运用朴素贝叶斯法预测
能否成功申请美研统计学专业

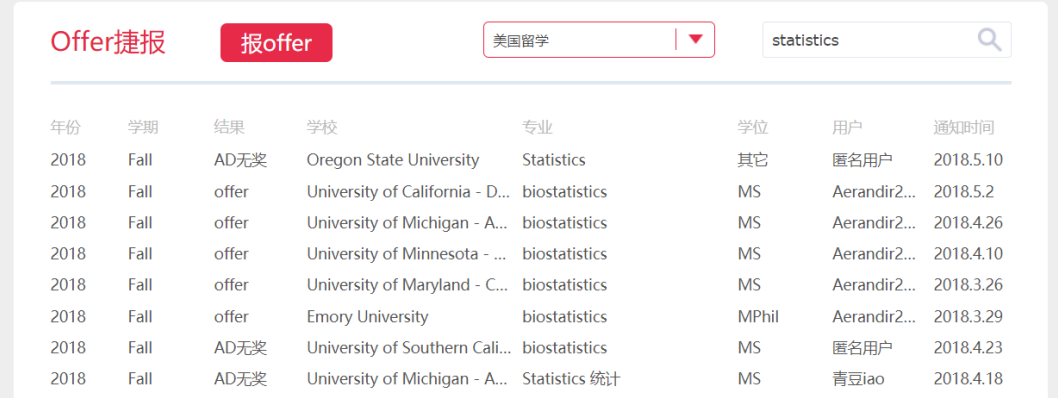
郑昊亮
2016310868

2018 年 7 月 17 日

1　数据

1.1　数据的获取

我想研究的是美国研究生统计学专业的申请与录取情况，为了获得相应的数据，我选择了留学论坛“**寄托天下**”。其中有一个板块为offer榜，网址为http://www.gter.net/offer ，网页如下所示：



年份	学期	结果	学校	专业	学位	用户	通知时间
2018	Fall	AD无奖	Oregon State University	Statistics	其它	匿名用户	2018.5.10
2018	Fall	offer	University of California - D...	biostatistics	MS	Aerandir2...	2018.5.2
2018	Fall	offer	University of Michigan - A...	biostatistics	MS	Aerandir2...	2018.4.26
2018	Fall	offer	University of Minnesota - ...	biostatistics	MS	Aerandir2...	2018.4.10
2018	Fall	offer	University of Maryland - C...	biostatistics	MS	Aerandir2...	2018.3.26
2018	Fall	offer	Emory University	biostatistics	MPhil	Aerandir2...	2018.3.29
2018	Fall	AD无奖	University of Southern Cali...	biostatistics	MS	匿名用户	2018.4.23
2018	Fall	AD无奖	University of Michigan - A...	Statistics 统计	MS	青豆iao	2018.4.18

以用户“青豆iao”为例，点击其所在栏（即上图中最下一行），会跳转到新的网页，网址为 http://bbs.gter.net/thread-2164306-1-1.html ，页面内容如下所示：

经过点击查看其他用户的网页，我发现 Offer 榜网页的规律为：

1、网址的形式均为“http://bbs.gter.net/thread-”＋“一串数字序列”＋“-1-1.html”，因此只需记录网址对应的数字序列，即可得到该网站。

2、由于 Offer 榜的发布是通过在“报Offer”网页中填表形成的，因此每个网页大体上都有如上图所示的形式；首先是Offer情况，如果有多个，则从上往下依次排列；然后是个人情况。Offer情况均含有“申请学校”“学位”“专业”“申请结果”这些我所需要的信息，个人

情况中一般也都含有“TOEFL”“GRE”“本科学校档次”“本科成绩和算法、排名”这些我所需要的信息。

进一步，通过查看网页源码（如下图所示），我发现：

1、我所需要的信息都在属性class=“typeoption”的div节点下。

2、该div节点的第一个table节点的summary属性的值“offer n”中的 n 实际上指的是offer栏目数加个人信息栏目数，也就是申请情况数+1。

3、对于申请信息，在每个table节点下的tbody节点下，每个tr节点都对应一个信息，内容在td中，且不同网页的源码格式是相同的。

4、对于个人信息，在最后一个table节点下的tbody节点下，虽然不同网页的源码格式并不完全相同，但相同内容的th元素的内容是相同的。

通过以上对网页的描述和分析，我就可以借助 R 进行爬虫，获得我所需要的数据。

由于我想研究的是申请美国学校的情况，因此我选取的网页均源于“美国留学”区，申请的时间范围是2015年——2018年。将网址对应的数字序列存储于向量中，再将该向量输入我的爬虫函数，便获得了我的原始数据，245个样本单元，9个变量。具体程序见“郑昊亮_2016310868_SCEXAM.R”。

1.2 数据的处理

在上一小节获得原始数据之后，由于存在缺省值的情况，以及如“result_raw”“apply_for_raw”等变量的取值较为复杂，为此需要对原始数据进行进一步处理，来获得更适合描述、建模的数据。

对于缺省值的问题，我直接将存在缺省值的样本单元删除。对于“result_raw”，我将“AD无奖”“AD小奖”“offer”这3个水平统一化为“Offer”水平，将“被拒” “Waiting list”这2个水平统一化为“Rejected”水平，并将新的数据存储为“result”。对于“apply_for_raw”，我根据 USNews 2018 美国大学统计学专业研究生排名，将申请学校对应的排名（“top10”等，两两不重叠）存储为“apply_for”。

经整理后剩余202个样本单元,数据说明如下：

特征	含义
index	数据所在网页的网址对应的数字序列
result_raw	从网页中爬取到的申请结果
result	整理过的申请结果，仅“Offer”“Rejected”两类
apply_for_raw	申请的美国学校的名称
apply_for	整理过的申请目标，学校的统计学专业排名所在区间
major	申请的专业（数据中全为统计类）
degree	申请的学位（数据中全为硕士学位）
gpa	申请者的加权平均分
TOEFL	申请者的托福成绩
GRE	申请者的GRE成绩
from_raw	从网页中爬取到的申请者的本科学校
from	整理过的申请者的本科学校

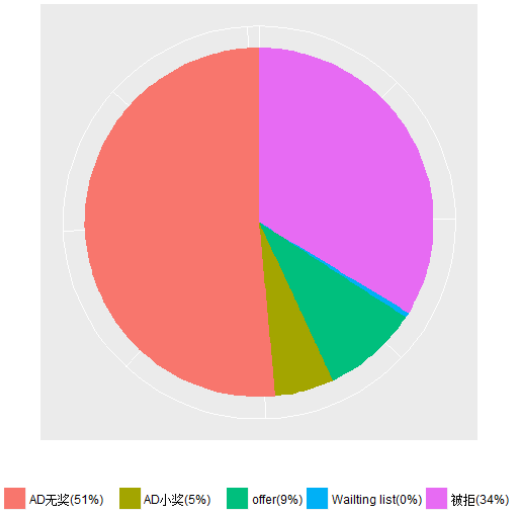
2 数据的描述与动机

2.1 描述

由于我的数据的特征较多，因此我将按照上表的顺序，从上到下对数据的重要特征依次进行描述。

1、result_raw

该特征共5个水平，分别为“AD无奖”“AD小奖”“offer”“Waiting list”“被拒”，其中“offer”指的是获得录取并拿到奖学金，而“AD”仅仅意味着录取。比例从下面的饼图可以看出，没有奖学金的占总结果的一半，Waiting list 极少。



2、result

该特征由result_raw转化而来，仅“Offer”“Rejected”2个水平，比例为Offer 占65.84%，Rejected 占34.16%,说明我的样本中获得录取的占比更大一些。

3、apply_for_raw

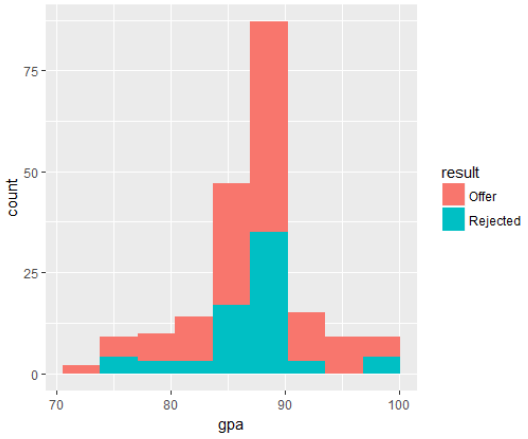
该特征共63个水平，意味着我的样本中，学生向63所不同的美国学校提交了申请，其中收到申请最多的是**哥伦比亚大学（占比10%）**，其次是密歇根大学安娜堡分校（占比6%），占比接近4%的有乔治华盛顿大学、约翰霍普金斯大学和明尼苏达大学双城分校。

4、apply_for

该特征共7个水平，分别为”top5””top10””top20”... “top 50”和“else”，其中占比最大的是 else（27.2%）和 top20（24.8%），意味着在样本中申请排名50以后的学校，以及排名20到10的学校最多。

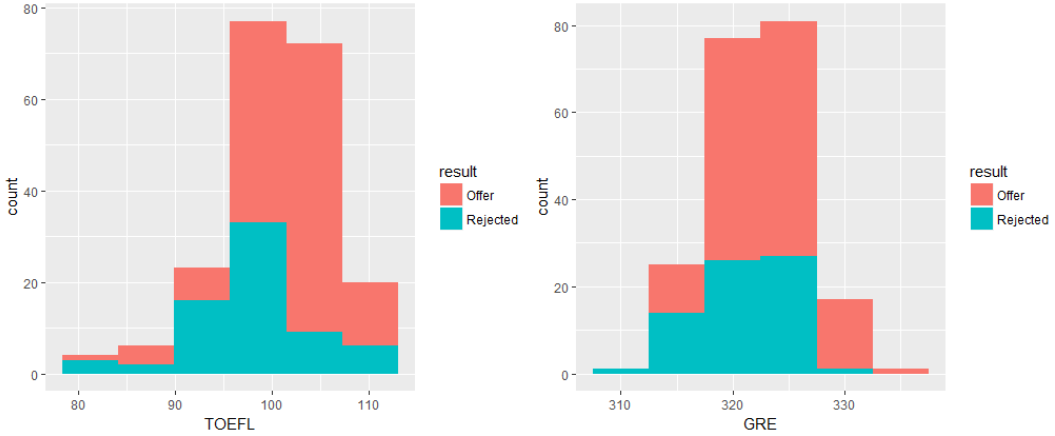
5、gpa

gpa整体的中位数和平均数均为87，通过以下直方图，可以直观地得到其按录取与否分类的分布情况



6、TOEFL

TOEFL整体的中位数和平均数均为100，而被录取者的中位数和平均数要高于被拒绝者5分，通过以下左侧直方图，也可以直观地看到，被录取者在直方图中的分布整体偏右，被拒绝者偏左

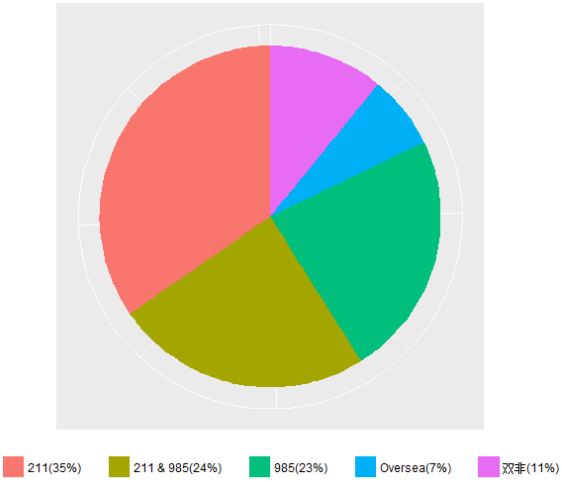


7、GRE

GRE整体的中位数和平均数均为322，被录取者的中位数和平均数与被拒绝者差别仅1分，通过以上右侧直方图，也可以直观地看到，GRE分数的分布较为集中，主要在 315-330 之间

8、from

下面的饼图直观地展示了我的样本中学申请者生的本科学校情况，绝大多数都是211及以上水平的学校



2.2 动机

通过以上的描述，联系到课堂上曾经学习过的逻辑斯蒂回归，我意识到：如果将只有两种类别的“result”视为因变量，将“apply_for”“gpa”“TOEFL”“GRE” “from”视为自变量，我就可以运用该数据研究关于“**能否成功申请美研统计学专业**”的二类分类问题。

通过查阅相关书籍（见附录　参考书），我了解到：分类是监督学习的一个核心问题，对于这类问题，有许多方法可以用于处理，包括：逻辑斯蒂回归模型、朴素贝叶斯法、决策树、支持向量机等。其中，由于朴素贝叶斯法没有在课堂上学过，符合作业要求，而且相对简单，我又有一定的贝叶斯统计知识的储备，容易实现；因此我选择**朴素贝叶斯法**作为我用于实现的模型，并在下一节中展开。对于在课堂上学过的逻辑斯蒂回归，以及较为直观的决策树模型，我将在第4节中作为对比，予以简单展现。

3 模型

3.1 模型概述

对于该二类分类问题，设输入空间为 $\mathbf{x} \subset R^n$的n维向量的集合，输出空间

为类标记集合 $\mathbf{y}=\{c_1,c_2\}$；输入为特征向量 $x\subset\mathbf{x}$，输出为类标记 $y\subset\mathbf{y}$。

朴素贝叶斯法分类时，对给定的输入 x，通过学习到的模型计算后验概率分布 $\mathrm{P}(\mathrm{Y}=c_k|\mathrm{X}=x)$，将后验概率最大的类作为 x 的类输出。后验概率计算公式根据贝叶斯定理：

$$\mathrm{P}(\mathrm{Y}=c_k|\mathrm{X}=x)=\frac{\mathrm{P}(\mathrm{X}=x|\mathrm{Y}=c_k)\mathrm{P}(\mathrm{Y}=c_k)}{\mathrm{P}(\mathrm{X}=x)} \tag{1}$$

其中，条件概率分布 $\mathrm{P}(\mathrm{X}=x|\mathrm{Y}=c_k)$ 有指数级数量的参数，其实际估计是不可行的，因此朴素贝叶斯法对条件概率分布作了条件独立性的假设，这是一个较强的假设，朴素贝叶斯法也由此得名。具体地，条件独立假设是

$$\begin{aligned}\mathrm{P}(\mathrm{X}=x|\mathrm{Y}=c_k)&=\mathrm{P}(\mathrm{X}^{(1)}=x^{(1)},\cdots,\mathrm{X}^{(n)}=x^{(n)}|\mathrm{Y}=c_k)\\&=\prod_{j=1}^{n}\mathrm{P}(\mathrm{X}^{(j)}=x^{(j)}|\mathrm{Y}=c_k)\end{aligned} \tag{2}$$

将式（2）代入式（1）有

$$\mathrm{P}(\mathrm{Y}=c_k|\mathrm{X}=x)=\frac{\mathrm{P}(\mathrm{Y}=c_k)\prod_{j=1}^{n}\mathrm{P}(\mathrm{X}^{(j)}=x^{(j)}|\mathrm{Y}=c_k)}{\mathrm{P}(\mathrm{X}=x)} \tag{3}$$

于是，朴素贝叶斯分类器可表示为

$$y=f(x)=\arg\max_{c_k}\frac{\mathrm{P}(\mathrm{Y}=c_k)\prod_{j=1}^{n}\mathrm{P}(\mathrm{X}^{(j)}=x^{(j)}|\mathrm{Y}=c_k)}{\mathrm{P}(\mathrm{X}=x)} \tag{4}$$

注意到，在式（4）中分母对所有 c_k 都是相同的，所以可以忽略分母部分。

接下来的关键就是关于先验概率$\mathrm{P}(\mathrm{Y}=c_k)$和条件概率$\mathrm{P}(\mathrm{X}^{(\mathrm{j})}=x^{(j)}|\mathrm{Y}=c_k)$的求解。对应于离散型特征和连续性特征，可以分别使用多项式模型和高斯模型进行处理:

1、多项式模型

当特征是离散的时候，使用多项式模型。多项式模型在计算先验概率和条件概率时，可以做平滑处理，以防止所要估计的概率值为0的情况出现。具体公式为

$$\mathrm{P}(\mathrm{Y}=c_k)=\frac{\sum_{i=1}^{N}I(y_i=c_k)+\lambda}{N+K\lambda} \tag{5}$$

其中，N 是总的样本个数，K 是总的类别个数（在本文考虑的情况中为 2 ），λ 是平滑值。

$$\mathrm{P}(\mathrm{X}^{(\mathrm{j})}=a_{jl}|\mathrm{Y}=c_k)=\frac{\sum_{i=1}^{N}I(x_i^{(j)}=a_{jl},y_i=c_k)+\lambda}{\sum_{i=1}^{N}I(y_i=c_k)+S_j\lambda} \tag{6}$$

其中，S_j是特征的维数，λ是平滑值。

当$\lambda = 1$时，称作Laplace平滑；当$0 < \lambda < 1$时，称作Lidstone平滑；$\lambda = 0$时不做平滑，为极大似然估计。

2、高斯模型

当特征是连续的时候，采用高斯模型。高斯模型假设每一维特征的条件分布都服从正态分布：

$$p(\mathrm{X^{(j)}} = a_{jl}|\mathrm{Y} = c_k) = \frac{1}{\sqrt{2\pi\sigma^2_{c_k,j}}}\mathrm{e}^{-\frac{(a_{jl}-\mu_{c_k,j})^2}{2\sigma^2_{c_k,j}}} \tag{7}$$

其中，$\mu_{c_k,j}$表示类别为c_k的样本中，第j维特征的均值；$\sigma^2_{c_k,j}$表示类别为c_k的样本中，第j维特征的方差。

根据以上的概述可以看出，当使用朴素贝叶斯法处理我的美研申请数据时，需要建立在特征之间相互独立，以及特征的条件分布服从正态分布之上。不得不承认的是，从直觉上讲，我的数据很难完美地满足独立性的假定，对于正态性的假定，也只是大致符合。因此会牺牲一定的分类准确率。

3.2 模型的实现与结果

1、模型的实现

根据概述中朴素贝叶斯法的思想与步骤，我的编程思路是：

创建一个函数，输入为训练数据集、待预测的数据、训练数据集中类标记所在列以及平滑值（默认为0）。首先验证输入数据的正确性。然后通过“训练数据集中类标记所在列”将输入的训练数据集分为标记和特征。对于特征，通过数据类型是因子还是数值，可以区分出离散型特征和连续型特征，并记录离散型特征和连续型特征所在列。

接下来是学习过程。首先可以计算先验概率。对于条件概率，要分离散型特征和连续型特征来考虑：对于离散型特征，可以先创建一个数组，每一层代表不同的特征，在同一层中，行代表标记的类，列代表该特征的不同的水平，交点存储的便是该特征的某一水平对应的条件概率。对于每个离散型特征，可以遍取其所有水平，一一计算概率，将数组填充。当循环结束时，便完成了一个条件概率表。对于连续型特征，首先使用apply函数计算均值和标准差，然后创建一个函数，其输入是类的取值以及特征的取值，输出则是对应的正态概率密度。

然后便是分类过程。由于考虑的问题只有两类，所以需要计算的便是两个后验概率，然后比大小，将后验概率大的类作为输出。计算后验概率，实际上是三个概率的乘积，分别是先验概率，离散型特征的条件概率及连续型特征的条件概率。先验概率已经求出；离散型特征的条件概率只需将输入的离散型特征与在学习过程中创建的表进行对照，选取对应的概率即可；连续型特征的条件概率只需将类与特征的取值输入在学习过程中创建的函数即可。

通过以上的过程，将其转换为代码，即可实现朴素贝叶斯法。具体程序见“郑昊亮_2016310868_SCEXAM.R”。

2、模型的结果

将我的数据随机取70%作为训练集，30%作为测试集，并将训练集、测试集中的特征数据以及类所在列（在我的数据中是1）输入我的朴素贝叶斯法函数中，得到预测结果。将预测结果与实际申请结果进行比较，发现**准确率为83.6%**。

此时测试集中录取率为70.4%，也就是说，如果只按照先验概率，全部预测为“Offer”，准确率为70.4%；运用朴素贝叶斯法提高了10%的准确率，说明使用该模型产生了一定的效果。

4 对比

4.1 与R包函数的对比

调用 R 包“klaR”，其中的函数 NaiveBayes 也可以执行朴素贝叶斯法。将我在上一节中用于训练的数据输入该函数中，得到对象 fit_NB，再将fit_NB和测试数据输入 predict 函数，得到了该函数的预测结果。

与我的预测进行对比。首先发现二者的准确率相同；进一步发现二者的所有预测结果也完全相同，说明我实现的函数应当是正确的。为了更加确认这一结果，我又变换了训练集与测试集的切分方式（即改变 set.seed），对于每一种训练集与测试集，都用我的函数与R包函数进行预测并比较，发现二者始终相同。因此从结果上讲，我的函数成功实现了朴素贝叶斯法。

在其他细节中，首先我的函数与R包函数都可以设置平滑值λ；其次，由于我仅仅是比较相对大小，没有对概率进行归一化，因此无法输出具体的概率值，这是我的函数的不足之处，有待改进。

4.2 与其他分类算法的对比

1、逻辑斯蒂回归

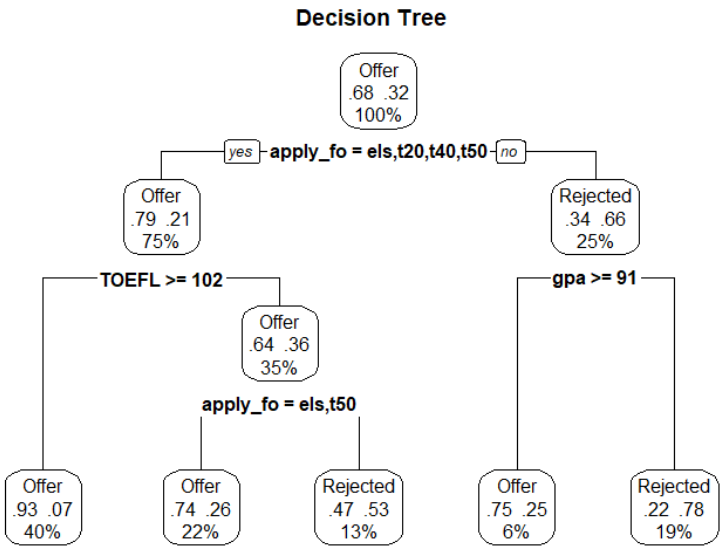
由于在课堂中学习并实现过逻辑斯蒂回归，有一定的基础，因此在这里进行一定的对比与探讨。首先，当变量非数值型，是定性变量时（如 apply_for、from），回归模型需要加入变量的水平数-1的哑变量才可以进行回归。其次，回归分析是基于一定的假设，并且需要处理异常值，因此直接使用，会有模型错误的风险。

直接调用 glm 函数，查看结果，回归系数表如下所示，发现只有4个系数显著，且预测效果很差，准确率仅16.4%；可能需要对数据及变量做进一步的分析与调整。

```
Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)     -0.49441   32.98291  -0.015  0.98804
apply_fortop10   6.17408    2.25131   2.742  0.00610 **
apply_fortop20   1.66965    1.18620   1.408  0.15926
apply_fortop30   1.92139    1.61790   1.188  0.23500
apply_fortop40   1.71987    1.53012   1.124  0.26101
apply_fortop5    4.50681    1.94145   2.321  0.02027 *
apply_fortop50  -1.94615    1.60929  -1.209  0.22654
gpa             -0.15284    0.10351  -1.477  0.13980
TOEFL           -0.32332    0.11143  -2.902  0.00371 **
GRE              0.13803    0.12531   1.102  0.27067
from211 & 985    0.01341    1.01285   0.013  0.98944
from985         -3.06154    1.44060  -2.125  0.03357 *
fromOversea    -14.34407 1685.80814  -0.009  0.99321
from双非        -1.51538    1.29544  -1.170  0.24209
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

2、决策树

调用 R 包“rpart”，其中的函数 rpart 可以执行基于 CART 算法的决策树模型。运用该模型预测，准确率为72.1%，说明有一定的效果。由于是分类问题，因此生成的是分类树，如下图所示。



从图中我们可以看到，根节点首先是根据申请目标 apply_for 进行分类，如果申请的是较靠后的学校（top40、top50、else），则向左，有更大的录取率；否则向右，有较大几率被拒绝。在申请较靠后的学校的情况下，如果托福成绩大于等于102，则向左，被录取，且概率极大；否则向右，若申请的是40名以后的学校，则录取，若不是，则拒绝。在申请较靠前的学校的情况下，若gpa大于等于91，则录取；否则拒绝。

这样的划分准则与我们在现实中的认知是十分吻合的：一般情况下，较好的学校会要求托福成绩100以上；并且gpa上90被公认是较有竞争力的体现。因为在我的样本中，GRE成绩的分布较为集中，可能因此不被视为是有效的划分特征，所以在决策树中没有出现。

5　结论

经过以上的几个阶段，我通过爬虫获得了美研统计学专业申请情况的一手数据，并对数据进行描述，有了直观的了解；然后通过学习并编程实现朴素贝叶斯法，我成功地完成了关于能否成功申请美研统计学专业的预测，并与R包函数和其他分类方法进行了对比，完成了本次报告的主题。对于本文的优点与不足，我总结如下：

优点：

1、运用R语言通过网络爬虫获取了第一手的资料，数据较为全面，且有一定的现实参考意义。

2、学习研究朴素贝叶斯法，以及其他分类算法，可以为将来进一步学习统计学习等知识作铺垫。

3、代码全部由自己编写、修改、调整，并且能够直接顺利运行。

不足：

1、实际上数据仅能反映“寄托天下”论坛中的用户的申请情况，且是愿意汇报自己结果的同学，因此有可能产生幸存者偏差，无法很好地反映总体情况。

2、由于数据并不能很好地符合朴素贝叶斯法的前提假定，因此牺牲了一定的分类准确率。

3、由于水平所限，代码中很多部分都是通过for循环完成，仍有改进的空间。

参考书

[1] 李航．统计学习方法[M]．北京：清华大学出版社，2012。

[2]（美）卡巴科弗著；王小宁等译．R语言实战：第2版[M]．北京：人民邮电出版社，2016。

[3] Deborah Nolan, Duncan Temple Lang． Data Science in R ： A Case Studies Approach to Computational Reasoning and Problem Solving[M]． CRC Press，2015