第一天：学习numpy基础，通过b站（代码相关部分有的视频直接用sklearn速成，有的连numpy都不用拿列表硬钢）、相关书籍（《机器学习》周志华，第四章）片段、AI来接触决策树、随机森林相关概念

第二天：进一步了解决策树细节，以及代码实现的可行思路，试图通过自身理解独立利用numpy推进代码，完成了数据获取，gini , calc\_best\_feature的相关函数。而后参考已有代码结构，给出不完善代码。

第三天：将按自己有限理解写出来残缺（完全跑不动）的代码喂给AI，反思逻辑的不完善处，学习现有代码，对决策树的含义有更深层了解，完善代码。而后把决策树改进为随机森林，其实我对随机子集和有放回抽样的根本原理（或者说数学原理）了解还不够深，后面还要多了解些机器学习的相关知识，计划先把周志华那本著作看完。

其实决策树的本质是一个节点，拿到的是它的根。它给我的感觉很像c语言的链表：每个结点只知道自己的下一个结点的信息，通过指针把世界各地的结点创造出来再串联起来，最后返回给用户拿到的是头结点，对应的决策树的根节点，用的时候就从头往后走。这个是我在给同学解答决策树相关疑惑的时候突然想到的，也算教学相长。

第四天：通过catch代码爬取评论信息，（轻薄本电脑算力太弱了，只爬取了每个动漫作品的前三页评论），然后上网了解BERT算法，这是对语言的处理，把文字信息转化成向量的数学语言与机器进行交流，总觉得这种数据化的投喂有一种行云流水的优雅。把爬取的jsonl数据获取出来成列表格式，下载了google-bert-chinese,正在学习如何利用已有模型进行代码实现。

第五天：我的决策树和随机森林是直接用中位数当分点的，这样准确率平均在%80附近，运气好才能到%90，得改，换用gini指数算每个分支的最小gini对应的分点。时间很赶。

在每一次都选取最佳的分裂点，得到的准确率提升到0.9到1.0之间。

通过AI写了一个BERT代码。可能因为我的爬虫数据只选择了前三页，其中好评居多，数据太少，也可能是我的代码逻辑太简单，没有考虑到更多更复杂的因素，导致最终的打分全在9~10之间，准确率只有0.02到0.1。不过好歹是自己第一个以某种形式训练出来的文本处理模型，需要我更深入的学习。这种有限时间内只能了解到一些表象的功能，但不能得知内在的原理，我不是很喜欢这种感觉。

是训练次数epoch少了，之前只有10次，太少了，把训练次数提高上来，准确率可以提到到0.2，如果把容错率调到两分误差之内，算来有0.6几的准确率，一分的容错有0.4左右，十万次训练可以达到一分容错率的0.64。但我只是看着自己IDE上的代码使得数据移动，那种兴奋还是会差一些。先就这样好了，静待今晚截止就好，等日后慢慢了解一些更深层的原理再来进行相关学习。