大家好，我叫A。我的大作业题目是哔哩轻小说网基于标题和摘要的日本轻小说多标签预测。

首先介绍一下研究背景。不知道同学们有没有遇到过这样一种情境，通过自己喜欢的标签分类来寻找自己喜欢的小说。你很喜欢那个标签，然而在那个标签分类页下的书，不是已经读过了，就是稍微看了看觉得不算太对胃口然后放下了，换言之，全是熟悉的面孔，你渴望这一标签下出现新的小说。你到茫茫书海开始搜寻，突然发现一本符合自己胃口的小说，你心想，奇怪，这没有被归到我所喜欢的那个标签吗。定睛一看，奥，原来是，没有标签。这是一个很严重的问题，我也是最近才发现的。很多这种小说网、电影网之类的，他都有这种漏打标签的现象，经过研究我发现他们都是人工打的标签。对于标签少的情境，一个小说对一个标签的情境工作量到还比较少，像这种标签多并且一个小说还对应不定个标签的，特别是在一次性收录大量小说的情境下，工作人员真的是会忙糊涂的，漏打标签基本上已经成为了常态化。然后为什么我会说这是一个很严重的问题呢，仅仅是你找不到它而已吗。很多人习惯通过标签去找，没有发现这本小说，就会导致它的阅读量不高，在排行榜的名次就会低，又有一部分人是通过排行榜来判定是否优质小说，那他们就不会去读、甚至根本就不会注意到这本小说，这再一次使这本小说损失读者，它的排名就高不起来，它就不会被读，就这样循环往复。然而实际上它可能很优秀，本应有许多读者喜欢。对于这个网站呢他只是收录这些翻译的日本轻小说对作者可能没什么影响。但是假设这是一个作者直接发布小说并获取收益的网站呢，收益上蒙受损失，阅读量不高，作者同时也会产生自我怀疑。轻生的作家大家都没少听说吧。这些自暴自弃的作家甚至会引起社会的动荡。所以我说这个缺少标签是一个非常严重的问题，通过这个蝴蝶效应我们真的可以扯出一系列这种连锁发生的社会问题来，这不是危言耸听。所以我的这个大作业所要实现的就是通过机器学习用小说的标题和摘要预测它所对应的一些标签，我一共从sklearn调用了10个机器学习包括集成学习的这些模型，并在最后给出了比较和分析以及一些应用时的建议。另外过程中还有一些没有预料到的收获，这个之后会谈。

首先是数据爬取，我把哔哩轻小说所有2938个系列的小说的标题、摘要和标签全都爬下来了。然后我们直接看一下这个前5条数据还有这个详细信息。你可以看到我前面说他们忘打标签不是胡说八道的，2938个里只有1837个标签不是空的，对吧，事实胜于雄辩。

然后是数据清洗与预处理。这里我把没有标签的行直接去掉了，然后标题和摘要直接放在一起做特征，用结巴给他分词，然后删除了停用词，停用词表用的是使用比较多的那几个停用词表的四合一版本，中文停用词表，哈工大停用词表，四川大学机器学习智能实验室停用词表，百度停用词表的四合一版本。

然后是可视化分析，首先我们先看下分词后的特征有哪些，以及标签有哪些，频率是怎么样的。

这个是特征的。

这个是标签的。

字可能不是很大，不知道大家有没有看清，可以看一下我的这个jupyter里的，我是输出了频率排前60的这些特征还有标签的。

然后我在这些出现率比较高的这些特征和标签里，选了几个做了下词云进行了进一步的分析。

首先看这两个词云。左边是特征含有“少女”的标签词云，右边是标签含有“校园”的特征词云。可以看到少女里校园非常多，校园里少女非常多，二者有很强的相关性。基本上看到少女或者美少女我们就可以猜到它的标签有校园。不难想象如果单独预测是否有校园，那他的假说函数里，少女和美少女这样的特征一定会有一个高的权重。因为他们的频率很高，换言之这就是我们在训练模型时需要利用的，也就是这些特征词的频率。基本上现在我们已经有了下一步处理的思路，但我还是继续做了点词云进行分析观察，并有了一定的额外收获。

接下来是特征含有“异 世界”的标签词云和标签含有“异世界”的特征词云。你可能会好奇我特征的异世界为什么中间有个空格，因为这是结巴分词的问题，他把异和世界拆开了。标签那个异世界没空格那当然了啊，这些标签一个个爬的，自己带空格，我没必要给他分词。从这两个词云中，你可以看到一个非常古怪的现象，特征异世界，标签词云里最多的居然不是异世界而是冒险，这都一样的词居然还不是最多，这用我们人的正常思维来看都知道这肯定不正常。哪怕你去异世界都是去冒险，那也是异世界这个标签的数量大于等于冒险这个标签的数量。为什么会这样，先说明这不是我做的有问题哈，我代码没有写错，实际上是他们网站打标签打的有问题，这个我们之后还要再谈到的。

这个是标签含有“励志” 的特征词云。我想说的是你看这个久美、宇治、还有丽奈，这都是人名对吧，他们的出现频率相当的高，这意味着励志的小说喜欢用这样的人名。实际上人名背后往往可能潜藏着大量的信息，并不是毫无意义。就像我说李华，你可能就会想到英语作文。人名确实是有潜在价值的，这是第一个额外收获。

然后是我们将文本转换为频率和编码。对于特征，我们用TF-IDF将其转换为频率。对于标签，我们用MultiLabelBinarizer热编码。看下面的展示了训练集和测试集的维度。特征词按空格拆分成了26175个特征转换为的是频率。标签被拆分成总共173个，它的值就是有标签就是1，没有就是0，用这样一个行向量来表示这条数据都有哪些标签。可以看到这个数据量真的挺大的，测试集这个特征矩阵维度1469乘26175，训练起来的话应该会很慢。其实我考虑过降维，但是我调了下sklearn的PCA我发现，它好像降维后列数必须不能超过行数，但是那样丢失的特征就太多了，我也不懂该怎么办了，所以最后就没有去降维。

然后就是训练模型和预测以及评估了，我一共选用了十个模型。分别是逻辑回归、K近邻、支持向量机、多项式朴素贝叶斯、决策树、自适应提升、引导聚集算法、极限树、梯度提升决策树、以及随机森林。然后首先是一开始的预测并不好，可以看到我这个逻辑回归准确率只有0.01。我的习惯是先看看我都预测了些什么鬼东西，再看一些评分。

这是前十条测试集内容，上面是我预测的，可以看见它几乎没有输出1的预测。其实我都不需要再看混淆矩阵了，很明显召回率出现了问题，R=TP/（TP+FN）嘛，真阳例比上真阳例+反阴例，也就是说实际为1的里面，我们预测为1的占比多少。我们几乎没有预测为1嘛。（备注：F1=2PR/(P+R)）

看下这个召回率，0.087，和我们想的一样，真的很低的，召回率低嘛，那把阈值调低多去预测1就好了，默认的是>=0.5预测为1嘛，我们给他调一调。因为我们不仅仅是希望它多去预测1，还希望尽可能准确，所以不能让查准率变得特别低，所以就是去调F1分数，我把阈值调到了0.2，基本上这附近F1分数就比较好了。可以看到F1分数从0.15涨到了0.43，预测准确率也从0.01涨到了0.048，这把我们看一下前十条预测的标签。上面是预测的，下面是真实的。这把预测得就正常多了，但是准确率还是挺低的，这是因为173个标签太多，难度太大的原因吗。

请看这个截图，这是刚才测试集前10部小说里3部小说的真实标签。我们从人类思维角度分析，你会发现爱情小说、恋爱、爱情这三个标签在意思上是非常接近的。甚至爱情小说和爱情在这里应该是完全等价的才对。因我们现在这些都是小说啊。为什么这个打的标签就是爱情，那个就是爱情小说呢，而且出现恋爱是不是也应该出现爱情呢，从人类思维角度来看，这个真实的标签是不是设置的有点不合理呢。然后我们前面还讲过一件事，情节里出现了异世界这样的字眼，然而标签却没有异世界这样一个问题。这个问题我也找到原因了。他非常多的标签把异世界和异界打混了。比方说这样一个明显的例子，题目都写着呢异世界，标签还打成了异界。实际上我们可以发现，我们预测准确率低的原因在于甲方他真实标签打的就不符合一个正常的逻辑，已经不是漏打标签的事了，很多打错的，设置不合理的。所以说这个准确率低不怪我们。我们从一个正常人的思维角度来看，这个真实的标签就是有问题的。但是它是标签，我们不能随心所欲随便乱改，哪怕它是错的，它也是我们预测的一项任务。如果要更改，我们需要向甲方提出这样一个问题并商议。所以这里我就接着这个标签去做了，没有做任何调整。

然后就到了最后的阶段了，用同样的方式，我一共调用了10个机器学习集成学习模型，预测的准确率和F1分数已经统计到了dataframe，并且分别按这两个标准进行了降序排序。可以看到SVM在两个评估标准中都拿下了冠军。前三名非常稳定，SVM，逻辑回归，和随机森林。这里我给出一些分析和应用的建议，由于时间有限，这里我不全说，简单挑几个比较一下。首先是SVM，为什么拿下冠军呢。首先呢，相比我们这个特征数，可以说我们的样本数是非常非常小的。SVM是一种小样本学习方法，它基本上不涉及概率测度以及大数定律等。它避开了从归纳到演绎的传统过程,简化了通常的分类和回归等问题。其次是这个维度上，我们的这个维度特别的高，SVM本身这个核函数就是把低纬度的数字映射到高维度，我觉得SVM本身就是比较善于处理高纬度的非线性映射问题。然后SVM 的最终决策函数只由少数的支持向量所确定,计算的复杂性取决于支持向量的数目,而不是样本空间的维数,这在某种意义上也避免了“维数灾难”。然后说说核函数，因为我嫌它慢，所以我只用了线性核函数，吴恩达在课程里说过，特征维度相比数据规模很大，这时候应该选用线性核函数，我的这个数据集正好就是符合他这一点，不过吴恩达没解释原因，我感觉应该就是计算速度的问题吧。抛开计算速度的问题不谈，我个人认为鲁棒径向基核函数也就是高斯核函数应该会表现得更好，因为它的参数决定了函数的作用范围，超出了这个范围，数据的作用就基本消失了，也就是说它对数据中的噪音的抗干扰能力是非常强的，其实我们这个数据的噪音是非常多的，前面可视化分析的时候我们也发现了，实际上一个标签有没有基本上是取决于这个标签对应的特征里高频的那少数几十个词，其他大部分词实际上是没有什么用的，但是他们会产生一些扰动产生这样一些噪音。然后再说说逻辑回归，这个方法其实好处就是模型清晰好理解，实施简单，计算量小，存储占用低，一般来讲逻辑回归的准确率不是很高，这次他的这个评分挺高的也是我没有想到的。一般来讲我建议我们拿到一个问题，先直接拿逻辑回归预测一下试试。然后再去做别的。然后再说随机森林，随机森林其实是个很不错的分类器，我使用它的体验就是，每次不管做什么问题，随机森林就算不是第一，也总是能给出一个很不错的结果。原因应该有以下几点，随机森林本身就是集成算法，一般来讲精度确实是会比大部分单个算法要好的。然后因为随机森林是树的组合，它可以很好地处理非线性数据，实际上它就是一个非线性分类模型。然后还有就是，我们这次的数据特征维度非常高，而随机森林不用做特征选择，对数据集适应能力很强，特别适合处理高纬度的数据。我想随机森林在这次评估中发挥的比较好应该就是以上原因。另外随机森林因为引入了样本随机和特征随机，不容易陷入过拟合，所以其实随机森林的表现十分的问题，我个人的体验也是这样，就算随机森林拿不下第一，它的评分也会稳定的高。所以说我的建议就是，当你不知道该用什么方法的时候，优先尝试一下。然后这次表现比较差的就是这个决策树和自适应提升了。决策树呢，本身就很容易过拟合，我这个数据维度这么大，分支就会很多，所以应该就是严重过拟合，泛化性差导致这次它表现得不好吧。Adaboost表现不好的原因我觉得是，这里173个标签很多对吧，我外面套上的一层OneVsRestClassifier，173个标签嘛每次单看一个标签和其他比，这个标签真实的1肯定占比不会大的，所以也就是说可以看成数据不平衡。而adaboost在面对数据不平衡问题时精度会很差，我觉得可能就是这个原因导致这次它表现得不怎么好。

到这里还没有结束，我们还有最后一件事情，我们需要给一个情节的字符串，就能做出一组标签的预测，所以最后我选用了评估最好的SVM分类器，然后把我们对这个新的字符串用我们训练好的各种模型进行的一系列处理封装成了函数。分词，删停用词，用训练好的TF-IDF模型转换为词频，然后用训练好的SVM分类器按阈值0.22进行预测，最后返回这样一组标签的结果。这段文本时我在另一个小说网找到的，中国作家写的类似同人文的东西。可以看到我们这个预测其实还不错的。我们最初提出的问题是小说网漏打标签现象严重，其实过程中我们也发现了，错打标签和设置不合理也很严重。实际上有时候人工打标签真的还不如直接用机器来预测标签。我很推荐这些小说网啊电影网啊来使用机器预测标签。当然我这个其实还有很多可以优化的地方，一个是时间原因没有好好调参，本来我就不是很擅长调超参数，总是依赖网格搜索，但实在太慢了，然后就放弃了，没有好好调参。另一个就是，用深度学习或许最后预测得会更好，这个我并没有去尝试。再就是我们不可能交付一个函数，所以最后可以再套个前端，比方说做个微信小程序之类的。我认为自然语言处理以及文本预测在未来会有非常好的发展前景，我非常期待这个行业未来会如何发展。通过这个大作业我也发现了自己诸多的不足之处，在未来的学习中我也会不断努力，去弥补这些不足。我的汇报就到此结束，谢谢大家。