1. （必填）自己提出的问题的理解（罗列全部）：
2. **提出的问题1：**

Navie Bayes和Logistic回归区别是什么：

讨论后的理解：

前者是生成式模型，后者是判别式模型，二者的区别就是生成式模型与判别式模型的区别。

1）首先，Navie Bayes通过已知样本求得先验概率P(Y), 及条件概率P(X|Y), 对于给定的实例，计算联合概率，进而求出后验概率。也就是说，它尝试去找到底这个数据是怎么生成的（产生的），然后再进行分类。哪个类别最有可能产生这个信号，就属于那个类别。

优点：样本容量增加时，收敛更快；隐变量存在时也可适用。

缺点：时间长；需要样本多；浪费计算资源

2）相比之下，Logistic回归不关心样本中类别的比例及类别下出现特征的概率，它直接给出预测模型的式子。设每个特征都有一个权重，训练样本数据更新权重w，得出最终表达式。梯度法。

优点：直接预测往往准确率更高；简化问题；可以反应数据的分布情况，类别的差异特征；适用于较多类别的识别。

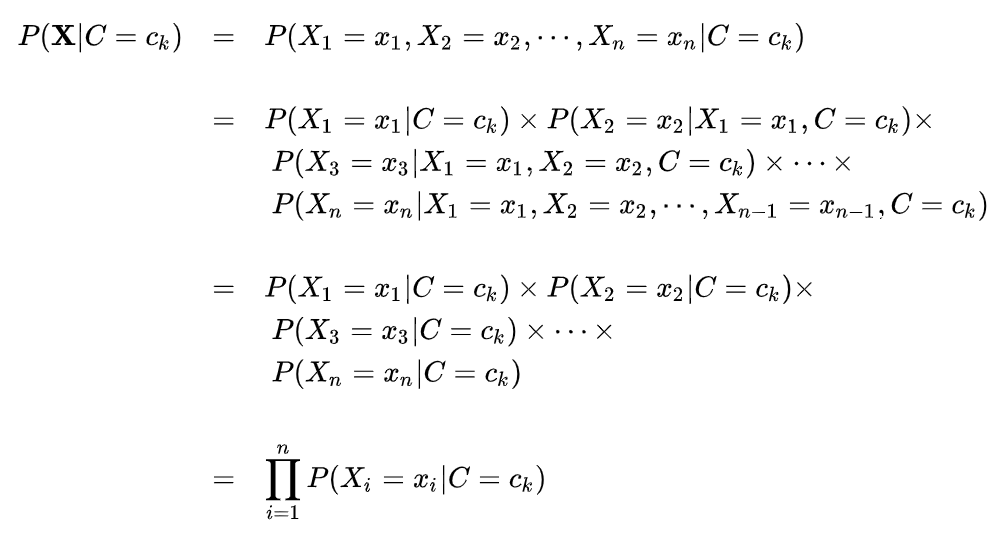
缺点：收敛慢；不适用于有隐变量的情况。

1. **提出的问题2：**

贝叶斯和朴素贝叶斯的区别？

讨论后的理解：

朴素贝叶斯分类器（Naïve Bayes classifiers）代表着一类应用贝叶斯定理的分类器的总称。朴素（naive）在这里有着特殊的含义、代表着一个非常强的假设。在朴素贝叶斯中，仅仅假设特征之间满足条件独立性，而非一般的独立性。



待估计的参数从2\*（2\*\*n - 1）个直接减少到2n个。这大大简化了对样本空间的要求以及求解的计算量，使得朴素贝叶斯算法非常简单。条件独立性的假设便是“朴素”一词的来源。因此，朴素贝叶斯通常也被称为简单贝叶斯（simple Bayes）或独立贝叶斯（independence Bayes）

1. （必填）别人提出的问题的理解（选择几个问题罗列，并给出理解）：
2. **问题3**：

为什么说朴素贝叶斯是稳定的算法？

自己的理解：

1)对于分类任务来说，只要各类别的条件概率排序正确、无需精准概率值即可导致正确分类；

2)如果属性间依赖对所有类别影响相同，或依赖关系的影响能相互抵消，则属性条件独立性假设在降低计算开销的同时不会对性能产生负面影响。

1. **问题4：**

朴素贝叶斯的主要缺点

自己的理解：

理论上，朴素贝叶斯模型与其他分类方法相比具有最小的误差率。但是实际上并非总是如此，这是因为朴素贝叶斯模型假设属性之间相互独立，这个假设在实际应用中往往是不成立的，在属性个数比较多或者属性之间相关性较大时，分类效果不好。而在属性相关性较小时，朴素贝叶斯性能最为良好。对于这一点，有半朴素贝叶斯之类的算法通过考虑部分关联性适度改进。

需要知道先验概率，且先验概率很多时候取决于假设，假设的模型可以有很多种，因此在某些时候会由于假设的先验模型的原因导致预测效果不佳。

由于我们是通过先验和数据来决定后验的概率从而决定分类，所以分类决策存在一定的错误率。

对输入数据的表达形式很敏感。

1. **问题5：**

几种朴素贝叶斯算法，高斯朴素贝叶斯（GaussianNB）、多项式朴素贝叶斯（MultinomialNB）和伯努利朴素贝叶斯（BernoulliNB）分别适用于什么场景：

讨论后的理解：

高斯朴素贝叶斯：特征变量是连续变量，符合高斯分布，比如说人的身高，物体的长度。

多项式朴素贝叶斯：特征变量是离散变量，符合多项分布，在文档分类中特征变量体在一个单词出现的次数，或者是单词的 TF-IDF 值等。多项式朴素贝叶斯实际上符合多项式分布，不会存在负数，所以传入输入的时候，可以使用MinMaxScaler进行归一化。

伯努利朴素贝叶斯：特征变量是布尔变量，符合 0/1 分布，在文档分类中特征是单词是否出现。伯努利朴素贝叶斯是以文件为粒度，如果该单词在某文件中出现了即为1，否则为0。而多项式朴素贝叶斯是以单词为粒度，会计算在某个文件中的具体次数。而高斯朴素贝叶斯适合处理特征变量是连续变量，且符合正态分布（高斯分布）的情况。比如身高、体重这种自然界的现象就比较适合用高斯朴素贝叶斯来处理。而文本分类是使用多项式朴素贝叶斯或者伯努利朴素贝叶斯。

三、（必填）读书计划

1、本周完成的内容章节：统计学习方法（第四章）

2、下周计划：统计学习方法（第五章）

四、（选做）读书摘要及理解或伪代码的具体实现（读书摘要、伪代码的具体实现代码等可以写到这个部分）

1、读书摘要及理解（选做）

1）贝叶斯方法

贝叶斯方法是以贝叶斯原理为基础，使用概率统计的知识对样本数据集进行分类。由于其有着坚实的数学基础，贝叶斯分类算法的误判率是很低的。贝叶斯方法的特点是结合先验概率和后验概率，即避免了只使用先验概率的主管偏见，也避免了单独使用样本信息的过拟合现象。贝叶斯分类算法在数据集较大的情况下表现出较高的准确率，同时算法本身也比较简单。

2）朴素贝叶斯算法

朴素贝叶斯算法（Naive Bayesian）是应用最为广泛的分类算法之一。朴素贝叶斯方法是在贝叶斯算法的基础上进行了相应的简化，即假定给定目标值时属性之间相互条件独立。也就是说没有哪个属性变量对于决策结果来说占有着较大的比重，也没有哪个属性变量对于决策结果占有着较小的比重。虽然这个简化方式在一定程度上降低了贝叶斯分类算法的分类效果，但是在实际的应用场景中，极大地简化了贝叶斯方法的复杂性。

2、伪代码的具体实现(选做)

from sklearn import datasets, cross\_validation, naive\_bayes

import matplotlib.pyplot as plt

# 可视化手写识别数据集Digit Dataset

def show\_digits():

digits = datasets.load\_digits()

fig = plt.figure()

for i in range(20):

ax = fig.add\_subplot(4, 5, i+1)

ax.imshow(digits.images[i], cmap = plt.cm.gray\_r, interpolation='nearest')

plt.show()

show\_digits()

# 加载Digit数据集

def load\_data():

digits = datasets.load\_digits()

return cross\_validation.train\_test\_split(digits.data, digits.target,

test\_size = 0.25, random\_state = 0)

def test\_GaussianNB(\*data):

X\_train, X\_test, y\_train, y\_test = data

cls = naive\_bayes.GaussianNB()

cls.fit(X\_train, y\_train)

print('GaussianNB Classifier')

print('Training Score: %.2f' % cls.score(X\_train, y\_train))

print('Test Score: %.2f' % cls.score(X\_test, y\_test))

X\_train, X\_test, y\_train, y\_test = load\_data()

test\_GaussianNB(X\_train, X\_test, y\_train, y\_test)

def test\_MultinomialNB(\*data):

X\_train, X\_test, y\_train, y\_test = data

cls = naive\_bayes.MultinomialNB()

cls.fit(X\_train, y\_train)

print('MultinomialNB Classifier')

print('Training Score: %.2f' % cls.score(X\_train, y\_train))

print('Test Score: %.2f' % cls.score(X\_test, y\_test))

X\_train, X\_test, y\_train, y\_test = load\_data()

test\_MultinomialNB(X\_train, X\_test, y\_train, y\_test)

def test\_BernoulliNB(\*data):

X\_train, X\_test, y\_train, y\_test = data

cls = naive\_bayes.BernoulliNB()

cls.fit(X\_train, y\_train)

print('BernoulliNB Classifier')

print('Training Score: %.2f' % cls.score(X\_train, y\_train))

print('Test Score: %.2f' % cls.score(X\_test, y\_test))

X\_train, X\_test, y\_train, y\_test = load\_data()

test\_BernoulliNB(X\_train, X\_test, y\_train, y\_test)