朴素贝叶斯

此代码来自 阿里天池：https://tianchi.aliyun.com/notebook/192338 推荐自学

1 朴素贝叶斯

1.1 贝叶斯公式

贝叶斯公式是英国数学家提出的一个数据公式:

$$ p(A|B)=\frac{p(A,B)}{p(B)}=\frac{p(B|A) \cdot p(A)}{\sum_{a \in \mathcal{F}_A}p(B|a) \cdot p(a)} $$

- p(A,B)：表示事件A和事件B同时发生的概率。
- p(B)：表示事件B发生的概率，叫做先验概率；
- p(A)：表示事件A发生的概率。
- p(A|B)：表示当事件B发生的条件下，事件A发生的概率叫做后验概率。
- p(B|A)：表示当事件A发生的条件下，事件B发生的概率。

1.2 朴素贝叶斯算法

朴素贝叶斯法 = 贝叶斯定理 + 特征条件独立。

贝叶斯法一定要计算两个概率：条件概率：$P(X^{(i)}=x^{(i)}|Y=c_k)$和类c_k的先验概率：$P(Y=c_k)$。

$$P(Y=c_k|X=x)=\frac{\prod_{i=1}^{n} P(x_i | Y=c_k)P(Y=c_k)}{\sum_{k}\prod_{i=1}^{n} P(x_i | Y=c_k)P(Y=c_k)}$$
我们为了选择后验概率最大的结果，进行概率的比较，由于分母一致，这里直接去掉分母，得到最后的计算公式。

$$ y=arg\ max_{c_k}P(Y=c_k)\prod_{j}P(X^{(j)}=x^{(j)}|Y=c_k) $$

2 朴素贝叶斯-鸢尾花数据集分类

Step1: 库函数导入

In [1]:
```
import warnings
warnings.filterwarnings('ignore')
import numpy as np
# 加载莺尾花数据集
from sklearn import datasets
# 导入高斯朴素贝叶斯分类器
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
```

Step2: 数据导入&分析

我们需要计算两个概率分别是：

- 条件概率：$P(X^{(i)}=x^{(i)}|Y=c_k)$
- 类c_k的先验概率：$P(Y=c_k)$。

通过分析发现训练数据是数值类型的数据，这里假设每个特征服从高斯分布，因此我们选择高斯朴素贝叶斯来进行分类计算。

In [2]:
```
X, y = datasets.load_iris(return_X_y=True)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=2023)
```

Step3: 模型训练

In [3]:
```
# 使用高斯朴素贝叶斯进行计算
clf = GaussianNB(var_smoothing=1e-8)
clf.fit(X_train, y_train)
```

Out[3]:
```
GaussianNB(var_smoothing=1e-08)
```

Step4: 模型预测

In [4]:
```
# 评估
y_pred = clf.predict(X_test)
acc = np.sum(y_test == y_pred) / X_test.shape[0]
print("Test Acc : %.3f" % acc)
```

```
# 预测
y_proba = clf.predict_proba(X_test[:1])
print(clf.predict(X_test[:1]))
print("预计的概率值:", y_proba)
```

```
Test Acc : 1.000
[2]
预计的概率值: [[2.57698140e-197 1.75975179e-005 9.99982402e-001]]
```

Step5: 原理简析

高斯朴素贝叶斯假设每个特征都服从高斯分布，我们把一个随机变量X服从数学期望为μ，方差为σ^2的数据分布称为高斯分布。对于每个特征我们一般使用平均值来估计μ和使用所有特征的方差估计σ^2。

$$ P(X^{(i)}=x^{(i)}|Y=c_k) = \frac{1}{\sqrt{2\pi\sigma^2_y}} \exp\left(-\frac{(x^{(i)} - \mu_{c_k})^2}{2\sigma^2_{c_k}}\right) $$
从上述例子中的预测结果中，我们可以看到类别2对应的后验概率值最大，所以我们认为类别2是最优的结果。

1.3 朴素贝叶斯-新闻分类

In [5]:
```
from sklearn.datasets import fetch_20newsgroups
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report


news=fetch_20newsgroups(subset='all')
```

如果存在下载速度慢或者出现网络报错问题，

请参照博客教程自行解决。

In [ ]:
```
# 进行数据分割
x_train, x_test, y_train, y_test = train_test_split(news.data, news.target, test_size=0.25)
# 对数据集进行特征抽取
tf = TfidfVectorizer()
# 以训练集当中的词的列表进行每篇文章重要性统计['a','b','c','d']
x_train = tf.fit_transform(x_train)
x_test = tf.transform(x_test)
```

```
# 进行朴素贝叶斯算法的预测
mlt = MultinomialNB(alpha=1.0)
print(x_train)
```

In [7]:
```
mlt.fit(x_train, y_train)   #MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)
y_predict = mlt.predict(x_test)
print("预测的文章类别为：", y_predict)
#预测的文章类别为：  [ 3 16  5 ...  0  5  8]
# 得出准确率
print("准确率为：", mlt.score(x_test, y_test))
#准确率为：  0.8414685908319185
print("每个类别的精确率和召回率：", classification_report(y_test, y_predict, target_names=news.target_names))
```

```
预测的文章类别为：  [10 10 17 ...  3 12 16]
准确率为：  0.8578098471986417
每个类别的精确率和召回率：                           precision    recall  f1-score   support

           alt.atheism       0.85      0.68      0.76       197
         comp.graphics       0.87      0.81      0.84       232
 comp.os.ms-windows.misc       0.88      0.86      0.87       242
comp.sys.ibm.pc.hardware       0.76      0.87      0.81       228
   comp.sys.mac.hardware       0.86      0.90      0.88       209
          comp.windows.x       0.91      0.85      0.88       242
            misc.forsale       0.93      0.63      0.75       244
               rec.autos       0.90      0.89      0.89       268
         rec.motorcycles       0.94      0.95      0.95       255
      rec.sport.baseball       0.98      0.95      0.96       266
        rec.sport.hockey       0.94      0.98      0.96       258
               sci.crypt       0.85      0.98      0.91       269
         sci.electronics       0.91      0.80      0.85       251
                 sci.med       0.96      0.88      0.92       257
               sci.space       0.90      0.95      0.92       248
  soc.religion.christian       0.53      1.00      0.69       239
      talk.politics.guns       0.74      0.99      0.85       214
   talk.politics.mideast       0.97      0.96      0.96       257
      talk.politics.misc       0.98      0.68      0.80       184
      talk.religion.misc       0.94      0.22      0.35       152

                accuracy                           0.86      4712
               macro avg       0.88      0.84      0.84      4712
            weighted avg       0.88      0.86      0.85      4712
```

4 朴素贝叶斯算法的优缺点

优点：

- （1）朴素贝叶斯模型发源于古典数学理论，有稳定的分类效率。
- （2） 分类准确度高，速度快
- （3）对缺失数据不太敏感，算法也比较简单，常用于文本分类，如垃圾邮件的分类，信用评估，钓鱼网站检测等。。

缺点：

- （1）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给定输出类别的情况下,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。
- （2）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，有时会由于假设的先验模型的原因导致预测效果不佳。
- （3）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。
- （4）对输入数据的表达形式很敏感。