**华中科技大学计算机科学与技术学院**

**机器学习报告**



专 业： 计算机科学与技术

班 级： 计算机xxxx班

学 号： U20xxxxxxx

姓 名： xxx

成 绩：

指导教师： xxx

**完成日期： xxxx年xx月xx日**

# 题目六

## 一、实验题目：基于朴素贝叶斯分类器的语音性别识别

## 二、实验要求

1.题目背景

使用朴素贝叶斯在语音上做一个小应用——分辨声音是男性还是女性。具体题目参考https://www.kaggle.com/primaryobjects/voicegender。

2.数据集

在上述参考网站中可以下载。这个数据集是基于对男女语音段进行合理的声音预处理而得到的语音特征(并不包含原始语音段)。集合中共有3168条数据，男女各有1584条，每条数据可视作一个长度为21的一维数组。其中前20个数值是这条语音的20个特征值，这些特征值包括了语音信号的长度、基频、标准差、频带中值点\一分位频率\三分位频率等；最后一个数值是性别标记。数据集中直接以字符串，即male和female进行标注。按照7：3的比例划分数据集为训练集和测试集。

3.任务描述

使用朴素贝叶斯的方法，可以先对所有特征值做统计，通过连续性参数估计(高斯分布)方法得到参数。而后使用预测函数预测测试集。

4.评测标准

要求得到2×2预测情况，如表2-1所示。

表2-1 实验预测情况评测标准

|  |  |
| --- | --- |
| 男声正确率 | 男声错误率 |
| 女声正确率 | 女声错误率 |

## 三、算法设计

1.数据预处理

观察数据集是否有缺失，缺失的情况如何，数据集部分截图如图3-1所示。

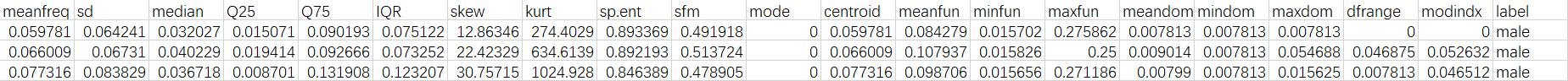


图3-1 数据集部分截图

一条数据有20个特征和该条数据的分类标签(male或female)。数据集中存在数据的缺失，对数据缺失的情况进行统计。统计的结果如表3-1和图3-2所示。

表3-1 数据集的数据缺失情况统计

|  |  |  |  |
| --- | --- | --- | --- |
| 特征名称 | mode(声频的模频率) | dfrange(声信号主频测量范围) | modindx(调制指数) |
| 缺失数据量 | 236 | 65 | 65 |
| 缺失比例 | 7.45% | 2.05% | 2.05% |
| 有数据缺失的特征占总特征的15% | | | |

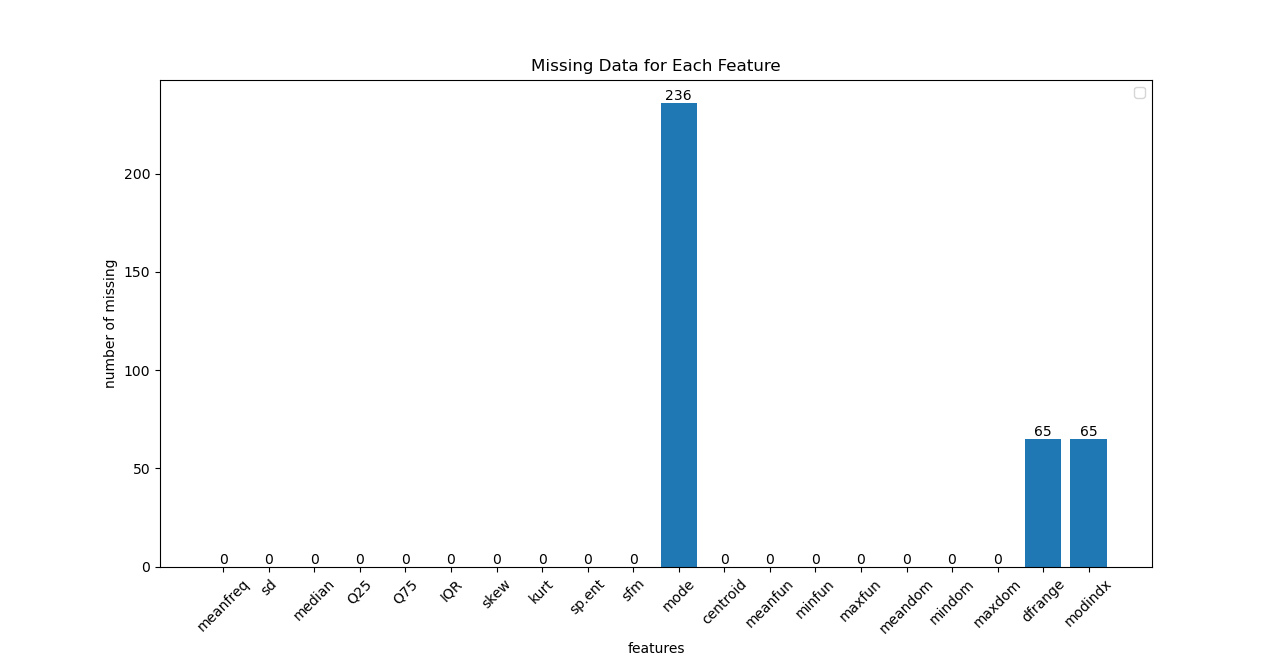


图3-2 数据集的数据缺失情况统计柱状图

有数据缺失的特征占比不大，特征中缺失的数据量占比也不大。因此，用有缺失数据的特征的平均值代替其缺失的数据。

2.朴素贝叶斯分类器

(1)数据具有连续特征——高斯模型

由图3-1知，数据的特征具有连续性，可以使用高斯模型。

其中、分别表示类别为的样本中，第维特征的均值、方差。

①分别求出训练集中男女声在每一特征向量下的平均值和方差。记男声为、，女声为、。

②分别求出训练集中男女声的比例。记男声为，女声为。

③计算

其中n=20。记男声为，女声为。

④比较与的大小。若，则测试数据为男声；若，则测试数据为女声。

(2)工程应用——离散化

考虑到训练集占总数据集的70%，即有3168×70%=2218条数据参与训练过程，这样的数据量可能不足以体现出数据满足高斯分布。考虑到本次实验实质上是朴素贝叶斯的工程应用，先将连续的数据进行离散化，而后使用贝叶斯定理。

假设20个特征向量为。表示特征取第个分类。则一个测试样本向量可以表示为：test\_vector=[]，其为男声的概率为：

。

①对数据进行离散化。

其中、分别为该特征的最大值和最小值，为离散程度，为向下取整。需要找到一个合适的。

②得到训练集中任一样本分类为male的概率，分类为male情况下的各特征的概率，所有分类下各特征的概率。计算连乘时，为避免某些特征值概率为0而影响总体概率，每个特征值最少个数为1，取对数减小四舍五入的误差。

③计算得到男声概率，女声概率。比较与的大小。若，则测试数据为男声；若，则测试数据为女声。

3.朴素贝叶斯分类器性能评估

计算训练结果的准确率、召回率、F1值。引入原数据集上的多项式模型训练结果，将三种分类器的结果进行对比。

## 四、实验环境与平台

实验环境条件：i7-8550U 1.80GHz，256M内存；WINDOWS10下python3.8.0。实验中使用到的python库为matplotlib3.2.2(绘图)、numpy1.19.0(数据处理)、pandas1.0.5(数据集文件读取)、sklearn0.23.1(机器学习)。

## 五、程序实现

1.高斯模型

查阅相关资料，了解到可以使用sklearn库中的朴素贝叶斯高斯模型进行实验，具体代码如下。

#载入库及函数

import pandas as pd

from sklearn.preprocessing import LabelEncoder

from sklearn.impute import SimpleImputer

from sklearn.model\_selection import train\_test\_split

from sklearn.naive\_bayes import MultinomialNB

from sklearn.naive\_bayes import GaussianNB

from sklearn.metrics import classification\_report

from sklearn.metrics import precision\_score

from sklearn.metrics import recall\_score

from sklearn.metrics import f1\_score

def sk\_Gaussian(file\_name):

"""

:param

file\_name:文件名

:return

info\_Gaussian:男性正确率、女性正确率、高斯贝叶斯准确率

"""

#数据集读取和预处理

data=pd.read\_csv(file\_name)

x=data.iloc[:,:-1]

y=data.iloc[:,-1]

y=LabelEncoder().fit\_transform(y)

imp=SimpleImputer(missing\_values=0,strategy='mean')

x=imp.fit\_transform(x)

#按照7:3的比例划分训练集和测试集

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.3)

#高斯朴素贝叶斯预测

gnb=GaussianNB()

gnb.fit(x\_train,y\_train)

y\_predict=gnb.predict(x\_test)

#计算准确率指标

precision\_male=precision\_score(y\_test,y\_predict,pos\_label=1) #男性准确率

precision\_female=precision\_score(y\_test,y\_predict,pos\_label=0) #女性准确率

precision\_Gaussian=gnb.score(x\_test,y\_test) #高斯贝叶斯准确率

#将准确率信息打包返回

info\_Gaussian=[]

info\_Gaussian.append(precision\_male)

info\_Gaussian.append(precision\_female)

info\_Gaussian.append(precision\_Gaussian)

return info\_Gaussian

2.离散化

#载入库

import csv

import numpy as np

def load\_data\_set(file\_name,n):

"""

:param

file\_name:文件名

n:对浮点数的特征值进行离散化的离散程度

:return

train\_mat:离散化的训练数据集

train\_classes:训练数据集所属的分类(male/female)

test\_mat:离散化的测试数据集

test\_classes:测试数据集所属的分类(male/female)

label\_name:特征的名称

"""

data\_mat=[]

with open(file\_name)as file\_obj:

voice\_reader=csv.DictReader(file\_obj)

list\_class=[]

#文件头

label\_name=list(voice\_reader.fieldnames)

num=len(label\_name)-1

for line in voice\_reader.reader:

data\_mat.append(line[:num])

gender=1 if line[-1]=='male' else 0

list\_class.append(gender)

#求每一个特征的平均值

data\_mat=np.array(data\_mat).astype(float)

count\_vector=np.count\_nonzero(data\_mat,axis=0)

sum\_vector=np.sum(data\_mat,axis=0)

mean\_vector=sum\_vector/count\_vector

#数据缺失的地方用平均值填充

for row in range(len(data\_mat)):

for col in range(num):

if data\_mat[row][col]==0.0:

data\_mat[row][col]=mean\_vector[col]

#将数据连续型的特征值离散化处理

min\_vetor=data\_mat.min(axis=0)

max\_vetor=data\_mat.max(axis=0)

diff\_vector=max\_vetor-min\_vetor

diff\_vector/=(n-1)

new\_data\_set=[]

for i in range(len(data\_mat)):

line=np.array((data\_mat[i]-min\_vetor)/diff\_vector).astype(int)

new\_data\_set.append(line)

#随机划分数据集为训练集和测试集

test\_set=list(range(len(new\_data\_set)))

train\_set=[]

#从3168个数据集中抽取70%(2218个)的数据做为训练集,剩余为测试集

for i in range(2218):

random\_index=int(np.random.uniform(0,len(test\_set)))

train\_set.append(test\_set[random\_index])

del test\_set[random\_index]

#训练数据集

train\_mat=[]

train\_classes=[]

for index in train\_set:

train\_mat.append(new\_data\_set[index])

train\_classes.append(list\_class[index])

#测试数据集

test\_mat=[]

test\_classes=[]

for index in test\_set:

test\_mat.append(new\_data\_set[index])

test\_classes.append(list\_class[index])

return train\_mat,train\_classes,test\_mat,test\_classes,label\_name

def naive\_bayes(train\_matrix,list\_classes,n):

"""

:param

train\_matrix:训练样本矩阵

list\_classes:训练样本分类向量

n:对浮点数的特征值进行离散化的离散程度

:return

p\_1\_class:任一样本分类为1(male)的概率

p\_1\_feature:分类为1(male)情况下的各特征的概率

p\_feature:所有分类下各特征的概率

"""

#训练样本个数,特征个数

num\_train\_data=len(train\_matrix)

num\_feature=len(train\_matrix[0])

#分类为1的样本占比

p\_1\_class=sum(list\_classes)/float(num\_train\_data)

list\_classes\_1=[]

train\_data\_1=[]

for i in list(range(num\_train\_data)):

if list\_classes[i]==1:

list\_classes\_1.append(i)

train\_data\_1.append(train\_matrix[i])

#分类为1情况下的各特征的概率

train\_data\_1=np.matrix(train\_data\_1)

p\_1\_feature={}

for i in list(range(num\_feature)):

feature\_values=np.array(train\_data\_1[:,i]).flatten()

#避免某些特征值概率为0而影响总体概率,每个特征值最少个数为1

feature\_values=feature\_values.tolist()+list(range(n))

p={}

count=len(feature\_values)

for value in set(feature\_values):

#多个概率相乘结果较小,取对数减小四舍五入的误差

p[value]=np.log(feature\_values.count(value)/float(count))

p\_1\_feature[i]=p

#所有分类下的各特征的概率

p\_feature={}

train\_matrix=np.matrix(train\_matrix)

for i in list(range(num\_feature)):

feature\_values=np.array(train\_matrix[:,i]).flatten()

feature\_values=feature\_values.tolist()+list(range(n))

p={}

count=len(feature\_values)

for value in set(feature\_values):

#多个概率相乘结果较小,取对数减小四舍五入的误差

p[value]=np.log(feature\_values.count(value)/float(count))

p\_feature[i]=p

return p\_1\_class,p\_1\_feature,p\_feature

def classify\_bayes(test\_vector,p\_1\_class,p\_1\_feature,p\_feature):

"""

:param

test\_vector:需要分类的测试向量

p\_1\_class:任一样本分类为1(male)的概率

p\_1\_feature:分类为1(male)情况下的各特征的概率

p\_feature:所有分类下各特征的概率

:return 一个数字(1->male,0->female)

"""

#计算每个分类的概率(概率相乘取对数=概率各自对数相加)

sum=0.0

for i in list(range(len(test\_vector))):

sum+=p\_1\_feature[i][test\_vector[i]]

sum-=p\_feature[i][test\_vector[i]]

p1=sum+np.log(p\_1\_class)

p1=np.exp(p1)

p0=1-p1

if p1>p0:

return 1

else:

return 0

def test\_bayes(file\_name,n):

"""

:param

file\_name:文件名

n:对浮点数的特征值进行离散化的离散程度

:return

evaluation:评估指标

p\_1\_class:任一样本分类为1(male)的概率

"""

#建立训练集与测试集

train\_mat,train\_classes,test\_mat,test\_classes,label\_name=load\_data\_set(file\_name,n)

#得到贝叶斯参数

p\_1\_class,p\_1\_feature,p\_feature=naive\_bayes(train\_mat,train\_classes,n)

count\_info=[0.0]\*9

for i in list(range(len(test\_mat))):

test\_vector=test\_mat[i]

result=classify\_bayes(test\_vector,p\_1\_class,p\_1\_feature,p\_feature)

if test\_classes[i]==1:

count\_info[0]+=1 #男性人数

if result==test\_classes[i]:

count\_info[1]+=1 #正确人数

else:

count\_info[2]+=1 #错误人数

else:

count\_info[3]+=1 #女性人数

if result==test\_classes[i]:

count\_info[4]+=1 #正确人数

else:

count\_info[5]+=1 #错误人数

count\_info[6]+=1

count\_info[7]=count\_info[1]+count\_info[4] #总正确人数

count\_info[8]=count\_info[2]+count\_info[5] #总错误人数

#评估指标计算

evaluation=[]

male\_eva=[] #男性评估指标

male\_eva.append(count\_info[1]/count\_info[0]) #男性正确率

male\_eva.append(count\_info[2]/count\_info[0]) #男性错误率

male\_eva.append(count\_info[1]/(count\_info[1]+count\_info[5])) #男性召回率

male\_eva.append(2\*male\_eva[0]\*male\_eva[2]/(male\_eva[0]+male\_eva[2])) #男性F1值

female\_eva=[] #女性评估指标

female\_eva.append(count\_info[4]/count\_info[3]) #女性正确率

female\_eva.append(count\_info[5]/count\_info[3]) #女性错误率

female\_eva.append(count\_info[4]/(count\_info[4]+count\_info[2])) #女性召回率

female\_eva.append(2\*female\_eva[0]\*female\_eva[2]/(female\_eva[0]+female\_eva[2])) #女性F1值

total\_eva=[] #总体评估指标

total\_eva.append(count\_info[7]/count\_info[6]) #总正确率

total\_eva.append(count\_info[8]/count\_info[6]) #总错误率

evaluation.append(male\_eva)

evaluation.append(female\_eva)

evaluation.append(total\_eva)

return evaluation,p\_1\_class

3.其他

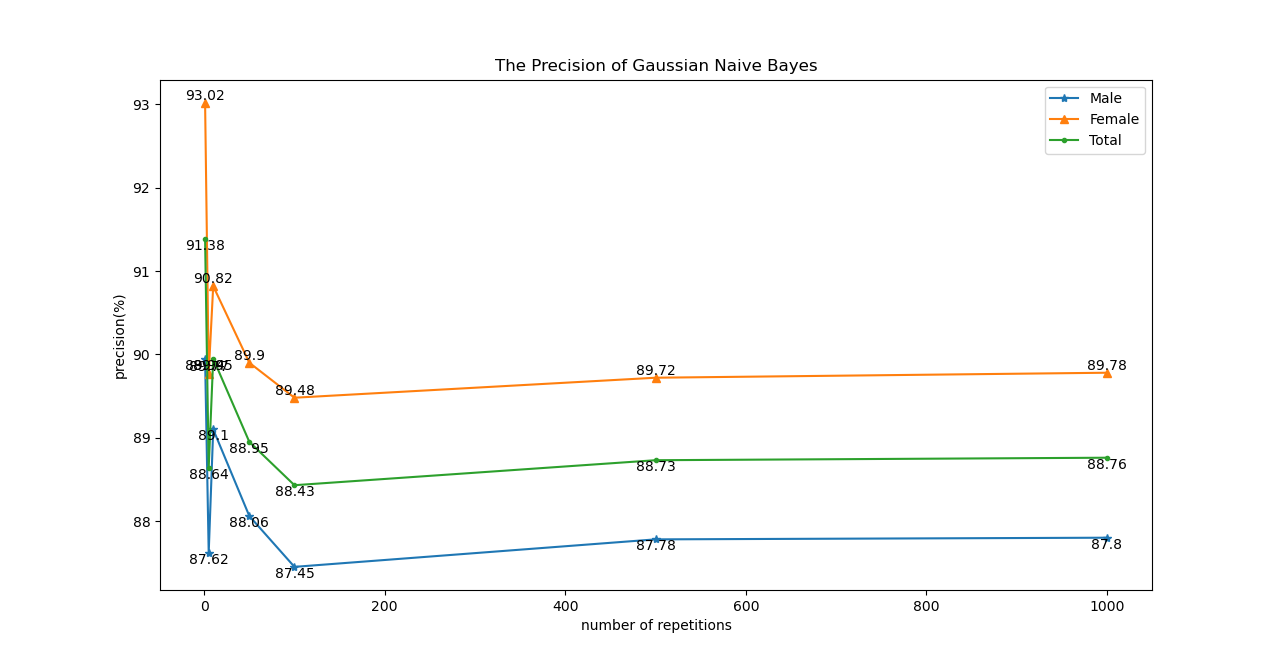
更多实验代码请查看我的github

https://github.com/BigPants2333/machine\_learning\_combat/tree/master/naivebayes\_gendervoice

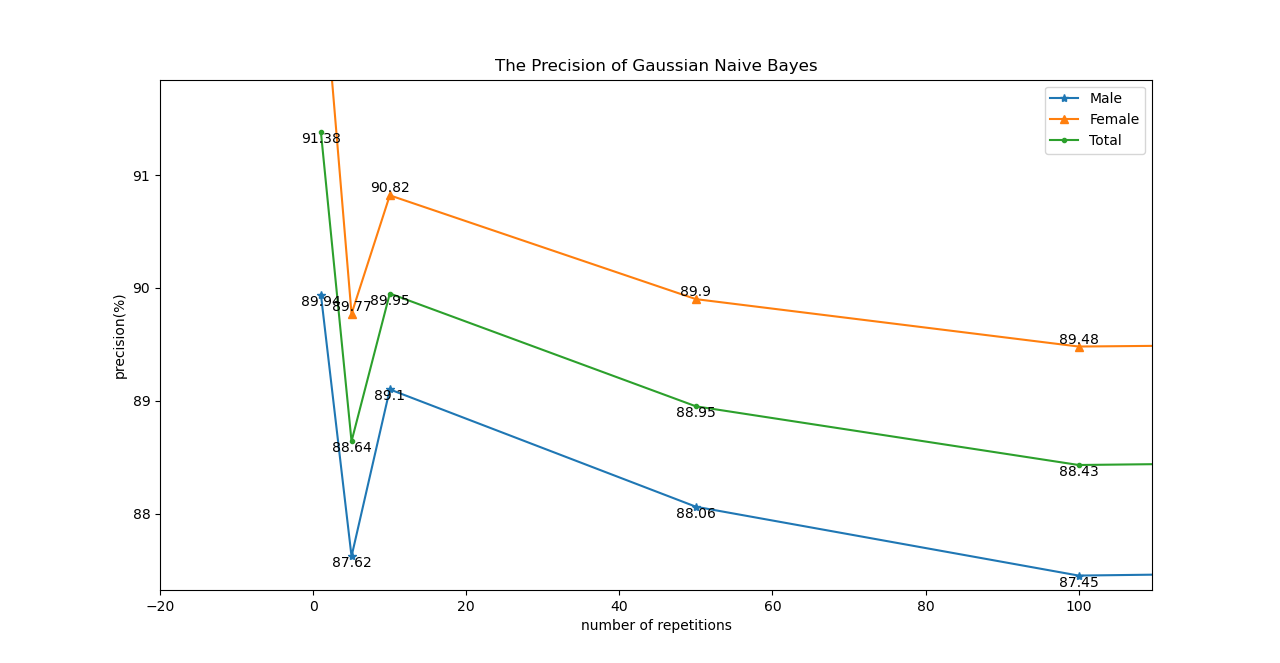
## 六、实验结果

1.高斯模型

对高斯模型进行重复训练，重复次数分别为1、5、10、50、100、500、1000。将重复训练过程中得到的男、女声和总体正确率取平均值，并绘制图像，如图6-1所示。由图可知，平均正确率随训练次数的增加呈现出先降低后升高再降低最后趋于平稳的特点，男声平均正确率上下浮动不超过3%，女声平均正确率上下浮动不超过4%，总体平均正确率上下浮动不超过3%。



(a) 整体趋势



(b)数据重叠部分放大

图6-1 高斯模型重复训练的平均正确率(趋势)

考虑到对高斯模型进行重复训练得到的平均正确率呈现出的特点，在平稳区间[100,1000]每隔100取一个值继续进行重复训练，得到的结果如图6-2所示。由图可知，男声平均正确率稳定在87.50%-88.00%，女声平均正确率稳定在89.50%-90.00%，总体平均正确率稳定在88.50%-89.00%。三种平均正确率上下浮动均不超过0.5%，可以认为平均正确率趋于稳定。

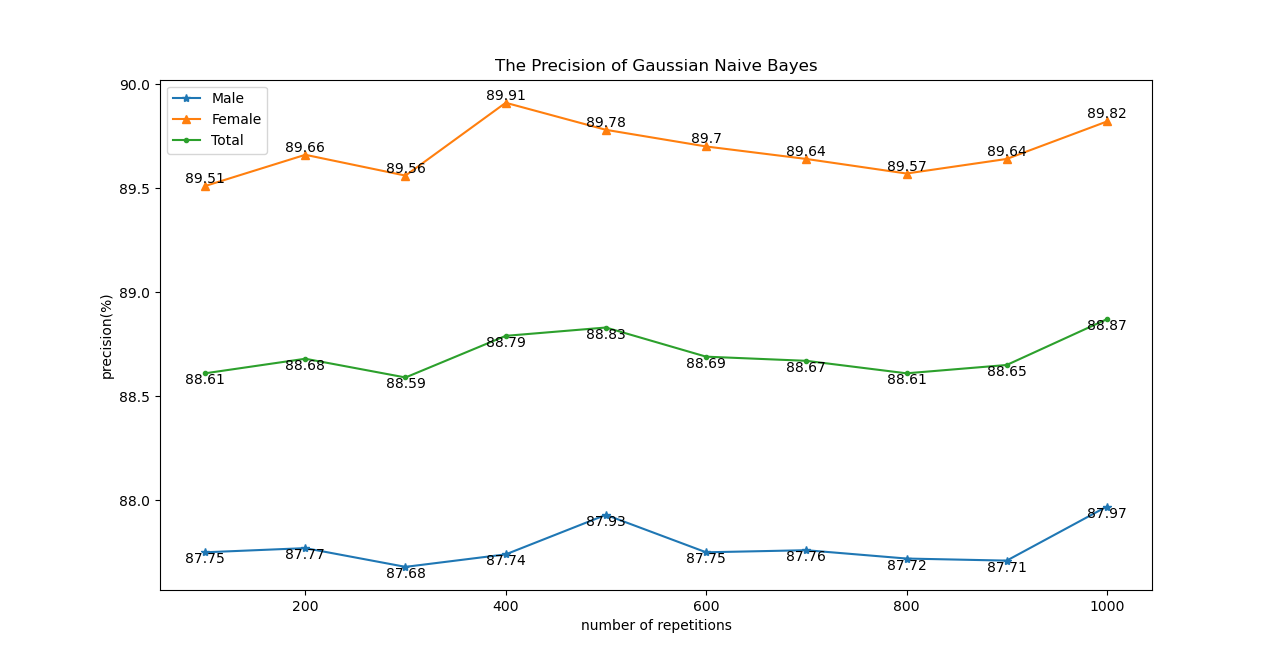


图6-2 高斯模型重复训练的平均正确率(稳定)

2.离散化

(1)离散程度n的选取

选取区间[2,2000]中的整数作为离散程度n，仅作一次训练，得到男、女声和总体正确率，并绘制图像，如图6-3所示。整体上来看，男声识别正确率在n较小时较高，随着n的增大而降低；女声识别正确率较为高且稳定；总体识别正确率在n较小时较高，随着n的增大而缓慢降低。

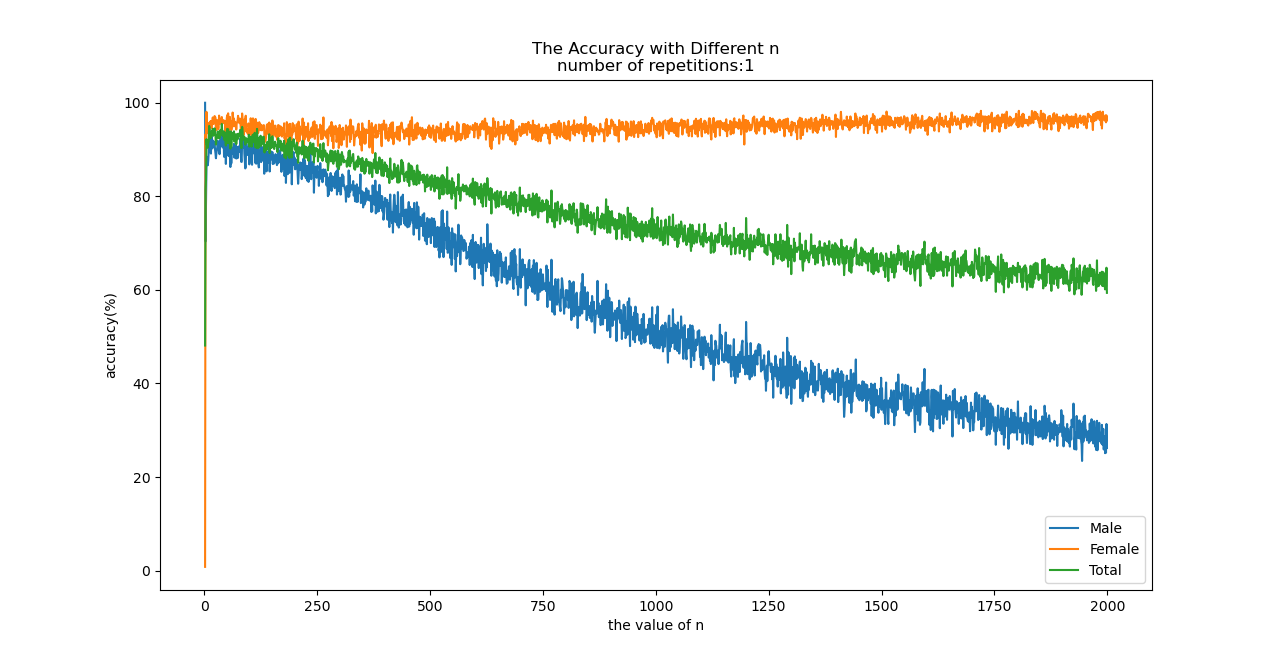
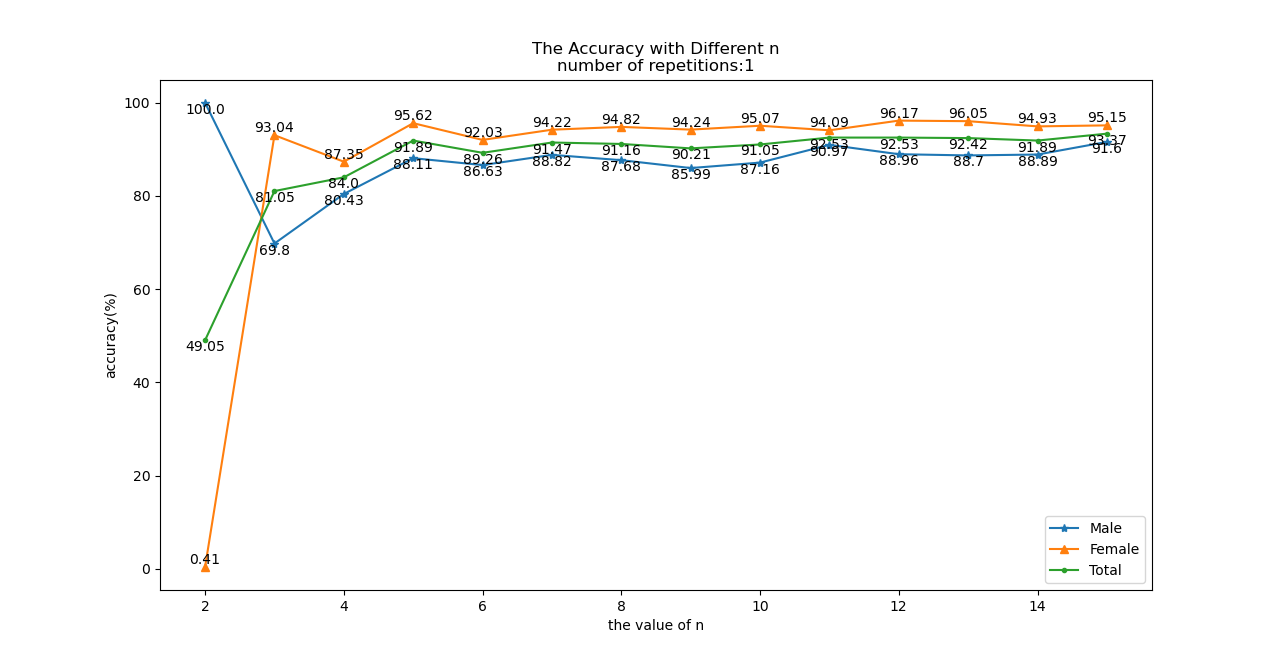
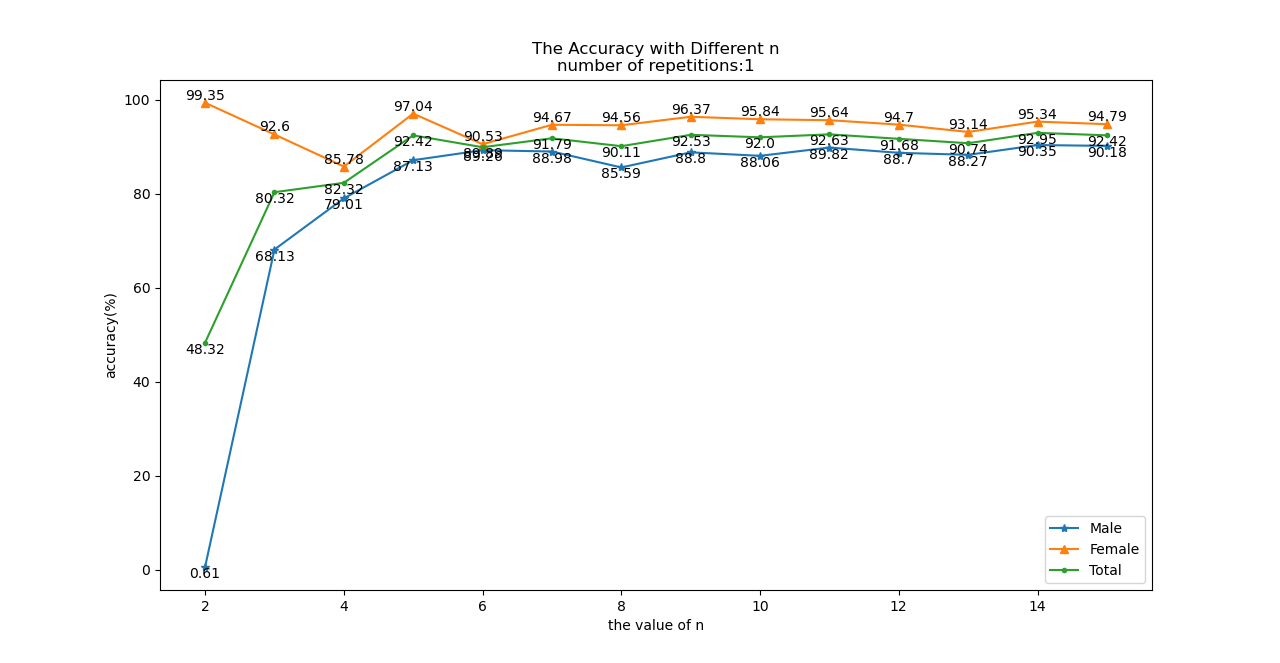


图6-3 正确率随离散程度增加的变化趋势

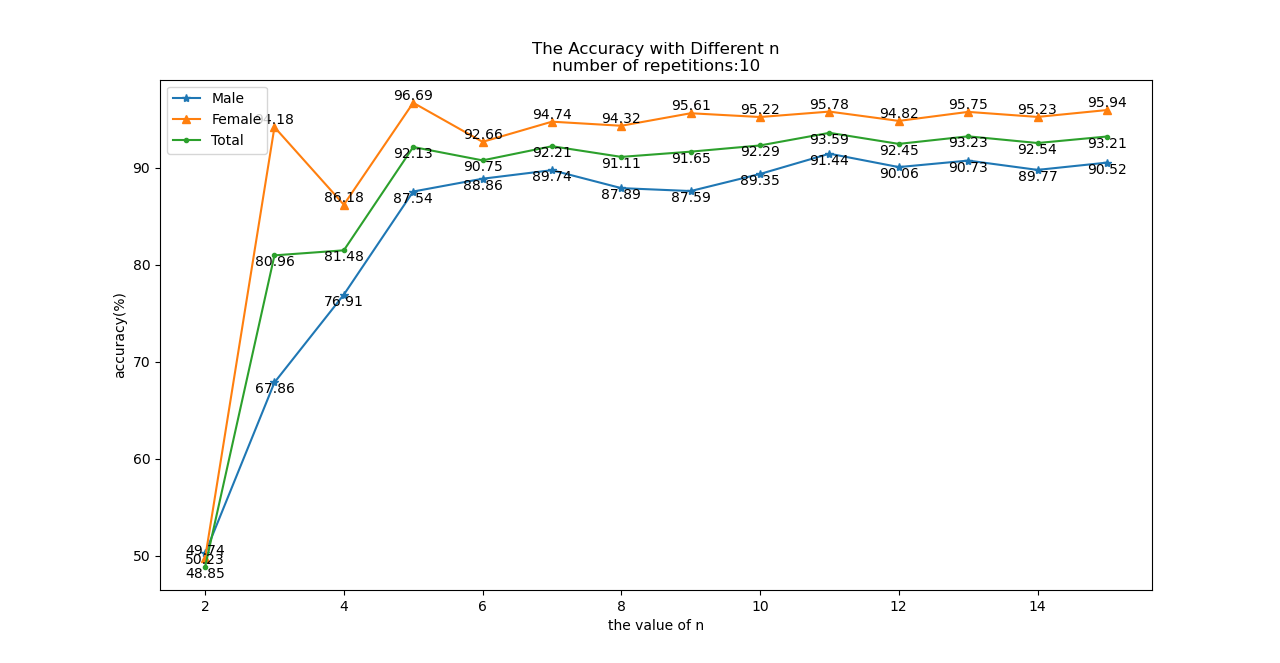
为进一步得到训练结果较好的n的取值，选取区间[2,15]中的整数作为离散程度n，并进行重复训练，重复次数分别为1、10、100、1000。将重复训练过程中得到的男、女声和总体正确率取平均值，并绘制图像，如图6-4所示。只进行一次训练时，在n<5的情况下，正确率较低且偶然性较大，如n=2时，可能男性正确率较高也可能女性正确率较高；当进行多次训练时，在n>10的情况下，正确率较为稳定且较高。因此，取离散程度n=11进行后续实验。



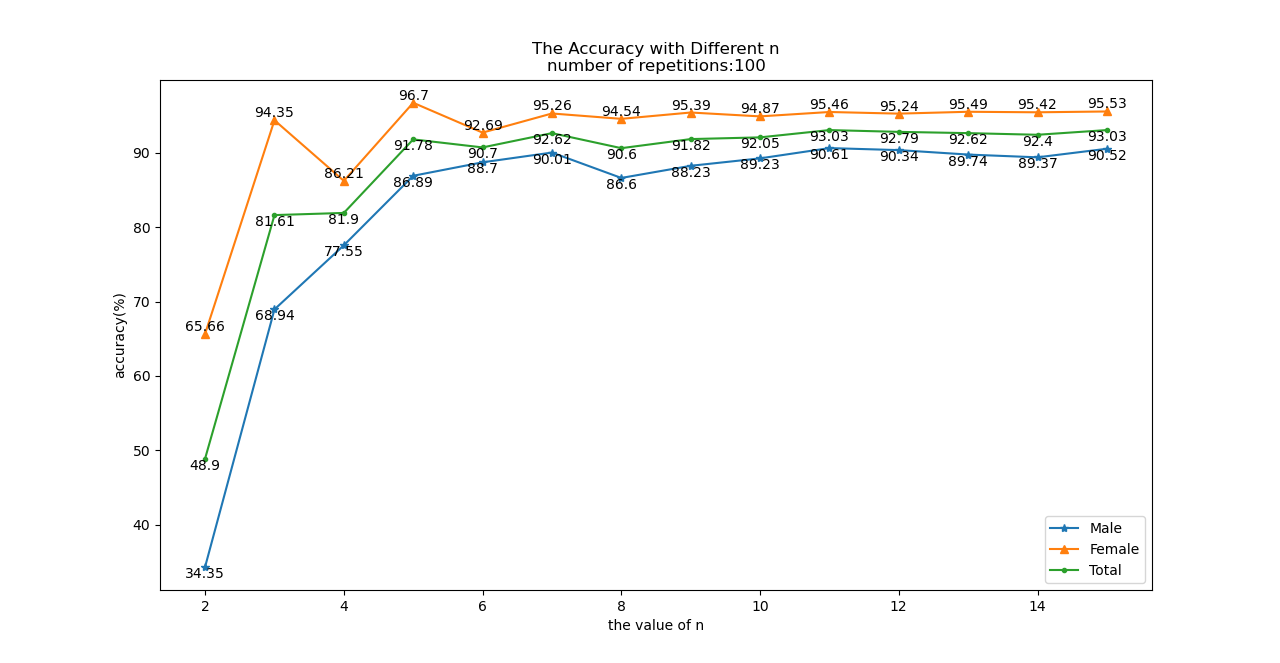
(a) 训练模型一次(n=2时男性正确率较高)



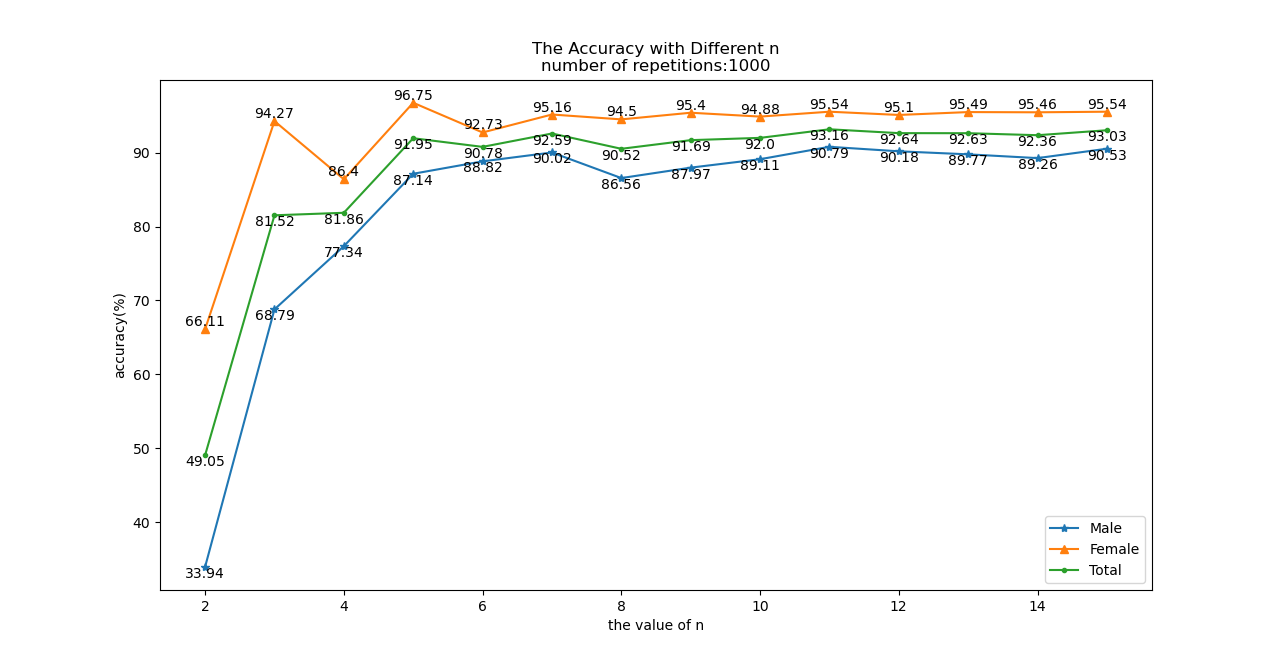
(b) 训练模型一次(n=2时女性正确率较高)



(c) 重复训练模型10次



(d) 重复训练模型100次

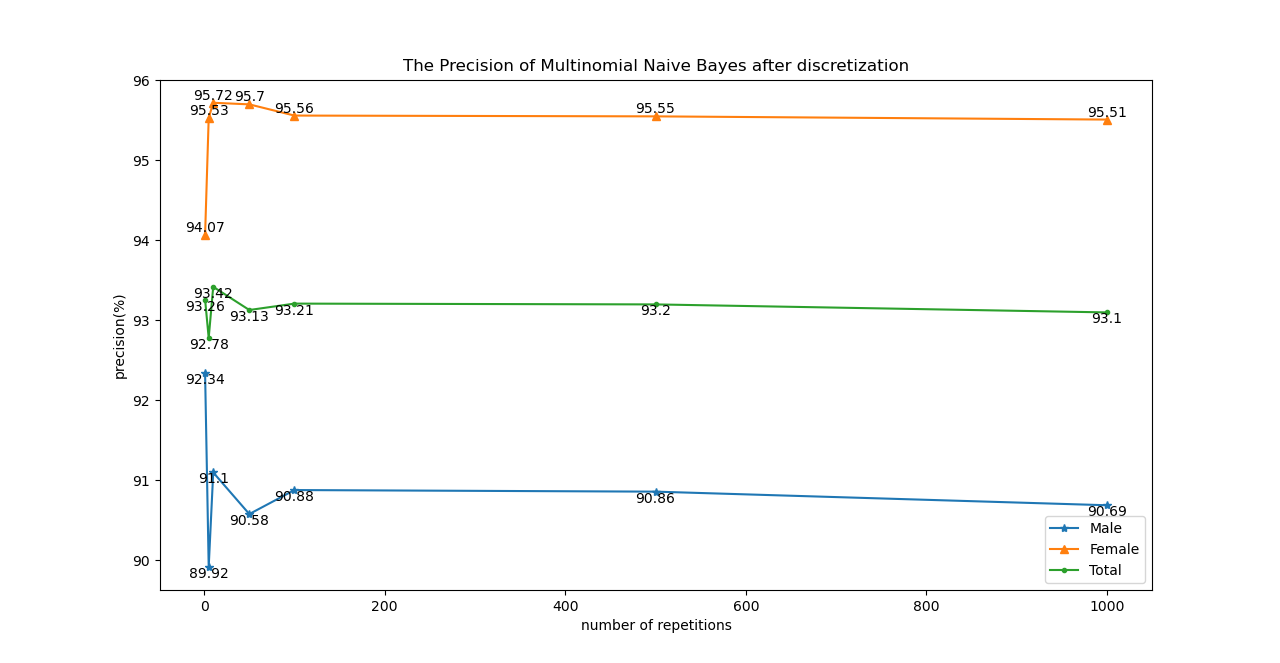


(c) 重复训练模型1000次

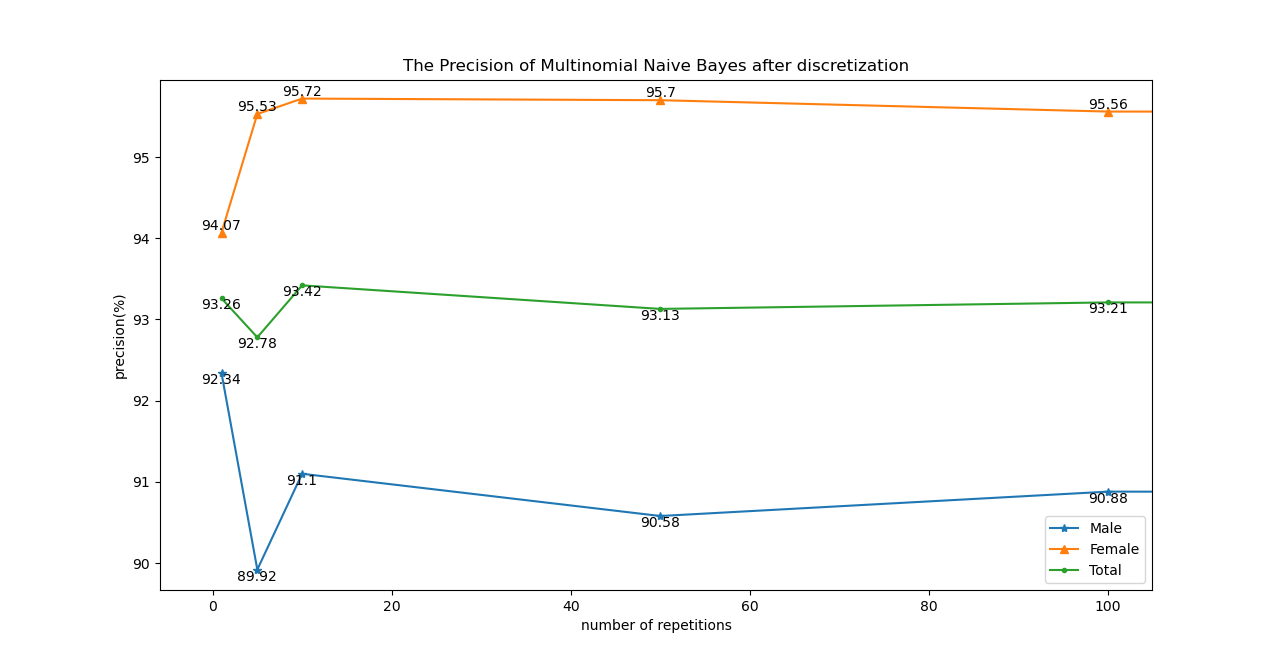
图6-4 离散程度较小时重复训练的平均正确率

(2)重复训练结果

在离散程度n=11的情况下，对模型进行重复训练，重复次数分别为1、5、10、50、100、500、1000。将重复训练过程中得到的男、女声和总体正确率取平均值，并绘制图像，如图6-5所示。男声正确率呈现出先下降后上升再下降再上升最后趋于稳定的特点，整体上下浮动不超过3%；女声正确率呈现出先上升再下降最后趋于稳定的特点，整体上下浮动不超过2%；总体正确率呈现出先下降后上升再下降再上升最后趋于稳定的特点，整体上下浮动不超过1%。



(a) 总体趋势



(b) 数据重叠部分放大

图6-5 离散程度n=11时重复训练的平均正确率(趋势)

考虑到离散程度n=11时，对模型进行重复训练得到的平均正确率呈现出的特点，在平稳区间[100,1000]每隔100取一个值继续进行重复训练，得到的结果如图6-6所示。由图可知，男声平均正确率稳定在90.60%-90.90%，女声平均正确率稳定在95.40%-95.60%，总体平均正确率稳定在93.00%-93.20%。三种平均正确率上下浮动均不超过0.3%，可以认为平均正确率趋于稳定。

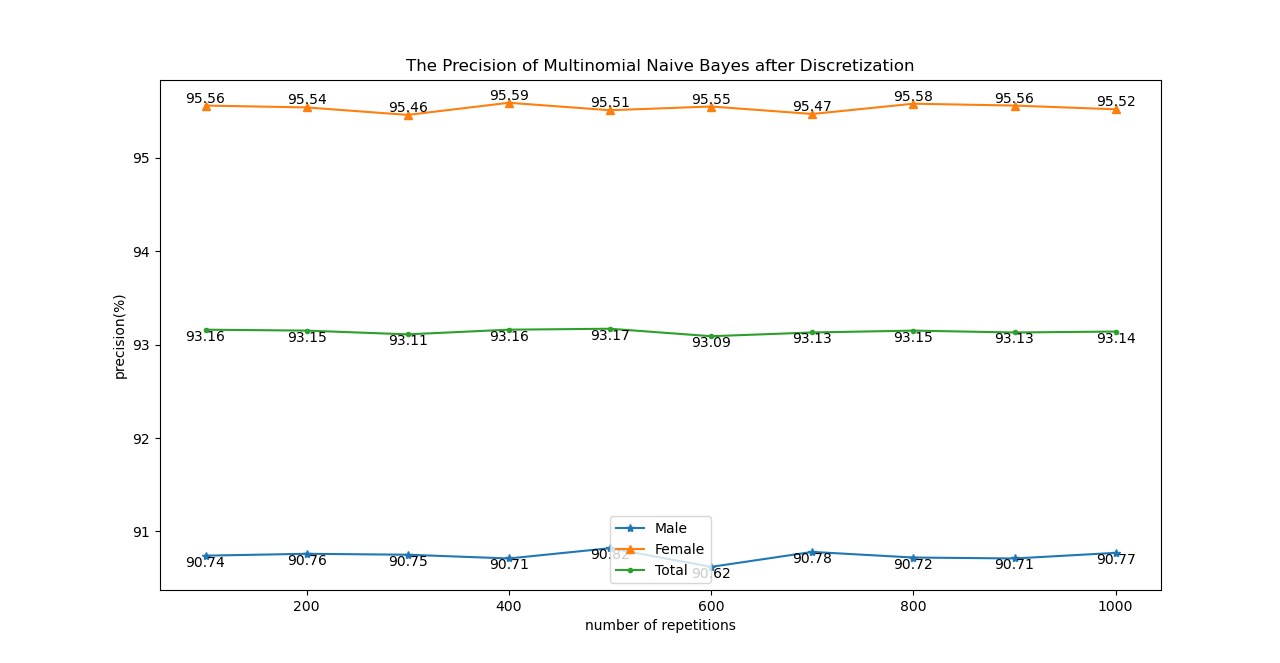


图6-5 离散程度n=11时重复训练的平均正确率(稳定)

## 七、结果分析

1.男声正确率小于女声正确率

不论是高斯模型还是离散化后的模型，在经过重复训练得到稳定的结果后，均表现出男声正确率小于女声正确率的特点。基于此，对原始数据集的数据缺失情况进行了进一步统计研究，结果如图7-1、图7-2和图7-3所示。

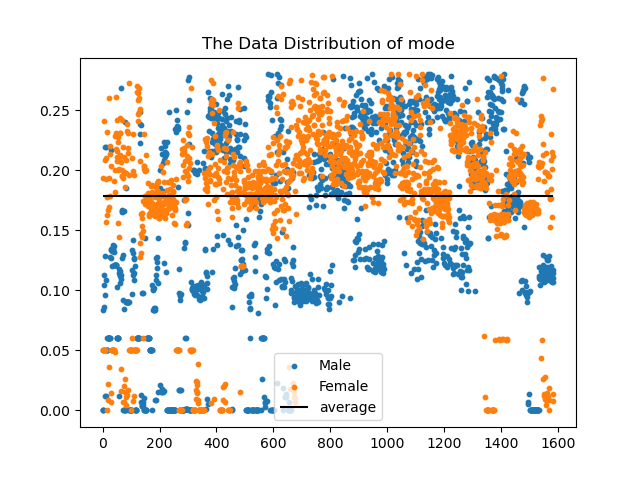
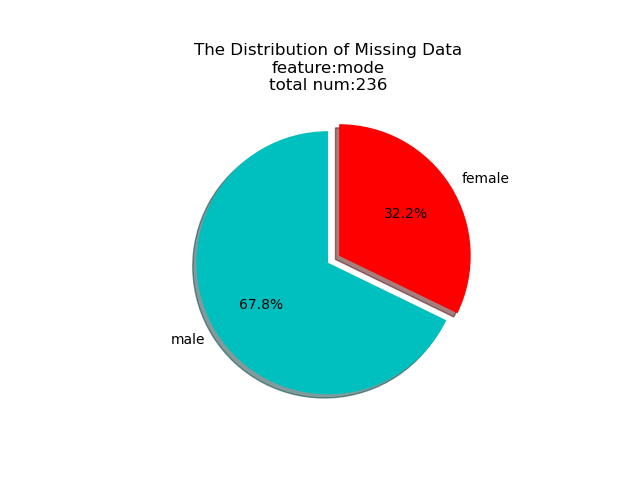


图7-1 特征mode的分布特点

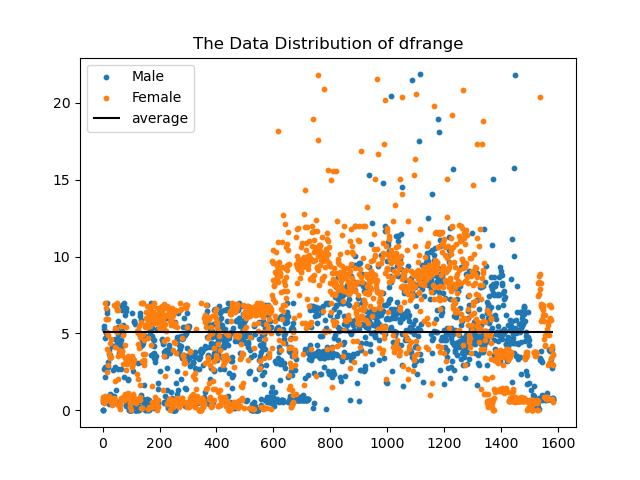
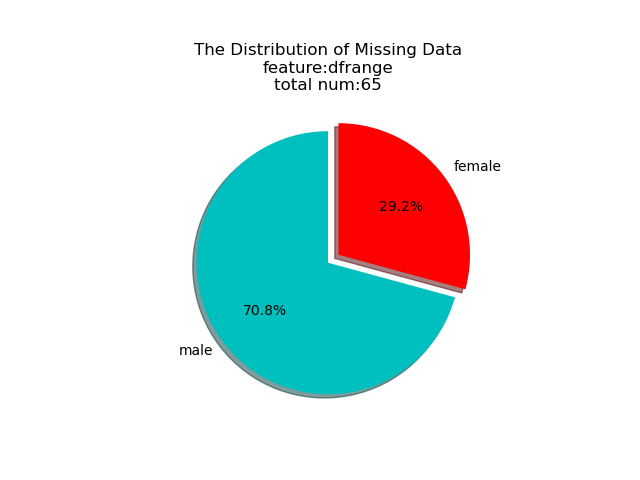


图7-2 特征dfrange的分布特点

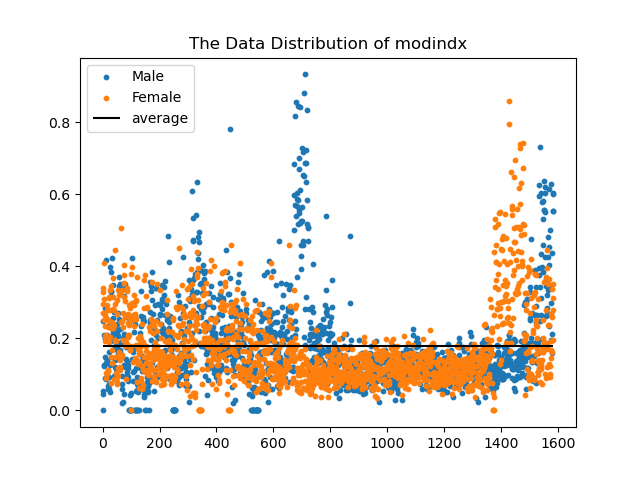
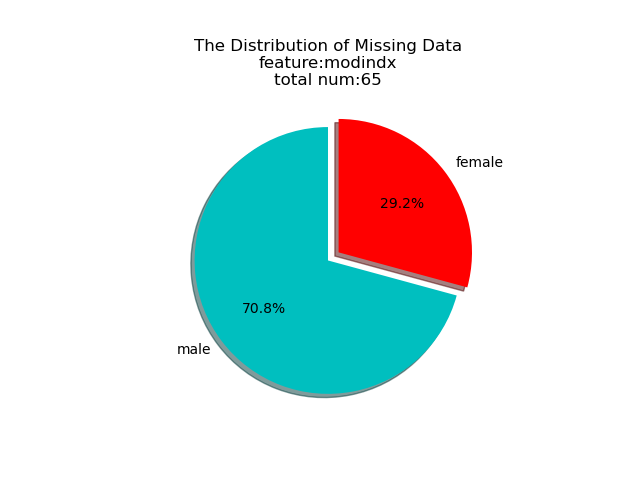
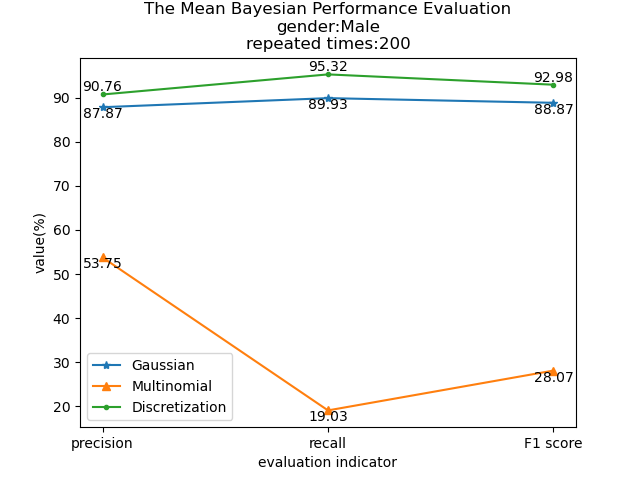


图7-1 特征modindx的分布特点

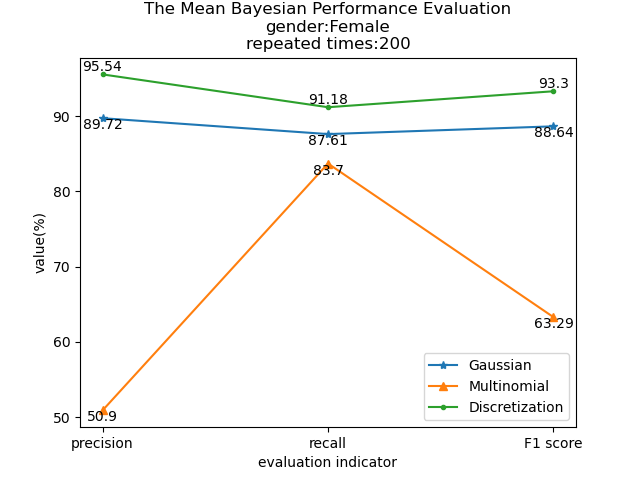
在有数据缺失的三个特征中，男声的数据缺失占比高于女声的数据缺失；使用总体的平均值代替缺失数据造成了一定的误差，而占比较高的男声对这种误差更为敏感。这可能就是造成男声正确率小于女声正确率的原因。可以在有数据缺失的特征中，使用男声的数据平均值替代男声缺失数据，使用女声的数据平均值替代女声缺失数据，加以改进。

2.离散化后的模型优于高斯模型

引入原数据集上未经离散化处理的多项式模型训练结果，将三种分类器的结果进行对比，如图7-4所示。离散化后的模型准确率、召回率、F1值均优于高斯模型，远优于未经离散化处理的多项式模型。



(a) 男声



(b) 女声

图7-4 三种贝叶斯分类器性能对比

一方面，训练集占总数据集的70%，即有3168×70%=2218条数据参与训练过程，这样的数据量可能不足以体现出数据满足高斯分布。另一方面，使用离散化的方法解决工程应用问题，既对异常数据有很强的鲁棒性，又相当于为模型引入了非线性，能够提升模型表达能力。这可能就是离散化后的模型优于高斯模型的原因。可以增加训练集的数量和增加特征向量的数量来增强高斯模型性能。

# 参考文献

[1]Kory Becker :Gender Recognition by Voice, https://www.kaggle.com/primaryobjects/voicegender

[2]Kory Becker :Voice Gender, https://github.com/primaryobjects/voice-gender

[3]何琨:机器学习内部讲义,第25-31页

[4]朴素贝叶斯理论推导与三种常见模型, https://blog.csdn.net/u012162613/article/details/48323777

[5]廖雪峰:python教程, https://www.liaoxuefeng.com/wiki/1016959663602400

[6]python的sklearn库的基本用法, https://www.cnblogs.com/zx931880423/p/11543256.html

[7]python: matplotlib数据可视化, https://www.jianshu.com/p/b694a18b765a

[8]Kaggle实战——Gender Recognition by Voice声音的性别区分, https://blog.csdn.net/peinbill/article/details/80106790

[9]机器学习——用朴素贝叶斯分类法辨别男女声音, https://www.cnblogs.com/beiyan/p/8337991.html