朴素贝叶斯分类实验

田田 经12计18 2021011048

一、朴素贝叶斯分类器实现及评估

1、实验设计

首先，我们需要将每一封邮件的全部内容转化成可以用来处理的特征，这个过程中我们需要选取将哪些信息使用什么样的模型转化为特征；然后，我们使用提取的特征和分类结果训练朴素贝叶斯分类器模型；最后，我们使用自定义的评估指标对模型的性能进行评估。

2、特征转换

(1) 选取邮件内容

需要我们处理的邮件都是英文的，分为邮件相关信息和邮件内容两个部分。

邮件相关信息中，包含了发信时间，发件人的 IP 地址，邮件的 ID ，发件人和收件人的邮箱地址，邮件的主题，邮件的优先级，邮件的客户端信息和邮件的内容类型与编码方式。手动观察发现并不是所有邮件都包含所有的邮件相关信息，为了具体确定在数据集中包含哪些邮件头部，使用 python 自带的库 email 遍历所有的邮件的头部，整理得总共包含782种邮件头，结合个人判断和课堂中的讨论，最终选择以下的邮件头信息用于转化特征：

```
From：发件人邮箱，可能可以通过后缀来判断是不是批量注册的垃圾邮件，如果是edu后缀说明应该不是垃圾邮件
Cc：抄送，认为如果出现了这个说明不是垃圾邮件
Bcc：密送，和抄送同理
Subject：主题，认为可以通过主题内容判断是不是垃圾邮件
Date：大半夜发送的邮件可能是垃圾邮件的概率会更大
Content-Type：正文内容类型和编码方式，认为HTML编码的是垃圾邮件的概率会更大
Content-Length：正文长度，觉得垃圾邮件可能会比较长
Content-Transfer-Encoding：定义如何对邮件进行编码
X-Authenticated, X-Authentication, X-Auth：是否通过认证，认为被认证过的不太可能是垃圾邮件
X-Priority, X-MSMail-Priority：邮件的优先级，认为被标注了优先的可能一般不是垃圾邮件
```

邮件内容由纯文本信息和 HTML 信息两部分组成，邮件可能两个部分都包含，也可能只包含一个部分。训练时将邮件内容全部输入用于转化特征。

(2) 选取转化模型

综合考虑实现的简便程度和模型的效果，我们使用词袋模型来进行特征的转化。我们将在问题三中对比讨论使用不同的特征转化方式对最终性能的影响。

结合课堂讨论的结果和我自己的思考，我认为邮件头的信息相对于邮件正文来说更加特别：邮件头的信息会更短，同时可能有更强的代表性。例如，如果邮件头中表示，这封邮件是被抄送或者被密送的，那么这封邮件大概率不是垃圾邮件，所以我们不应该把邮件头的信息和邮件正文的信息放在一起进行词频统计。并且，邮件头中的许多内容可能本身并不重要，重要的是这个内容是否存在：例如这封邮件是否被认证过的那三个指标，如果没有被认证的话无法说明什么，但是如果被认证了就可以说明大概率不是垃圾邮件。因此，有必要对邮件头的内容单独做特征的提取处理。

综合我人脑的分析结果，并结合了我了解的特征提取技术之后，我目前计划的特征提取方式如下。

对于邮件的正文，我们选择词袋模型，通过出现的频率来进行特征选择，用卡方检验选择最终被统计词频的特征数量。

卡方检验的原理是，衡量这个单词的出现频率和邮件类别的相关性，选择相关性比较高的单词作为特征。选择卡方检验的原因是，我原本计划将所有的单词的频率都纳入统计，但那样的话数据量过大，超过了我的电脑内存能够处理的上限，并且显然并不是所有的单词的出现频率都重要，绝大多数的单词可能只出现了几次并且无关紧要，因此我计划只将一部分单词的出现频率转化为最终的特征。

但问题在于，我们并不能简单根据单词的出现频率高低来选择是否将其纳入特征选择之中。例如，可能只有一小部分垃圾邮件是推销产品的，因此其中出现的和推销相关的单词可能出现频率很低，但是这些单词可以作为判断是否是垃圾邮件的重要判断依据。所以，我选择使用卡方检验来选择单词，通过出现频率和邮件分类的相关性来选择特征。在这里，卡方检验选取多少个单词统计频率是一个超参。

对于邮件头而言，由于邮件头的内容和正文相比通常较短，但邮件头的内容又非常重要。因此我们针对邮件头单独进行特征提取，不和邮件正文一起进行特征的选择。

首先，我们对于邮件头的信息单独做词频统计，使用卡方检验统计词频，这样可以让邮件头的信息有专门对应的特征向量，不会被邮件正文中出现频率更高或者相关性更高的单词挤掉。

然后，我们也对邮件头的信息做二值化，记录信息是存在还是缺失，为了使用之前提及的给根据邮件头中是否存在某些信息（抄送，密送等）来辅助判断是否是垃圾邮件。

考虑到邮件主题的内容可能比较重要，或许可以针对邮件主题和发件人的信息单独做词频统计和分析，以把类似 edu 等教育邮箱的后缀给区分出来。如果原本的模型性能不佳的话，可以使用这一分类方式。

3、训练过程

我选择的是朴素多项式贝叶斯网络实现。

刚开始训练的时候，我为邮件头选择的特征数是 300，为邮件内容选取的特征数是 2000，过拟合非常严重。在测试集上 accuracy 的值是0.9005，precision 的值是0.9842，整体来看表现比较好；但是在测试集上 accuracy 的结果只有0.35，甚至不如随机猜测的结果。降低特征词数量的参数可以提升，但是也无法让正确率过半。

通过在网络上搜索别人处理垃圾邮件分类问题的经验，我意识到我需要在训练前进行文本清洗，去除 HTML 格式的内容、多余空白字符和停用词，考虑到包含网址链接的邮件更有可能是垃圾邮件，但是网址中的内容可能并没有实际的含义，将所有的网址替换成 http。

随后调试参数体验为：增加用来计算特征的单词数能够提升训练集上的表现，但是会降低测试集的性能；降低用来计算特征的单词数会使得训练集的表现降低，但会使得测试集的性能和训练集的性能接近。也就是说，存在着过拟合的问题。因此尝试修改模型的结构，在原本进行课件中的贝叶斯计算的基础上加入拉普拉斯平滑操作，避免问题三中提及的零概率问题，尝试在整体单词数增加的情况下降低过拟合的问题。

并且，在调试参数的时候发现，可以通过直接查看选取的特征词的方式来检查是否模型在按照我设想的方式运行，因此引入这一方式来辅助调整模型。通过这一方式发现，如果不使用卡方检验，直接通过词袋模型来选取单词，选出来的特征词都不是正常的单词，感觉像是来源于一篇很长的乱码邮件，性能也较为糟糕。以及发现2022，111等数字经常出现在特征词中，但我没有发现在训练数据中这些数字和是否是垃圾邮件有什么关系，推测可能是直接通过了年份日期等信息判断，认为这并不是好的特征，于是在数据预处理的时候也把文本中的数字去除掉。

我最终选择使用的参数是：选取21个邮件头中的特征词，40个邮件文本中的特征词，将拉普拉斯平滑操作的 alpha 设定为100。具体的性能和表现将在下一部分中分析。

由于采用了五折交叉验证，因此对于每一折都会单独选取一次特征词，在报告中为了不让篇幅过长，我会合并在五次训练中选取的所有特征词。如果希望查看具体每一次训练的结果，可以在 `hw1.ipynb` 中查看具体的运行结果。

根据首字母顺序排序的模型选取的邮件头中的特征词为：

```
['_dragon', 'alternative', 'ascii', 'boundary', 'charset', 'dmdx', 'edt', 'est', 'flowed', 'format',
'forster', 'handyboard', 'html', 'iso', 'jonathan', 'jp', 'multipart', 'normal', 'paper', 'plain', 'text']
```

根据首字母顺序排序的模型选取的邮件文本中的特征词为：

```
['arizona', 'ascii', 'board', 'code', 'com', 'content', 'crust', 'data', 'date', 'dec', 'digest', 'dmdx',
'edu', 'esmtp', 'file', 'files', 'gt', 'id', 'kb', 'know', 'list', 'lt', 'mail', 'message', 'network', 'nil',
'owner', 'padding', 'pnfs', 'price', 'problem', 'product_table', 'psy', 'psych', 'px', 'received', 'reply',
'rpcss', 'send', 'sender', 'set', 'subject', 'telecom', 'thanks', 'thu', 'type', 'ucsb', 'university', 'use',
'using', 've', 'version', 'vulnerable', 'wed', 'wrote']
```

4、性能评估

从定性的角度分析，我认为垃圾邮件分类任务的特别之处在于：漏网之鱼是可以被容忍的，但将正常的邮件分类为垃圾邮件会有着比较严重的后果。

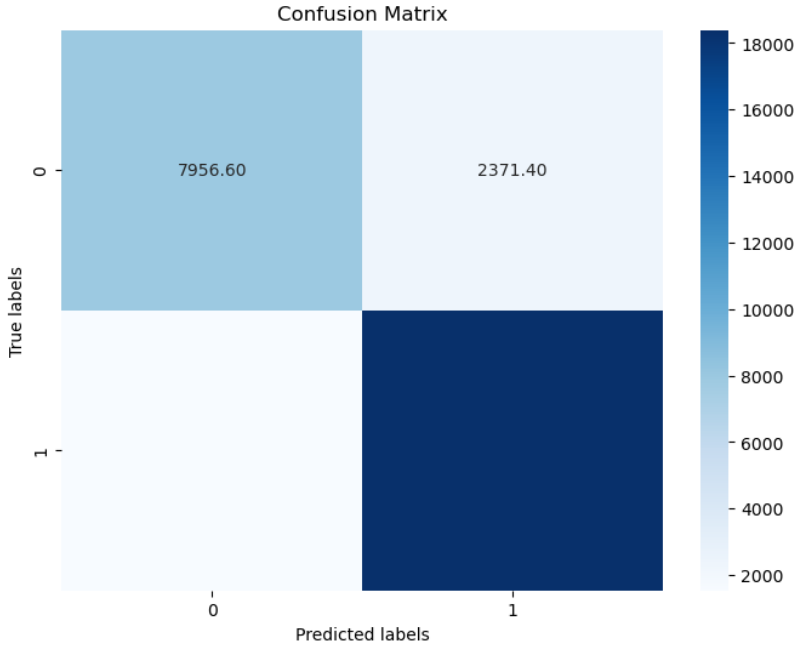
因此，我们使用准确率来衡量模型整体分类的成功率，使用准确率、精确率、召回率和 F1Score 来专门判断模型对垃圾邮件的整体判断能力，绘制混淆矩阵对应的热力图来整体观察分类器在分类结果上的表现，由于在训练数据集中垃圾邮件是非垃圾邮件的一倍，正例和反例数量并不接近，我们使用 PR 曲线关注分类器在召回率和精确率中的权衡，使用 Kappa 系数衡量分类的一致性。

由于使用的是五折交叉验证，下文中提及的所有指标都是分别在每折上计算出的结果的平均值。

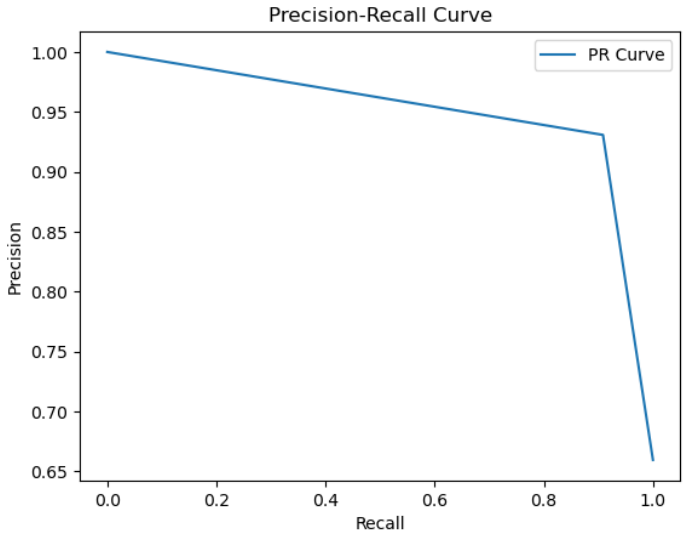
（1）训练集上的表现

在训练集上，模型拥有相对比较好的表现。准确率为 0.871，精确率为 0.887，召回率为0.923，F1Score 为0.904，有着比较好的判断能力和精确率与召回率之间的平衡。

由于五折交叉验证的结果是五折的平均，因此混淆矩阵中的四类数量都不是整数，这并不是由于计算错误导致的。整体来看错误主要集中在将正常邮件分类为垃圾邮件，这很不符合我们希望的结果。



PR 曲线主要关注不同召回率下的精确率表现，分析模型在精确率和召回率之间的平衡。PR 曲线下的面积越大，说明模型在垃圾邮件识别方面的性能越好。整体来看 PR 曲线下的面积比较大，在训练集上的识别能力比较好。

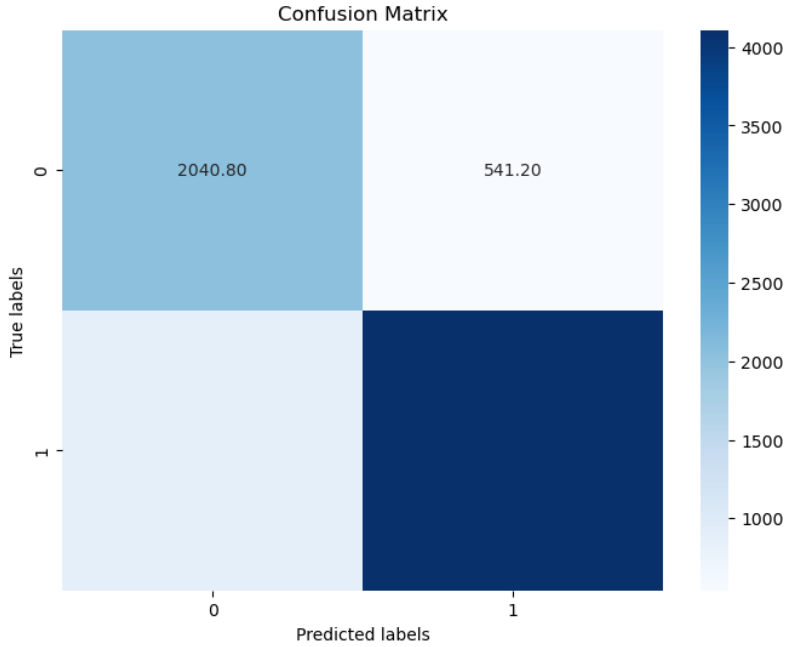


Kappa 统计量主要衡量分类预测的准确率和随机预测的差异，用来评价分类能力的一致性程度，一般认为大于0.7时模型的分类能力较好。在训练集上，Kappa 统计量为0.7065，说明模型在训练集上的一致性表现较好。

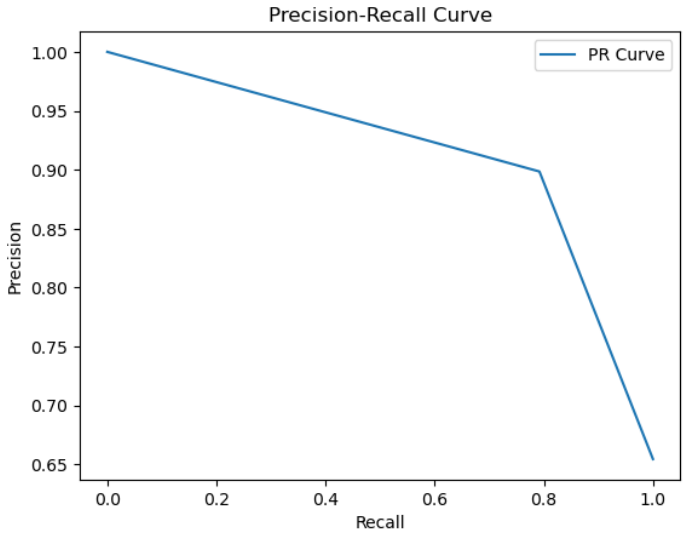
（2）测试集上的表现

在测试集上，模型在准确率上和训练集有较大差距。准确率为 0.813，精确率为 0.885，召回率为0.8246，F1Score 为0.852，有着比较好的判断能力和精确率与召回率之间的平衡。

通过混淆矩阵整体来看，错误主要集中在将垃圾邮件分类为正常邮件，这是我们能够接受的错误类型，表现优于训练集的情况。



整体来看， PR 曲线下的面积也比较大，在训练集上的识别能力同样比较好。



在测试集上，Kappa 统计量为0.5965，在一致性上的表现相比训练集有着明显的下滑。

整体来看，尽管已经为了避免过拟合降低了模型的大小，也为了避免问题三中提及的零概率事件引入了拉普拉斯平滑操作，依然存在着过拟合的问题，模型最终的表现也并不理想。

5、结果分析

（1）回答要求的问题

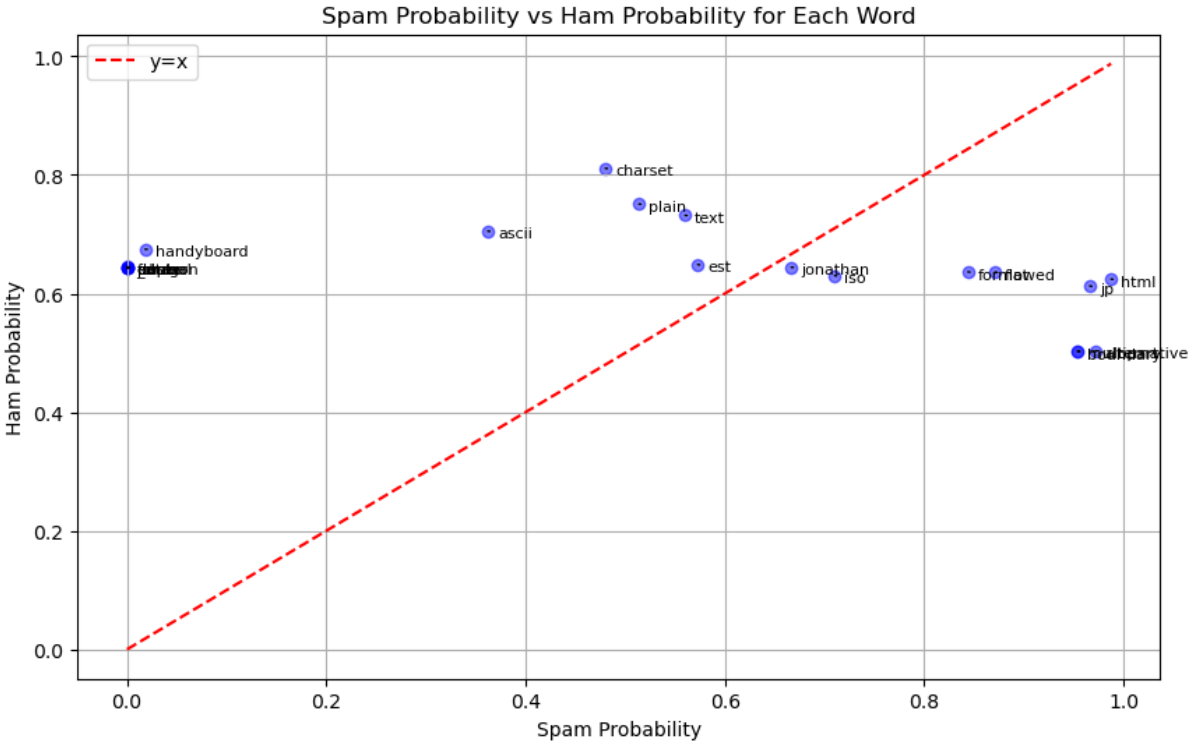
- **实验中遇到或者探究了什么问题?**
  - 本次实验中遇到的问题主要集中在训练过程这一部分，集中表现为模型的性能表现不好
  - 而具体地说，导致模型性能表现不好的原因有很多，我能够发现的有这些：
    - 1、邮件中可能含有比较长的 HTML 格式文本，会干扰词袋模型的选词
    - 2、如果只按照词频选词的话，选取出来的很多词没有含义，可能是一篇很长的邮件中反复多次出现的内容
    - 3、使用卡方检验后词袋模型选取出来的词经常是数字串，感觉没有什么含义
    - 4、在选取的特征数很多的情况下，有非常严重的过拟合问题
- **遇到问题后如何设计进一步的实验**
  - 遇到问题之后，首先是去分析导致问题的原因，通过仔细阅读代码梳理实验逻辑，上网查看别人解决相似问题的经验来找出问题所在，然后尝试引进新的技术或者修改参数来提升模型的性能

- **如何针对问题调整算法**
  - 首先要先了解导致这一问题的根本原因是什么，需要能够从最底层的逻辑上理解问题的成因
  - 然后需要了解有哪些相关的技术可以解决这一问题，可以上网查看他人的经验
  - 针对具体的问题，我对算法的调整如下：
    - 1、在进行特征转化之前进行文本清洗，去除 HTML 格式的内容和其他信息（具体操作在训练过程部分中有详细说明）
    - 2、引入卡方检验，选取是否出现和是否是垃圾邮件相关性高的词
    - 3、在文本清洗中去除数字
    - 4、引入拉普拉斯平滑技术避免零概率问题，反复调参尝试降低模型的大小，具体查看选取的特征词来判断是否选取了符合逻辑有效的特征词，但这个问题并没有被根本解决掉
- **调整是否提升了模型的效果**
  - 整体而言，调整是成功提升了模型的效果的
  - 对于过拟合之外的问题，我觉得我的调整手段都是比较有效的
  - 但我并没能成功解决掉过拟合的问题，只是勉强减弱了过拟合的影响

（2）进一步分析结果

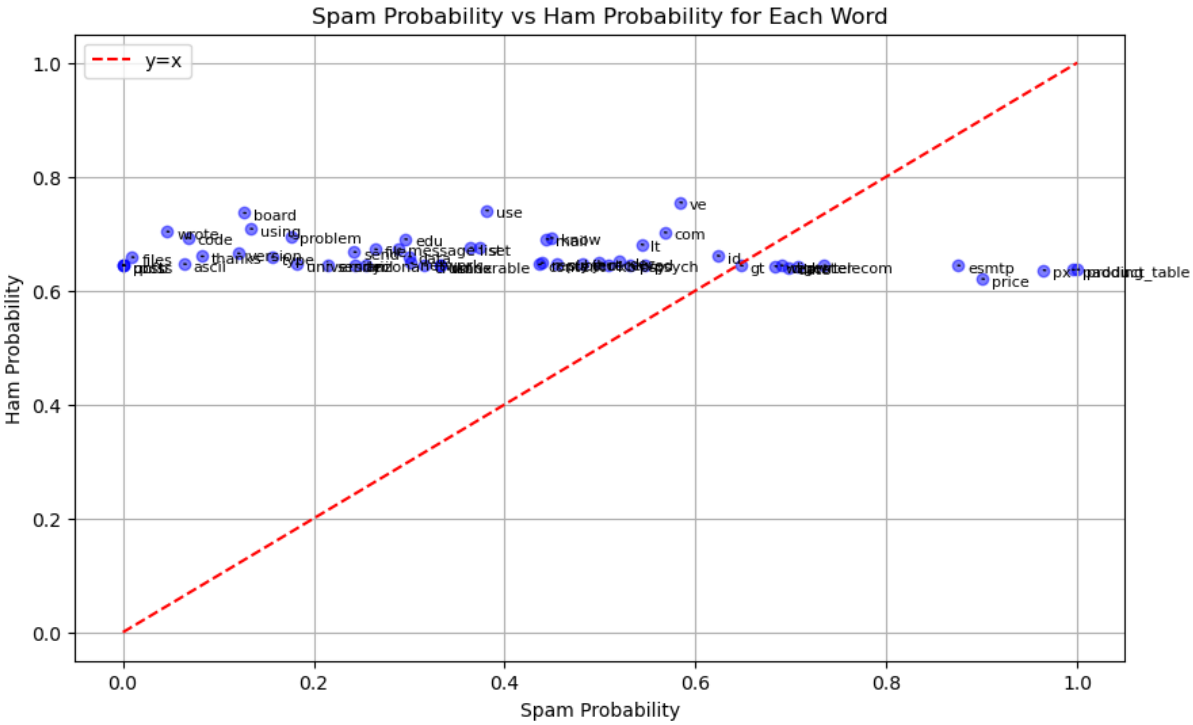
感觉提取出来的关键词看起来很奇怪，很多单词我无法联想到为什么会和是否是垃圾邮件存在相关联性，怀疑可能是因为某些错误被纳入到最终内容中的。因此，这一部分我希望具体看看特征词和是否是垃圾邮件之间的关系，主要从训练数据中入手。

针对邮件头中提取的特征词，使用全部的数据，我绘制了散点图。横坐标是分别对于这些提取出来的特征词，包含它的邮件是垃圾邮件的频率；纵坐标是不包含特征词的邮件是垃圾邮件的概率。



从图中可以看出，整体而言，大部分被选出来的特征词还是和 $y = x$这条线有着比较远的距离，即有着比较明显的不相干性。但也可以发现有许多被选取出来的词，比如`est`，既没有实际的含义也没有明显的和邮件是否是垃圾邮件的相关性，不明白为什么被算法选取了出来。

对邮件正文中的特征词做相同的操作，得到类似的图像。



分析的结果是类似的，同样存在许多和这条线非常接近的特征词，不理解为什么这些词能够被选取出来。

不过，我们或许可以基于此转变思路，从目前被选取出来的词中，再次选取和 $y = x$ 直线距离较远作为提取特征的单词，然后重新训练模型，观察性能是否提升。

根据这一思路，我们重新训练模型。对于每一次划分，先按照原先的方式和参数提取特征词，然后针对提取得到的特征词，计算他们和 $y = x$ 直线的距离，选取距离较远的点作为最终的特征词用于提取特征向量训练模型。

这样设计的算法中，除了原先的三个超参之外，新增加了如何在第一次提取得到的特征词中重新选择的超参。我最终调试后采用的方式是：选择离 $y = x$ 直线最远的五分之三的点，并舍弃掉最远的五个点（防止问题二中提及的零概率事件）。

最终在训练集上取得的结果和原先的模型取得的结果相似：

	准确率	精确率	召回率	F1Score	Kappa
新模型	0.8203	0.8403	0.9031	0.8692	0.5832
原先模型	0.8129	0.8848	0.8246	0.8529	0.5965

由于我在最后一天晚上突然想到的这个想法，因此没有太多时间去具体调试优化具体实现，只能匆忙简单实现基本的思路。但我认为通过这个方式构建的模型如果同样精心调试参数的话，一定能取得比原先的模型更好的结果。

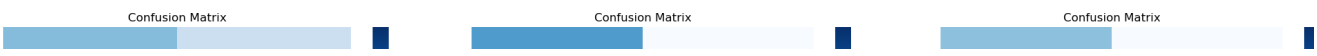
二、问题一的回答

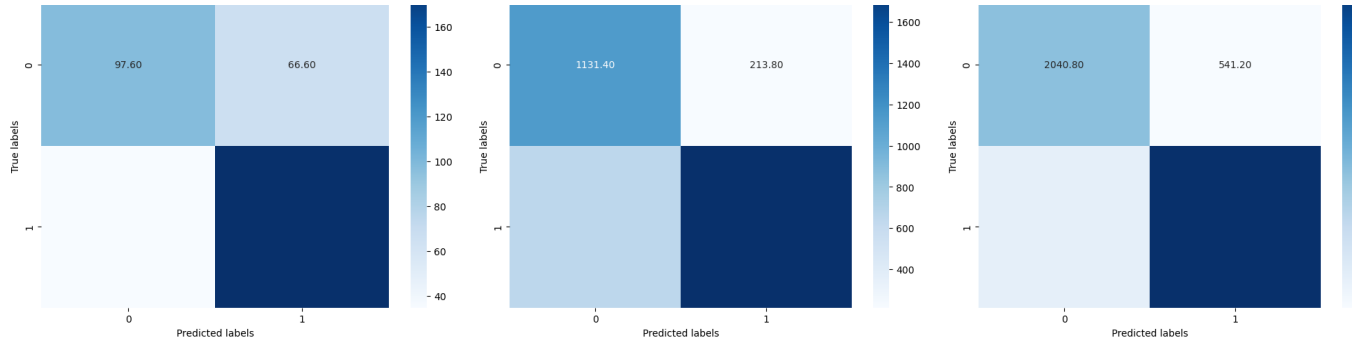
在这里我们研究训练集的大小对于分类器性能的影响。在具体的操作上，我们采纳建议，分别采样5%，50%和100%的训练集数据进行训练，然后观察在测试集上的表现。

在这里我对于训练集大小的理解是"整体数据量"的多少，因此我的操作是：分别取了整体数据的5%和50%，采用和使用全部数据相同的流程和评价指标训练检验分析，然后对比最终在测试集上的表现结果。

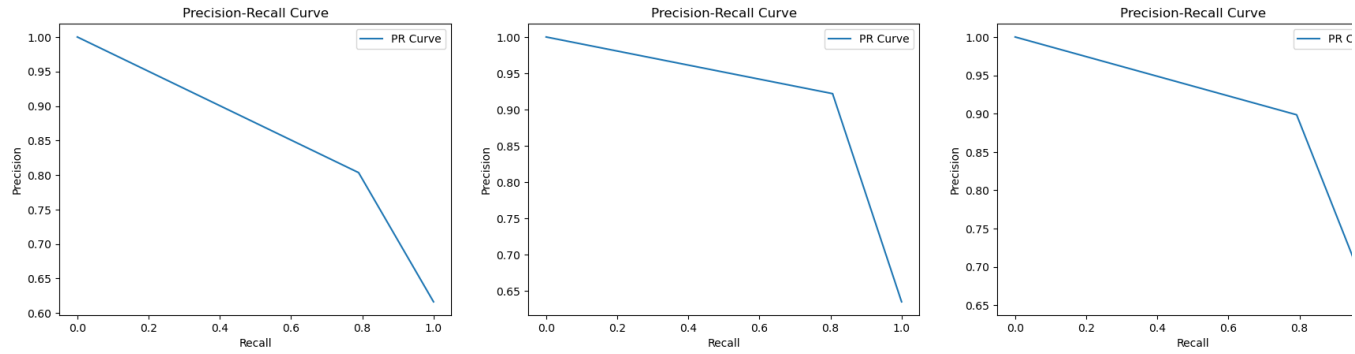
	准确率	精确率	召回率	F1Score	Kappa
5%的数据	0.7319	0.7301	0.8354	0.7773	0.4373
50%的数据	0.7712	0.8970	0.7330	0.8051	0.5335
100%的数据	0.8129	0.8848	0.8246	0.8529	0.5965

从前四个指标可以看出，随着整体训练数据的增加，模型的判断能力也是基本正相关提升的。在精确率和召回率上，训练数据量的提升并不是严格和判断能力正相关，但整体趋势也基本符合。Kappa 统计量也是稳定提升，说明随着数据量的增加，模型判断的一致性同样在稳定提升。





从混淆矩阵对应的热力图可以看出，随着数据量的增加，整体的判断能力也在提升，右上角处把正常邮件误分类为垃圾邮件的热力区域整体在变浅，符合我们希望的变化趋势。



从 PR 曲线也可以看出，随着数据量的增加，PR 曲线下的面积逐渐增加，整体的识别能力在逐渐提升。

整体而言，我们可以认为随着数据量的增加，模型在各方面的表现都会提升，这也符合我们基本的认知，即掌握的信息越多训练量越大判断能力就越强。不过这次对比的问题在于，如果只是想要比较数据量的影响的话，我们应该针对不同的数据量分别调试模型参数，让模型达到在不同数据量下的相对最优结果，但限于时间本次作业中没有进行如下操作。

	训练集准确率	训练集精确率	测试集准确率	测试集精确率
5%的数据	0.7319	0.7301	0.8666	0.8349
50%的数据	0.7712	0.8970	0.8671	0.9629
100%的数据	0.8129	0.8848	0.8708	0.8865

从这次对比中我们同样可以发现，数据量较少的模型相对于数据量较大的模型出现了更加严重的过拟合现象，这也符合我们的认知。另一个相对较为合理的解释是：由于并没有针对不同数据量的模型专门调参，而是使用了数据量最大的模型调试出来的表现相对较好的参数，从而导致对于数据量较少的模型，选用的特征词过多，从而进一步放大了过拟合。

三、问题二的回答

假如训练集中没有样本符合 $x_i = k, y = c$，说明对于这一特征词，它在 c 这一类样本中出现的频率是零。那么根据朴素贝叶斯的概率计算公式，任何包含这一特征词的样本都不可能被分类成这一类别。但是频率并不能代表概率，并且我们的训练集也并不能完全代表整体，如果只是因为训练集中不包含这一特征词，我们就认为这一单词不可能存在，那么会让我们的模型处理新数据的能力大大下降。

例如，假如"假期"这一单词在我们用来训练的数据范围中，从来没有在非垃圾邮件中出现，我们的朴素贝叶斯模型就会认为所有包含"假期"这一单词的邮件都不可能是非垃圾邮件，也就是只能是垃圾邮件。但是实际情况想必并不是这样，只是训练数据中并不包含这一单词，不代表这一单词真的和是不是垃圾邮件有什么很强的相关性。

我们可以通过平滑技术来解决这个问题，就像说明中给出的公式一样，在概率计算中在分子分母上都加一个很小的值。这样，如果训练中再次出现这个问题，我们就会让它的概率是一个比较小的值而不是零，从一票否决变为较为确定，增加我们模型的健壮性。

四、问题三的回答

除了词袋模型处理邮件正文的特征提取方式之外，我们还可以直接提取邮件头的信息，就像在特征转换那一部分中具体描述的那样。

我们也可以使用词袋模型之外的模型提取特征，例如使用更加复杂的 TF-IDF 模型，它不光关注特定此在这封邮件中的出现频率，还会关注这一单词在所有邮件中的出现频率，然后将这两个数值相乘得到最终用来比较计算的频率。或者使用更加复杂的基于神经网络的特征提取方式。

我们也可以不使用如此“自动化”的方式，也可以像结果分析中那样，在使用自动的分类提取筛选之后，手动选取比较符合逻辑并且显示出比较好的相关性的单词，指定只使用它们来选取特征。