应tensor对象；





【图6 Bert模型自定义dataloader部分 关键代码及运行结果】

1. 定义模型

继承自nn.Module类，定义Bert模型MyModel类，包括加载预训练的Bert模型，定义最后的预测层，以及定义前向传播过程forward函数。

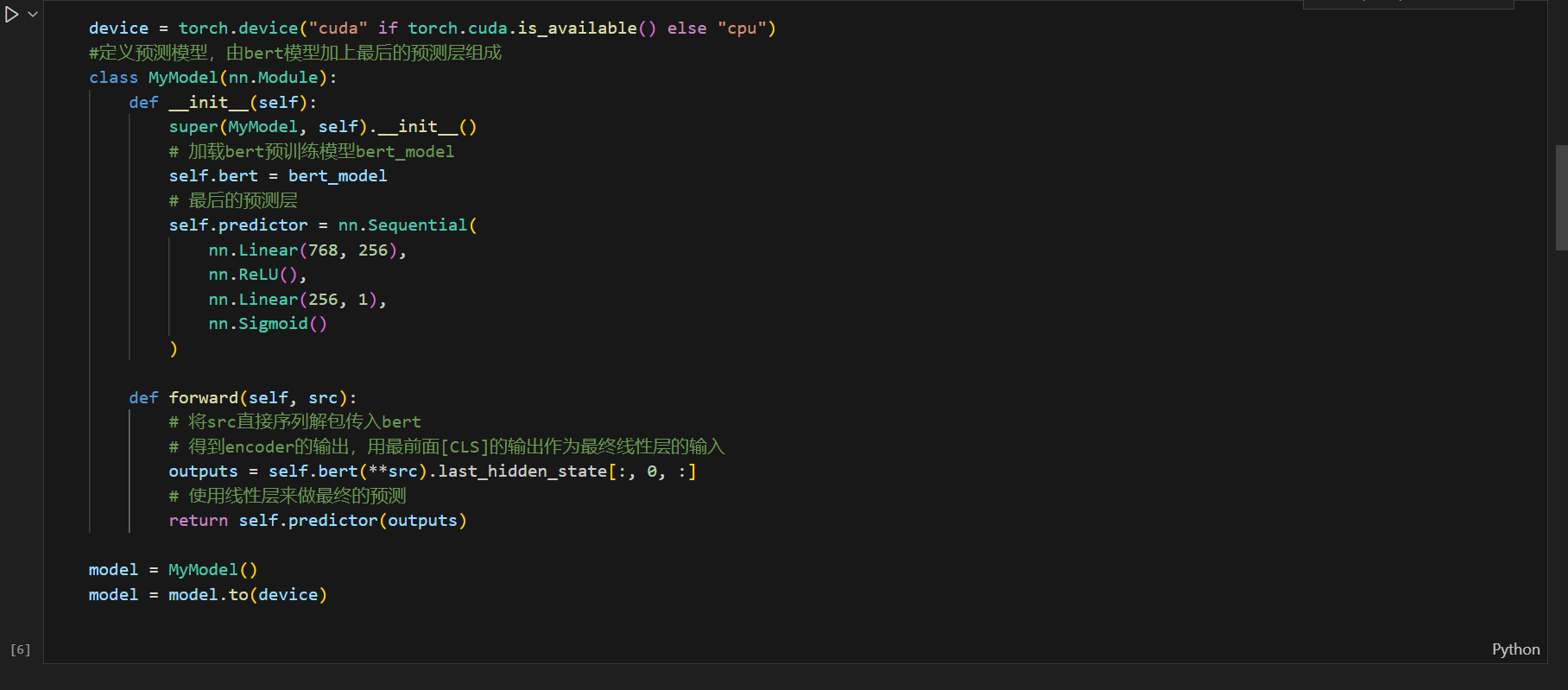
在Sequential容器中搭建预测层，包含：

一个线性层，将输入维度的768维特征映射到256维；

Relu激活函数，进行非线性变换；

一个线性层，将256维映射到1维；

Sigmoid激活函数，将输出映射到0~1之间，作为预测概率。



【图7 Bert模型定义 关键代码】

1. 模型训练和验证

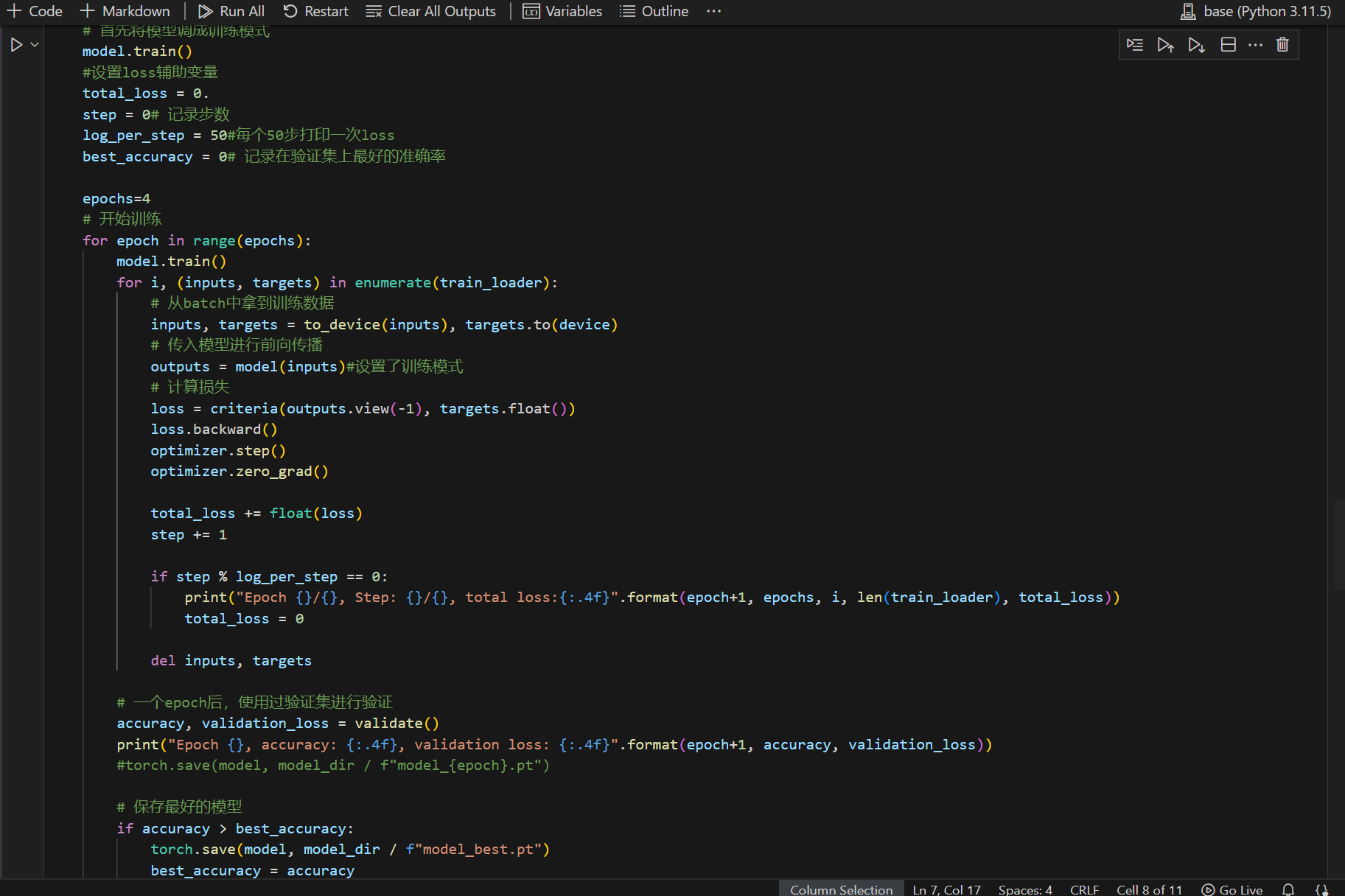
定义模型的循环训练，设置epochs=4；

每轮训练中，分别进行数据前向传播和反向传播；

前向传播输出outputs，与targets比较，计算损失loss，每隔50时间步打印一次；

反向传播中，optimizer.step（）更新模型参数，根据梯度和优化器设 置 来更新模型权重，其中优化器选择Adam；zero\_grad()清空模型梯度，为下一次迭代做准备；

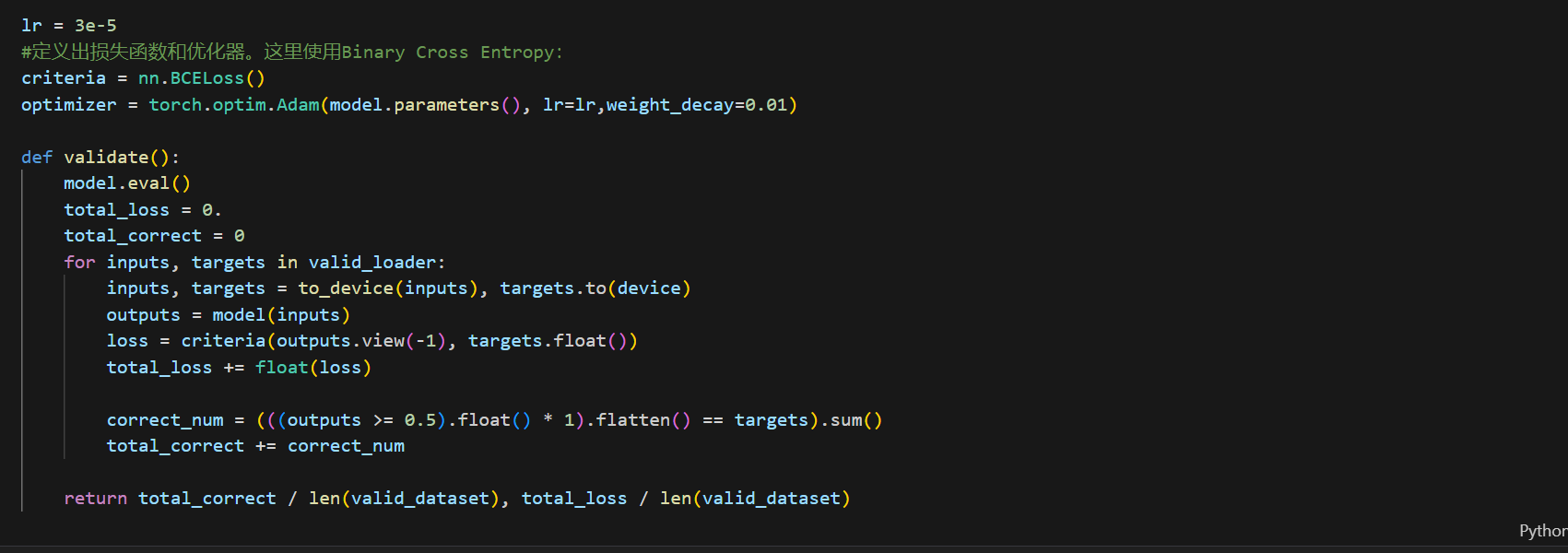
训练过程中，保存最好的模型用于后续测试集的预测；



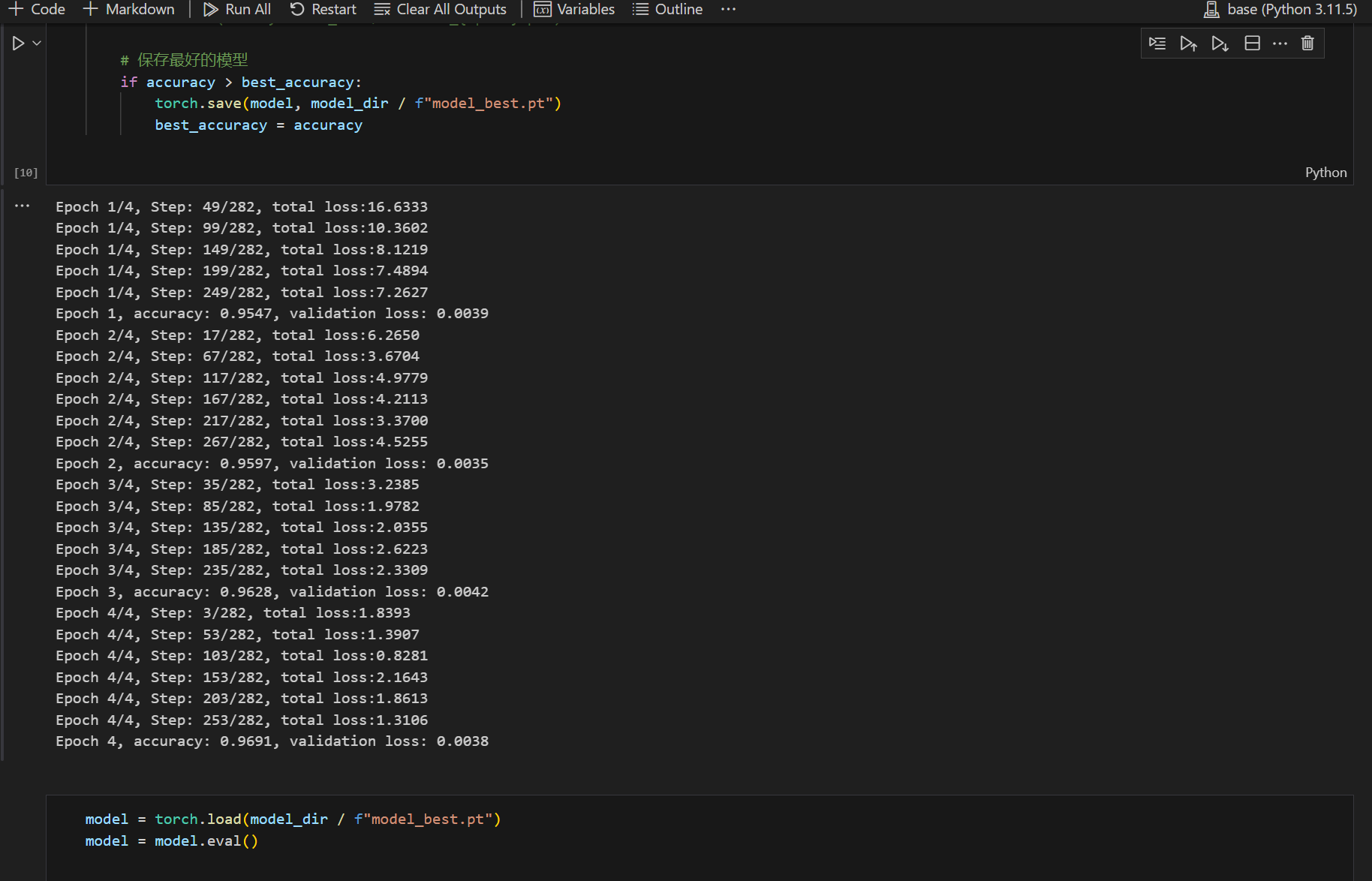
【图8 Bert模型训练部分 关键代码】

每个epoch后使用验证集进行验证，验证函数代码如图9；

进行前向传播、计算准确率和平均损失；计算准确率时，correct\_num记录预测正确的样本数目，以0.5为阈值进行二分类结果计算；



【图 9 Bert模型 验证函数部分 关键代码】



【图10 Bert模型 训练部分 运行结果】

1. 预测

加载测试集，进行前向传播；将数据移动到指定设备上加速计算（后来发现我的电脑配置无法用GPU，因此实际上还是在cpu上计算的）。

以0.3为阈值进行二分类结果输出，并将label和id两列封装保存在文件’submit\_task2.csv’中，作为提交文件；



【图11 Bert模型 测试 关键代码】

1. 结果分析

保存**验证集**上结果最优的一次，计算相关参数：

Precision:模型预测为正例的样本中，真正为正例的比例；

Recall:模型能够正确预测为正例的样本占所有正例样本的比例；

F1-score:精确率和召回率的调和平均值，用于综合考虑模型的精确性和召回性能；

AUC:ROC曲线下的面积，用于评估二分类模型的性能；AUC的取值范围在0和1之间，值越接近1表示模型性能越好；



【图12 Bert模型 模型分析 关键代码】

1. 模型改进过程

**有效**：

（1）降低二分类阈值（0.5->0.3) **（0.84->0.86)**

**弃用**：

（2）添加L2正则化系数---模型尚欠拟合

(3) 将bert和Bi\_LSTM模型进行软投票---效果不如bert单独作用

**可能的改进**:

(4)增加训练轮数，提高拟合程度---由于电脑配置原因，无法使用gpu ，cpu运行耗时巨大，未能实现较高epoch的训练

1. **总结**

通过大作业，我学习了很多深度学习的知识，对神经网络有了一定的了解。同时也熟悉了用机器学习模型解决问题的基本流程：数据处理、特征提取、模型训练、预测和模型分析等，并结合不同的实际问题选择相应的方法。

从这两次模型的应用中，我也发现自己分析和调整模型的能力较弱。例如LSTM模型，通过增加LSTM层使得模型性能提升之后没能再进一步，发现了模型过拟合，应用L2正则化方法，Dropout等有微小提升，但是加入earlystopping 并没有达到理想的效果。

Bert模型，通过分析loss和增加L2系数后的情况发现，显然我的模型还欠拟合，一个比较可观的方向是增加训练轮数，但是由于各种原因，我的电脑未能使用gpu，在cpu上训练耗时巨大，且训练到一定轮数电脑会崩，所以尚未能达到较高水平，这也是当前比较遗憾的地方。

总体而言，当前的很多学习是为用而学，学到的一些知识还不够全面，希望以后能够进一步学习机器学习的相关知识，并提升我的模型分析调参能力。