

# Python编程及人工智能应用

## 第六章 朴素贝叶斯分类及Python实现

<https://bolei-zhang.github.io/course/python-ai.html>

- K均值聚类一定会收敛（达到稳定状态）吗？
- 定理：对于任意给定的迭代聚类中心初值（或者任意给定的一种划分方式），K-means算法的目标函数一定会收敛。
- 证明思路：在每次更新节点分配（Update cluster assignment）和聚类中心（Update Cluster centers）的时候，目标函数不增

$$V = \sum_{i=1}^k \sum_{x \in S_i} (x - \mu_i)^2$$

- 理解和掌握**朴素贝叶斯分类**的定义
- 掌握朴素贝叶斯分类的Python实现方法
- 了解**连续型特征问题的朴素贝叶斯分类法**

# 贝叶斯分类法简介



**托马斯·贝叶斯** ( Thomas Bayes , 1702-1763 ) , 18世纪英国神学家、数学家、数理统计学家和哲学家, 概率论理论创始人, 贝叶斯统计的创立者, “归纳地”运用数学概率, “从特殊推论一般、从样本推论全体”的第一人。



所谓的贝叶斯方法源于他生前为解决一个“逆概”问题写的一篇文章, 而这篇文章是在他死后才由他的一位朋友发表出来的。

在贝叶斯写这篇文章之前, 人们已经能够计算“**正向概率**”, 如“假设袋子里面有 $N$ 个白球,  $M$ 个黑球, 你伸手进去摸一把, 摸出黑球的概率是多大”。

而一个自然而然的问题是反过来的“**逆概率问题**”: 如果我们事先并不知道袋子里面黑白球的比例, 而是闭着眼睛摸出一个 (或好几个) 球, 观察这些取出来的球的颜色之后, 那么我们可以就此对袋子里面的黑白球的比例作出什么样的推测。(现实世界本身就是不确定的, 人类的观察能力是有局限性的)

# 例子



•一所学校里面有 60% 的男生，40% 的女生。男生总是穿长裤，女生则一半穿长裤一半穿裙子。有了这些信息之后我们可以容易地计算“随机选取一个学生，他（她）穿长裤的概率和穿裙子的概率是多大”，这个就是前面说的“正向概率”的计算。然而，假设你走在校园中，迎面走来一个穿长裤的学生（很不幸的是你高度近似，你只看得见他（她）穿的是否长裤，而无法确定他（她）的性别），你能够推断出他（她）是男生的概率是多大吗？

•解释：

•假设学校里面人的总数是  $U$  个。

•60% 的男生都穿长裤，于是我们得到了  $U * 60% * 100%$  个穿长裤的（男生）

•40% 的女生里面又有一半（50%）是穿长裤的，于是我们又得到了  $U * 40% * 50%$  个穿长裤的（女生）。

•加起来一共  $0.6U + 0.2U = 0.8U$  个穿长裤的，其中有  $0.2U$  个女生。

•该问题为一个**单特征的二分类问题**

- 条件概率公式

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

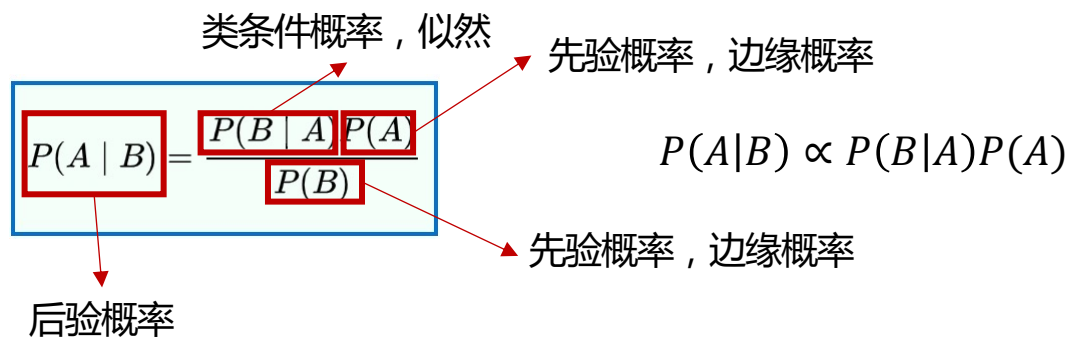
- 条件概率是指事件A在另外一个事件B已经发生条件下的发生概率

- 全概率公式

$$\begin{aligned} P(B) &= P(B \cap A_1) + P(B \cap A_2) + \dots + P(B \cap A_n) \\ &= P(B|A_1)P(A_1) + P(B|A_2)P(A_2) + \dots + P(B|A_n)P(A_n) \end{aligned}$$

- 若事件 $A_1, A_2, \dots, A_n$ 构成一个完备事件组（互不相容，其和为全集）且都有正概率

- 贝叶斯公式



The diagram shows the formula  $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$  with several annotations:

- A red box around the entire formula is labeled "后验概率" (Posterior probability).
- A red box around  $P(A|B)$  is labeled "后验概率" (Posterior probability).
- A red box around  $P(B|A)$  is labeled "类条件概率, 似然" (Class conditional probability, Likelihood).
- A red box around  $P(A)$  is labeled "先验概率, 边缘概率" (Prior probability, Marginal probability).
- A red box around  $P(B)$  is labeled "先验概率, 边缘概率" (Prior probability, Marginal probability).

To the right of the formula, the proportionality relationship is given:  $P(A|B) \propto P(B|A)P(A)$ .

- 其中A与B为事件且 $P(B) > 0$

# 朴素贝叶斯分类法

- 设输入样本 $x$ 有 $d$ 个特征，表示为 $x = [x_1, x_2, \dots, x_{d-1}, x_d]$
- 所属类别 $y$ 有 $K$ 个可能的取值，分别为 $c_1, c_2, \dots, c_K$
- 则根据贝叶斯公式，输入样本 $x$ 属于类别 $c_k$ 的概率为

$$\begin{aligned} P(y = c_k | x) &= \frac{P(x | y = c_k)P(y = c_k)}{P(x)} \\ &= \frac{P([x_1, x_2, x_3, \dots, x_d] | y = c_k)P(y = c_k)}{P(x)} \\ &= \frac{P(x_1 | y = c_k, x_2 | y = c_k, x_3 | y = c_k, \dots, x_d | y = c_k)P(y = c_k)}{P(x)} \end{aligned}$$

- 根据定义，**朴素贝叶斯分类法**假定样本各个特征是**相互条件独立的**（以类别为条件）
  - $P(X, Y | Z) = P(X | Z)P(Y | Z)$
- 在该假设前提下，可以运用贝叶斯定理进一步分解上述式子

- 分子部分可以分解如下

$$\begin{aligned} &P(x_1 | y = c_k, x_2 | y = c_k, x_3 | y = c_k, \dots, x_d | y = c_k)P(y = c_k) \\ &= P(x_1 | y = c_k) * P(x_2 | y = c_k) * P(x_3 | y = c_k) * \dots * P(x_d | y = c_k)P(y = c_k) \\ &= P(y = c_k) \prod_{i=1}^d P(x_i | y = c_k) \end{aligned}$$

- 分解后各部分的概率值比分解前的联合概率更容易从数据求得
- 朴素贝叶斯方法**独立性假设**存在误差甚至错误



•分母部分可以计算如下

$$\begin{aligned} P(\mathbf{x}) &= \sum_{k=1}^K P(\mathbf{x}, y = c_k) \\ &= \sum_{k=1}^K P(\mathbf{x} | y = c_k) P(y = c_k) \quad (\text{全概率公式}) \\ &= \sum_{k=1}^K P([x_1, x_2, x_3, \dots, x_d] | y = c_k) P(y = c_k) \\ &= \sum_{k=1}^K P(x_1 | y = c_k, x_2 | y = c_k, x_3 | y = c_k, \dots, x_d | y = c_k) P(y = c_k) \\ &= \sum_{k=1}^K P(x_1 | y = c_k) * P(x_2 | y = c_k) * P(x_3 | y = c_k) * \dots * P(x_d | y = c_k) * P(y = c_k) \quad (\text{特征条件独立性假设}) \\ &= \sum_{k=1}^K P(y = c_k) * \prod_{i=1}^d P(x_i | y = c_k) \end{aligned}$$

•最终样本x所属的类别的概率公式

$$P(y = c_k | x) = \frac{P(y = c_k) * \prod_{i=1}^d P(x_i | y = c_k)}{\sum_{k=1}^K P(y = c_k) * \prod_{i=1}^d P(x_i | y = c_k)}$$

# 房屋好卖预测案例



样本	户型	居室数	所在楼层	是否好卖
训练样本1	大户型	4	低楼层	是
训练样本2	大户型	3	高楼层	是
训练样本3	中户型	3	中楼层	是
训练样本4	中户型	2	高楼层	是
训练样本5	大户型	4	低楼层	否
训练样本6	大户型	3	中楼层	否
训练样本7	中户型	3	中楼层	是
训练样本8	中户型	2	高楼层	否
训练样本9	小户型	2	高楼层	是
训练样本10	小户型	2	低楼层	否
测试样本1	中户型	3	顶楼	否
测试样本2	中户型	2	高楼层	否
测试样本3	大户型	5	高楼层	是
测试样本4	小户型	3	高楼层	是
测试样本5	小户型	2	低楼层	否

共有10个训练样本，每个样本有3个特征：面积、居室数、所在楼层。面积的取值是大户型、中户型、小户型；居室数的取值是2、3、4；所在楼层的取值是低楼层、中楼层、高楼层；是否好卖是类别，只有两个取值：是、否

# 房屋好卖预测案例(演示)



- 预测一个新房屋是否好卖，其特征是小户型、3居室、高楼层

$$P(y = \text{好卖} | x = [\text{小户型}, \text{三居室}, \text{高楼层}])$$

$$= \frac{P(y = \text{好卖}) * P(x_1 = \text{小户型} | y = \text{好卖}) * P(x_2 = \text{三居室} | y = \text{好卖}) * P(x_3 = \text{高楼层} | y = \text{好卖})}{P(x = [\text{小户型}, \text{三居室}, \text{高楼层}])}$$

$$P(y = \text{不好卖} | x = [\text{小户型}, \text{三居室}, \text{高楼层}])$$

$$= \frac{P(y = \text{不好卖}) * P(x_1 = \text{小户型} | y = \text{不好卖}) * P(x_2 = \text{三居室} | y = \text{不好卖}) * P(x_3 = \text{高楼层} | y = \text{不好卖})}{P(x = [\text{小户型}, \text{三居室}, \text{高楼层}])}$$

$$P(y = \text{好卖}) = \frac{6}{10} \quad P(y = \text{不好卖}) = \frac{4}{10} \quad P(x_1 = \text{小户型} | y = \text{好卖}) = \frac{1}{6} \quad P(x_2 = \text{三居室} | y = \text{好卖}) = \frac{3}{6}$$

$$P(x_3 = \text{高楼层} | y = \text{好卖}) = \frac{3}{6} \quad P(x_1 = \text{小户型} | y = \text{不好卖}) = \frac{1}{4} \quad P(x_2 = \text{三居室} | y = \text{不好卖}) = \frac{1}{4} \quad P(x_3 = \text{高楼层} | y = \text{不好卖}) = \frac{1}{4}$$

$$P(y = \text{好卖} | x = [\text{小户型}, \text{三居室}, \text{高楼层}]) = \frac{\frac{6}{10} \times \frac{1}{6} \times \frac{3}{6} \times \frac{3}{6}}{P(x)} = \frac{\frac{40}{10}}{P(x)}$$

$$P(y = \text{不好卖} | x = [\text{小户型}, \text{三居室}, \text{高楼层}]) = \frac{\frac{4}{10} \times \frac{1}{4} \times \frac{1}{4} \times \frac{1}{4}}{P(x)} = \frac{\frac{160}{10}}{P(x)}$$

- 第一步，建立样本特征矩阵，进行特征转换
  - 户型特征转换为“户型=小户型”、“户型=中户型”、“户型=大户型”三个新特征
  - 居室数特征转换为“居室数=2”、“居室数=3”、“居室数=4”、“居室数=5”四个新特征
  - 所在楼层特征转换为“楼层=低楼层”、“楼层=中楼层”、“楼层=高楼层”、“楼层=顶楼”四个新特征
  - 总共得到11个新特征
- 表中的第一个训练样本，转换后的特征向量为 $[0,0,1, 0,0,1,0, 1,0,0,0]$ ，其中第一个1表示“户型=大户型”，第二个1表示“居室数=4”，第三个1表示“楼层=低楼层”
- 第二个训练样本和第三个训练样本转换后的特征值分别为： $[0,0,1, 0,1,0,0, 0,0,1,0]$ 和 $[0,1,0, 0,1,0,0, 0,1,0,0]$

- 第二步，计算先验概率

- 不失一般性，对于任何一个特征fea和一个类别值 $c_1$ ，设fea值为1且属于类别 $c_1$ 的训练样本数量为fea\_samples\_c1；属于类别 $c_1$ 的训练样本数量为c1\_samples；则特征fea对类别 $c_1$ 的条件概率可简单计算为：

$$P(fea = 1 | c = c_1) = \frac{fea\_samples\_c1}{c1\_samples}$$

- 对于一个类别c1，该类别出现的先验概率为：

$$P(c = c_1) = \frac{c1\_samples}{n\_samples}$$

## •第三步，平滑处理

- 以上先验概率的计算存在一个缺陷：即有些特征值没有在训练样本中出现过，这些特征的先验概率就没有被计算
- “居室数=五居室”、“楼层=顶楼”这两个特征，没有在训练数据中出现过，却在测试数据中出现了
- 特征值没有在训练数据中出现是因为训练数据量不够，没有覆盖所有的情况，并非是因为这些特征值出现的概率是零
- 常用的方法是“平滑处理 ( Smoothing ) ”

$$P(fea = 1 | c = c_1) = \frac{fea\_samples\_c1 + \alpha}{c1\_samples + \lambda * \alpha}$$

- 当  $\alpha \in (0,1)$  时，被称为利德斯通平滑 ( Lidstone smoothing )
- $\lambda$ 的取值通常为特征的数量，如本例中可设 $\lambda=11$ ，这是为了保证所有特征对类别 $c_1$ 的先验条件概率之和为1

$$\sum_{i=1}^d P(fea_i = 1 | c = c_1) = 1$$

## •第四步，对数化处理

- 以上对于先验概率的计算还存在另一个缺陷：分子是多个概率值（0~1之间）连乘，会导致结果是一个非常小的小数，最后甚至会超出计算机浮点数的范围而无法继续计算。

解决的方法是对概率取对数，即进行对数化处理，大大改善浮点运算的性能

## •进行对数化处理

$$\begin{aligned} & \log \left[ P(y = c_k) * \prod_{i=1}^d P(x_i | y = c_k) \right] \\ &= \log P(y = c_k) + \sum_{i=1}^d \log P(x_i | y = c_k) \end{aligned}$$

$$P(y = c_k | x) = \frac{e^{\log P(y=c_k) + \sum_{i=1}^d \log P(x_i|y=c_k)}}{\sum_{k=1}^K e^{\log P(y=c_k) + \sum_{i=1}^d \log P(x_i|y=c_k)}}$$

- 需要计算样本 $x$ 属于各个类别的概率，然后将样本 $x$ 分类为概率最大的类别
  - 由于最终比较的是属于各个类别的概率大小，而对于所有的类别，分母部分是固定不变的，因此只需要计算并比较分子部分就可以
  - 对于任何一个测试样本，由于其特征值已确定，并且各个特征的 $\log P(fea_i = 1 | c = c_k)$  和  $\log P(c = c_k)$ 都已计算出来
- 以测试样本1为例
  - 此时 $x = \text{np.array}([0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1])$
  - 设 $\text{prob\_fea0}$ 为一个11元素的向量，其中各个元素表示对应的特征对类别0的条件概率的对数值； $\text{prob\_c0}$ 是类别0在整个训练样本集中所占的比例，即类别0出现的先验概率
  - 设 $\text{prob\_fea1}$ 为一个11元素的向量，各元素表示对应特征对类别1的条件概率的对数值； $\text{prob\_c1}$ 是类别1在整个训练样本集中所占的比例，即类别1出现的先验概率
  - $x$ 属于类别0的概率计算公式的分子部分是： $x \cdot \text{prob\_fea0} + \text{prob\_c0}$ ， $x$ 属于类别1的概率计算公式的分子部分是： $x \cdot \text{prob\_fea1} + \text{prob\_c1}$ ；比较两者大小即可确定所属类别



## •运行代码6.1进行演示

- trainX和trainY分别是训练数据的特征矩阵和类别。其中，trainX是一个10行11列的二维矩阵，每行是一个样本，每列是一个转换后的新特征；trainY是一个向量，共有10个元素，每个元素对应一个样本所属类别
- testX和testY分别是测试数据的特征矩阵和类别
- train()函数输入训练数据，训练一个朴素贝叶斯模型，其实质是计算各个特征对各个类别的先验条件概率以及计算各个类别出现的概率
- classifiy()函数输入测试数据，根据先验概率进行类别的预测

代码6.1自定义编码实现朴素贝叶斯分类求解房屋好卖问题

```
trainX = np.array ([ [ 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0 ],  
                     [ 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0 ],  
                     [ 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0 ],  
                     [ 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0 ],  
                     [ 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0 ],  
                     [ 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0 ],  
                     [ 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0 ],  
                     [ 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0 ],  
                     [ 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0 ],
```

运行结果：

[ True True True True False]

其中True表示类别1，False表示类别0，与实际测试数据类别testY相比，错了2个，对了3个，正确率（Accuracy）是60%，精准率（Precision）是50%，召回率（Recall）是100%

- sklearn.naive\_bayes.MultinomialNB类

- 构造方法

- model = MultinomialNB ( alpha = 1.0, fit\_prior = True, class\_prior = None )

- alpha :拉普拉斯或利德斯通平滑的参数 $\lambda$  , 默认为1.0

- fit\_prior :是否学习先验概率 $P(Y=c)$  , 默认为True

- class\_prior : 形似数组的结构 , 结构为(n\_classes, ) , 默认为None

- 主要方法和属性

- fit(x, y) : 用训练数据x和y拟合朴素贝叶斯模型。

- predict(x) : 用拟合好的模型预测测试数据x的标签。

- class\_log\_prior\_ : 各类的对数概率值 , 可使用np.exp()函数转化为概率值。

- feature\_log\_prior\_ : 各特征的对数概率值 , 可使用np.exp()函数转化为概率值

# 求解步骤与编码实现



•演示代码6.2，使用MultinomialNB类进行求解房屋是否好卖

```
import numpy as np
from sklearn.naive_bayes import MultinomialNB #导入朴素贝叶斯类
trainX = np.array ( [ [ 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0 ],
                      [ 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0 ],
                      [ 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0 ],
                      [ 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0 ],
                      [ 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0 ],
                      [ 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0 ],
                      [ 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0 ],
                      [ 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0 ],
                      [ 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0 ],
                      [ 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0 ] ] )
trainY = np.array ( [ 1, 1, 1, 1, 0, 0, 1, 0, 1, 0 ] )
model = MultinomialNB ( ) #创建朴素贝叶斯对象
model.fit ( trainX, trainY ) #训练模型
```

```
testX = np.array ( [ [ 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1 ],
                     [ 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0 ],
                     [ 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0 ],
                     [ 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0 ],
                     [ 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0 ] ] )
testY = np.array ( [ 0, 0, 1, 1, 0 ] )
print ( model.predict ( testX ) ) #预测并输出结果
```

运行结果：

[1 1 0 1 0]

正确率 ( Accuracy ) 是40%，精准率 ( Precision ) 是33%，召回率 ( Recall ) 是50%

# 连续型特征值朴素贝叶斯



- 特征值存在连续值的情况，如下表

身高（厘米）	体重（公斤）	脚的尺寸（厘米）	性别
183	82.10	30.48	男
180	86.02	27.94	男
170	77.15	30.48	男
180	75.26	25.4	男
153	45.00	15.24	女
168	68.50	20.32	女
165	58.80	17.78	女
175	68.00	22.86	女

- 身高、体重、脚的尺寸是三个特征，取值都是连续型数值
- 对连续型特征进行计数是没有意义的，比如统计体重为80.00、80.01、80.02、80.03...的人所占的比重
- 对于连续型特征，无法像离散型特征一样使用
- 利用概率密度函数来代替概率，即通过概率密度函数计算某一点的概率密

$$\frac{fea\_samples\_c1}{c1\_samples}$$

- 朴素贝叶斯公式则变为

$$P(y = c_k | \mathbf{x}) = \frac{pdf(x_1 | y = c_k) * pdf(x_2 | y = c_k) * \dots * pdf(x_d | y = c_k) * P(y = c_k)}{evidence}$$

- 与离散情况下的贝叶斯公式相似，分母evidence表示全概率，一般不必计算，只需比较不同分类中分子值的大小即可判断出所属的类别
- 即使是连续型随机变量，每种类别所占的比重仍然是可以计算的。例如，可以从表训练数据中计算得到两个类别（男、女）的概率为

$$P(y = \text{男}) = P(y = \text{女}) = 0.5$$

- 概率密度函数pdf要怎么确定和计算是关键问题，在大部分情况下，可以假定随机变量对类别的条件概率服从正态分布

$$P(x_i | y = c_k) = \frac{1}{\sigma_{i,k} \sqrt{2\pi}} * e^{-\frac{(x_i - \mu_{i,k})^2}{2\sigma_{i,k}^2}}$$

# 连续型特征值朴素贝叶斯



- 设类别“男”对应 $c_1$ ，类别“女”对应 $c_2$ ；身高、体重、脚的尺寸三个特征分别设为 $x_1$ 、 $x_2$ 、 $x_3$

$$\mu_{1,1} = \frac{183+180+170+180}{4} = 178.25$$

$$\mu_{2,1} = \frac{82.10+86.02+77.15+75.26}{4} = 80.13$$

$$\sigma_{1,1}^2 = \frac{(183-178.25)^2 + (180-178.25)^2 + (170-178.25)^2 + (180-178.25)^2}{4-1} = 32.25$$

- 使用正态分布概率密度函数计算后验概率

- 计算身高、体重、脚尺寸分别为175、70、28的测试样本的类别

$$pdf(x_1 | y = c_1) = \frac{1}{\sigma_{1,1}\sqrt{2\pi}} * e^{-\frac{(x-\mu_{1,1})^2}{2\sigma_{1,1}^2}} = \frac{1}{5.68 * \sqrt{2\pi}} * e^{-\frac{(175-178.25)^2}{2*5.68^2}} = 0.06$$

$$pdf(x_2 | y = c_1) = 0.009 \quad pdf(x_3 | y = c_1) = 0.1595 \quad pdf(x_2 | y = c_2) = 0.0241$$

$$pdf(x_3 | y = c_2) = 0.0029 \quad pdf(x_1 | y = c_2) = 0.0247$$

$$P(y = c_1) = 0.5$$

$$P(y = c_2) = 0.5$$

$$P(y = c_1 | x) = \frac{0.06 * 0.009 * 0.1595 * 0.5}{evidence} = \frac{4.3065e-06}{evidence} \quad P(y = c_2 | x) = \frac{0.0247 * 0.0241 * 0.0029 * 0.5}{evidence} = \frac{8.63e-07}{evidence}$$

# 基于GaussianNB类实现



#代码6.3 使用GaussianNB类实现连续型特征朴素贝叶斯分类求解性别分类

```
import numpy as np
```

```
from sklearn.naive_bayes import GaussianNB #导入朴素贝叶斯类
```

```
#训练数据
```

```
trainX = np.array ( [ [ 183, 82.10, 30.48 ],  
                      [ 180, 86.02, 27.94 ],  
                      [ 170, 77.15, 30.48 ],  
                      [ 180, 75.26, 25.40 ],  
                      [ 153, 45.00, 15.24 ],  
                      [ 168, 68.50, 20.32 ],  
                      [ 165, 58.80, 17.78 ],  
                      [ 175, 68.00, 22.86 ] ] )
```

```
trainY = np.array ( [ 0, 0, 0, 0, 1, 1, 1, 1 ] )
```

```
model = GaussianNB ( ) #创建高斯朴素贝叶斯对象
```

```
model.fit ( trainX, trainY ) #训练模型
```

```
testX = np.array ( [ [ 175, 70, 28 ] ] ) #测试数据
```

```
print ( model.predict ( testX ) ) #预测并输出结果
```

- 定义了训练数据变量trainX，共8行3列，每一行代表一个训练样本，每一类代表一个特征属性；定义了训练数据的分类标签trainY，共8个元素，每个元素代表对应样本的类别，其中0表示男性，1表示女性。
- 创建了GaussianNB对象model，并调用model.fit()函数根据输入训练数据计算朴素贝叶斯模型的参数。
- 创建测试数据testX，调用model.predict()函数进行预测，最后输出预测结果0，表示男性类别

- 本章介绍了朴素贝叶斯分类法的定义与实现包括针对离散型特征的朴素贝叶斯分类和针对连续型特征的朴素贝叶斯分类。
- 通过本章的学习，读者对朴素贝叶斯分类法有了一个基本的了解，并掌握使用Python语言实现处理离散型和连续型特征的朴素贝叶斯分类的基本方法



# 十大数据挖掘算法（2008）

---



- K-means
- kNN
- Naïve Bayes
- SVM
- Aprior
- EM
- PageRank
- CART
- C4.5
- AdaBoost