EDU

CSDN学院　IT实战派

朴素贝叶斯模型

讲师：刘顺祥

课程目标

1. 理解朴素贝叶斯模型的思想和理论
2. 掌握几种常用的贝叶斯分类器
3. 朴素贝叶斯模型的应用实战

模型思想

该分类器的实现思想非常简单，即通过已知类别的训练数据集，计算样本的先验概率，然后利用贝叶斯概率公式测算未知类别样本属于某个类别的后验概率，最终以最大后验概率所对应的类别作为样本的预测值。

贝叶斯理论

条件概率

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

全概率公式

$$P(A) = \sum_{i=1}^{n} P(AB_i) = \sum_{i=1}^{n} P(B_i)P(A|B_i)$$

$$P(C_i|X) = \frac{P(C_iX)}{P(X)} = \frac{P(C_i)P(X|C_i)}{\sum_{i=1}^{k} P(C_i)\,P(X|C_i)}$$

贝叶斯理论

$$P(C_i|X) = \frac{P(C_iX)}{P(X)} = \frac{P(C_i)P(X|C_i)}{\sum_{i=1}^{k} P(C_i)\,P(X|C_i)}$$

其中，C_i表示样本所属的某个类别。对于上面的条件概率公式而言，样本最终属于哪个类别C_i，应该将计算所得的最大概率值$P(C_i|X)$对应的类别作为样本的最终分类，所以上式可以表示为：

$$y = f(X) = P(C_i|X) = argmax \frac{P(C_i)P(X|C_i)}{\sum_{i=1}^{k} P(C_i)\,P(X|C_i)}$$

贝叶斯理论

$$y = f(X) = P(C_i|X) = argmax \frac{P(C_i)P(X|C_i)}{\sum_{i=1}^{k} P(C_i)\,P(X|C_i)}$$

分母$P(X) = \sum_{i=1}^{k} P(C_i)\,P(X|C_i)$是一个常量，它与样本属于哪个类别没有直接关系，所以计算$P(C_i|X)$的最大值就转换成了计算分子的最大值，即$argmax\ P(C_i)P(X|C_i)$；

如果分子中的$P(C_i)$项未知的话，一般会假设每个类别出现的概率相等，只需计算$P(X|C_i)$的最大值，然而在绝大多数情况下，$P(C_i)$是已知的，它以训练数据集中类别C_i的频率作为先验概率，可以表示为N_{C_i}/N。

贝叶斯理论

假设数据集一共包含p个自变量，则X可以表示成$(x_1, x_2, \cdots, x_p)$，进而条件概率$P(X|C_i)$可以表示为：

$$P(X|C_i) = P(x_1, x_2, \cdots, x_p|C_i)$$

为了使分类器在计算过程中提高速度，提出了一个假设前提，即自变量是条件独立的（自变量之间不存在相关性），所以上面的计算公式可以重新改写为：

$$P(X|C_i) = P(x_1, x_2, \ldots, x_p|C_i) = P(x_1|C_i)P(x_2|C_i)\cdots P(x_p|C_i)$$

高斯贝叶斯分类器

如果数据集中的自变量X均为连续的数值型，则在计算$P(X|C_i)$时会假设自变量X服从高斯正态分布，所以自变量X的条件概率可以表示成：

$$P(x_j|C_i) = \frac{1}{\sqrt{2\pi}\sigma_{ji}} exp\left(-\frac{(x_j - \mu_{ji})^2}{2{\sigma_{ji}}^2}\right)$$

高斯贝叶斯分类器

Age	Income	Loan
23	8000	1
27	12000	1
25	6000	0
21	6500	0
32	15000	1
45	10000	1
18	4500	0
22	7500	1
23	6000	0
20	6500	0

假设某金融公司是否愿意给客户放贷会优先考虑两个因素，分别是年龄和收入。现在根据已知的数据信息考察一位新客户，他的年龄为24岁，并且收入为8500元，请问该公司是否愿意给客户放贷？

朴素贝叶斯模型

高斯贝叶斯分类器

Age	Income	Loan
23	8000	1
27	12000	1
25	6000	0
21	6500	0
32	15000	1
45	10000	1
18	4500	0
22	7500	1
23	6000	0
20	6500	0

（1）因变量各类别频率

$$P(loan = 0) = 5/10 = 0.5$$

$$P(loan = 1) = 5/10 = 0.5$$

（2）均值

$$\mu_{Age_0} = 21.40 \qquad \mu_{Age_1} = 29.8$$

$$\mu_{Income_0} = 5900 \qquad \mu_{Income_1} = 10500$$

（3）标准差

$$\sigma_{Age_0} = 2.42 \qquad \sigma_{Age_1} = 8.38$$

$$\sigma_{Income_0} = 734.85 \qquad \sigma_{Income_1} 2576.81$$

朴素贝叶斯模型

高斯贝叶斯分类器

Age	Income	Loan
23	8000	1
27	12000	1
25	6000	0
21	6500	0
32	15000	1
45	10000	1
18	4500	0
22	7500	1
23	6000	0
20	6500	0

（4）单变量条件概率

$$P(Age = 24|loan = 0) = \frac{1}{\sqrt{2\pi} \times 2.42} exp\left(-\frac{(24-21.4)^2}{2 \times 2.42^2}\right) = 0.0926$$

$$P(Age = 24|loan = 1) = \frac{1}{\sqrt{2\pi} \times 8.38} exp\left(-\frac{(24-29.8)^2}{2 \times 8.38^2}\right) = 0.0375$$

$$P(Income = 8500|loan = 0) = \frac{1}{\sqrt{2\pi} \times 734.85} exp\left(-\frac{(8500-5900)^2}{2 \times 734.85^2}\right) = 1.0384 \times 10^{-6}$$

$$P(Income = 8500|loan = 1) = \frac{1}{\sqrt{2\pi} \times 2576.81} exp\left(-\frac{(8500-10500)^2}{2 \times 2576.81^2}\right) = 1.1456 \times 10^{-4}$$

高斯贝叶斯分类器

Age	Income	Loan
23	8000	1
27	12000	1
25	6000	0
21	6500	0
32	15000	1
45	10000	1
18	4500	0
22	7500	1
23	6000	0
20	6500	0

（5）贝叶斯后验概率

$$P(loan = 0|Age = 24, Income = 8500)$$
$$= P(loan = 0) \times P(Age = 24|loan = 0) \times P(Income = 8500|loan = 0)$$
$$= 0.5 \times 0.0926 \times 1.0384 \times 10^{-6} = 4.8079 \times 10^{-8}$$
$$P(loan = 1|Age = 24, Income = 8500)$$
$$= P(loan = 1) \times P(Age = 24|loan = 1) \times P(Income = 8500|loan = 1)$$
$$= 0.5 \times 0.0375 \times 1.1456 \times 10^{-4} = 2.1479 \times 10^{-6}$$

经过上面的计算可知，当客户的年龄为24岁，并且收入为8500时，被预测为不放贷的概率是4.8079×10^{-8}，放贷的概率为2.1479×10^{-6}，所以根据$argmax\, P(C_i)P(X|C_i)$的原则，最终该金融公司决定给客户放贷。

多项式贝叶斯分类器

如果数据集中的自变量X均为离散型变量，在计算概率值$P(X|C_i)$时，会假设自变量X的条件概率满足多项式分布，故概率值$P(X|C_i)$的计算公式可以表示为：

$$P(x_j = x_{jk}|C_i) = \frac{N_{ik} + \alpha}{N_i + n\alpha}$$

其中，x_{jk}表示自变量x_j的取值；N_{ik}表示因变量为类别C_i时自变量x_j取x_{jk}的样本个数；N_i表示数据集中类别C_i的样本个数；α为平滑系数，用于防止概率值取0可能，通常将该值取为1，表示对概率值做拉普拉斯平滑；n表示因变量的类别个数。

朴素贝叶斯模型

多项式贝叶斯分类器

Ocuupation	Edu	Income	Meet
公务员	本科	中	1
公务员	本科	低	1
非公务员	本科	中	0
非公务员	本科	高	1
公务员	硕士	中	1
非公务员	本科	低	0
公务员	本科	高	1
非公务员	硕士	低	0
非公务员	硕士	中	0
非公务员	硕士	高	1

假设影响女孩是否参加相亲活动的重要因素有三个，分别是男孩的职业、受教育水平和收入状况；如果女孩参加相亲活动，则对应的Meet变量为1，否则为0。请问在给定的信息下，对于高收入的公务员，并且其学历为硕士的男生来说，女孩是否愿意参与他的相亲？

朴素贝叶斯模型

多项式贝叶斯分类器

Ocuupation	Edu	Income	Meet
公务员	本科	中	1
公务员	本科	低	1
非公务员	本科	中	0
非公务员	本科	高	1
公务员	硕士	中	1
非公务员	本科	低	0
公务员	本科	高	1
非公务员	硕士	低	0
非公务员	硕士	中	0
非公务员	硕士	高	1

（1）因变量各类别频率

$$P(Meet=0)=4/10=0.4$$
$$P(Meet=1)=6/10=0.6$$

（2）单变量条件概率

$$P(Occupation=\text{公务员}|Meet=0)=\frac{0+1}{4+2\times1}=\frac{1}{6}$$
$$P(Occupation=\text{公务员}|Meet=1)=\frac{4+1}{6+2\times1}=\frac{5}{8}$$
$$P(Edu=\text{硕士}|Meet=0)=\frac{2+1}{4+2\times1}=\frac{3}{6}$$
$$P(Edu=\text{硕士}|Meet=1)=\frac{2+1}{6+2\times1}=\frac{3}{8}$$
$$P(Income=\text{高}|Meet=0)=\frac{0+1}{4+2\times1}=\frac{1}{6}$$
$$P(Income=\text{高}|Meet=1)=\frac{3+1}{6+2\times1}=\frac{4}{8}$$

朴素贝叶斯模型

多项式贝叶斯分类器

Ocuupation	Edu	Income	Meet
公务员	本科	中	1
公务员	本科	低	1
非公务员	本科	中	0
非公务员	本科	高	1
公务员	硕士	中	1
非公务员	本科	低	0
公务员	本科	高	1
非公务员	硕士	低	0
非公务员	硕士	中	0
非公务员	硕士	高	1

（3）贝叶斯后验概率

$$P(Meet=0|Occupation=\text{公务员},Edu=\text{硕士},Income=\text{高})=\frac{4}{10}\times\frac{1}{6}\times\frac{3}{6}\times\frac{1}{6}=\frac{1}{180}$$

$$P(Meet=1|Occupation=\text{公务员},Edu=\text{硕士},Income=\text{高})=\frac{6}{10}\times\frac{5}{8}\times\frac{3}{8}\times\frac{4}{8}=\frac{18}{256}$$

时，女生愿意见面的概率约为0.0703、不愿意见面的概率约为0.0056。

所以根据$argmax\ P(C_i)P(X|C_i)$的原则，最终女生会选择参加这

伯努利贝叶斯分类器

当数据集中的自变量X均为0-1二元值时，通常会优先选择伯努利贝叶斯分类器。利用该分类器计算概率值$P(X|C_i)$时，会假设自变量X的条件概率满足伯努利分布，故概率值$P(X|C_i)$的计算公式可以表示为：

$$P(x_j|C_i) = px_j + (1-p)(1-x_j)$$

其中，x_j为第j个自变量，取值为0或1；p表示类别为C_i时自变量取1的概率，该概率值可以使用经验频率代替。

$$p = P(x_j = 1|C_i) = \frac{N_{x_j} + \alpha}{N_i + n\alpha}$$

其中，N_i表示类别C_i的样本个数；N_{x_j}表示在类别为C_i时，x_j变量取1的样本量；α为平滑系数，同样是为了避免概率为0而设置的；n为因变量中的类别个数。

朴素贝叶斯模型

伯努利贝叶斯分类器

x_1=推荐	x_2=给力	x_3=吐槽	x_4=还行	x_5=太烂	类别
1	1	0	0	0	0
1	0	0	1	0	0
1	1	0	1	0	0
1	0	1	1	0	1
1	1	1	0	1	1
0	0	1	0	1	1
0	0	0	0	1	1
0	1	1	0	1	1
0	0	1	0	1	1
0	1	0	0	0	0

假设对10条评论数据做分词处理后，得到如表12-5所示的文档词条矩阵，矩阵中含有5个词语和1个表示情感的结果，其中类别为0表示正面情绪，1表示负面情绪。如果一个用户的评论中仅包含“还行”一词，请问该用户的评论属于哪种情绪？

朴素贝叶斯模型

伯努利贝叶斯分类器

x_1=推荐	x_2=给力	x_3=吐槽	x_4=还行	x_5=太烂	类别
1	1	0	0	0	0
1	0	0	1	0	0
1	1	0	1	0	0
1	0	1	1	0	1
1	1	1	0	1	1
0	0	1	0	1	1
0	0	0	0	1	1
0	1	1	0	1	1
0	0	1	0	1	1
0	1	0	0	0	0

（1）因变量各类别频率

$$P(\text{类别}=0)=4/10=2/5$$
$$P(\text{类别}=1)=6/10=3/5$$

（2）单变量条件概率

$$P(x_1=0|\text{类别}=0)=(1+1)/(4+2)=1/3$$
$$P(x_1=0|\text{类别}=1)=(4+1)/(6+2)=5/8$$
$$P(x_2=0|\text{类别}=0)=(1+1)/(4+2)=1/3$$
$$P(x_2=0|\text{类别}=1)=(4+1)/(6+2)=5/8$$
$$P(x_3=0|\text{类别}=0)=(4+1)/(4+2)=5/6$$
$$P(x_3=0|\text{类别}=1)=(1+1)/(6+2)=1/4$$
$$P(x_4=1|\text{类别}=0)=(2+1)/(4+2)=1/2$$
$$P(x_4=1|\text{类别}=1)=(0+1)/(6+2)=1/8$$
$$P(x_5=0|\text{类别}=0)=(4+1)/(4+2)=5/6$$
$$P(x_5=0|\text{类别}=1)=(1+1)/(6+2)=1/4$$

朴素贝叶斯模型

伯努利贝叶斯分类器

x_1=推荐	x_2=给力	x_3=吐槽	x_4=还行	x_5=太烂	类别
1	1	0	0	0	0
1	0	0	1	0	0
1	1	0	1	0	0
1	0	1	1	0	1
1	1	1	0	1	1
0	0	1	0	1	1
0	0	0	0	1	1
0	1	1	0	1	1
0	0	1	0	1	1
0	1	0	0	0	0

（3）贝叶斯后验概率

$$P(\text{类别}=0|x_1=0,x_2=0,x_3=0,x_4=1,x_5=0)$$
$$=\frac{2}{5}\times\frac{1}{3}\times\frac{1}{3}\times\frac{5}{6}\times\frac{1}{2}\times\frac{5}{6}=\frac{5}{324}$$
$$P(\text{类别}=1|x_1=0,x_2=0,x_3=0,x_4=1,x_5=0)$$
$$=\frac{3}{5}\times\frac{5}{8}\times\frac{5}{8}\times\frac{1}{4}\times\frac{1}{8}\times\frac{1}{4}=\frac{3}{4096}$$

结果所示，当用户的评论中只含有“还行”一词时，计算该评论为正面情绪的概率约为0.015，评论为负面情绪的概率约为0.00073，故根据贝叶斯后验概率最大原则将该评论预判为正面情绪。

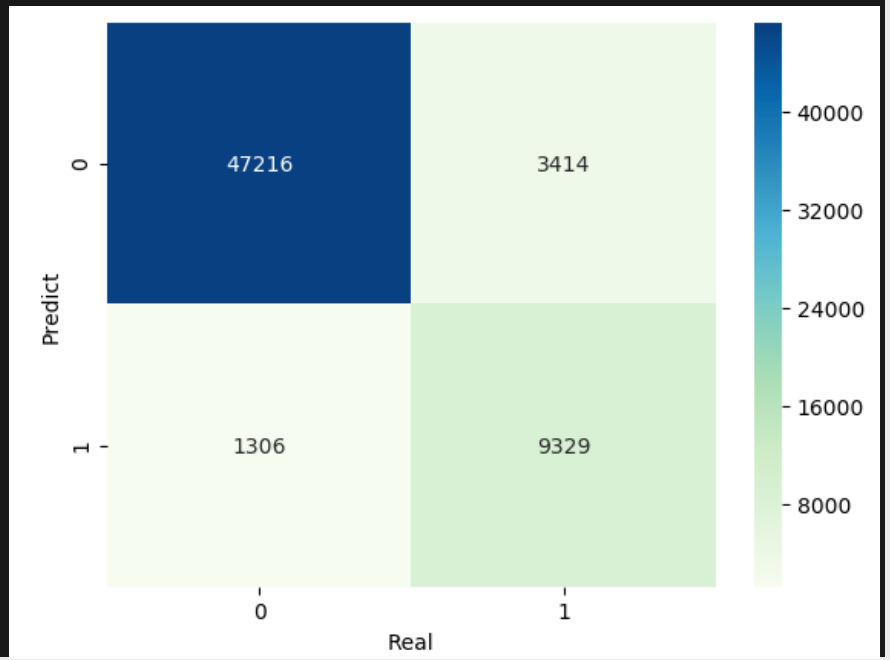
高斯贝叶斯—皮肤识别

```
# 读入数据
skin = pd.read_excel(r'C:\Users\Administrator\Desktop\Skin_Segment.xlsx')
# 样本拆分
X_train,X_test,y_train,y_test = model_selection.train_test_split(skin.iloc[:,:3], skin.y,
                                                                 test_size = 0.25, random_state=1234)
# 调用高斯朴素贝叶斯分类器的“类”
gnb = naive_bayes.GaussianNB()
# 模型拟合
gnb.fit(X_train, y_train)
# 模型在测试数据集上的预测
gnb_pred = gnb.predict(X_test)
```

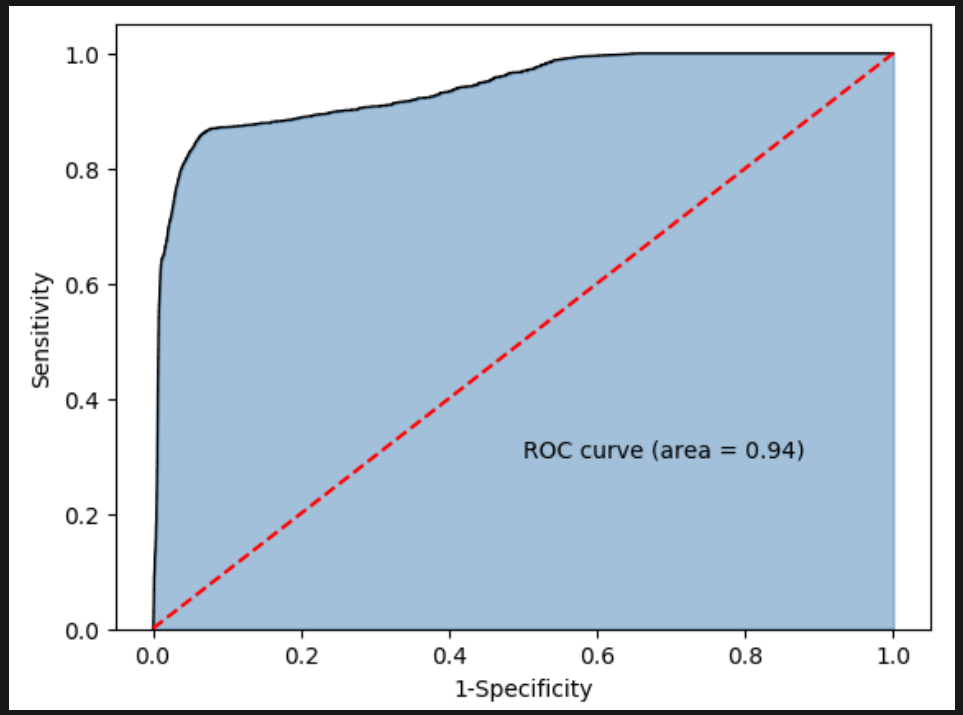
高斯贝叶斯—皮肤识别

```
# 构建混淆矩阵
cm = pd.crosstab(gnb_pred,y_test)
# 绘制混淆矩阵图
sns.heatmap(cm, annot = True, cmap = 'GnBu', fmt = 'd')
# 去除x轴和y轴标签
plt.xlabel('Real')
plt.ylabel('Predict')
# 显示图形
plt.show()

print('模型的准确率为：\n',metrics.accuracy_score(y_test, gnb_pred))
模型的准确率为：
 0.922957643026
```



高斯贝叶斯—皮肤识别



朴素贝叶斯模型

多项式贝叶斯分类器

MultinomialNB(alpha = 1.0, fit_prior = True, class_prior = None)

alpha：用于指定平滑系数a的值，默认为1.0

fit_prior：bool类型参数，是否以数据集中各类别的比例作为$P(C_i)$的先验概率，默认为True

class_prior：用于人工指定各类别的先验概率$P(C_i)$，如果指定该参数，则参数fit_prior不再有效

多项式贝叶斯—毒蘑菇识别

```
# 读取数据
mushrooms = pd.read_csv(r'C:\Users\Administrator\Desktop\mushrooms.csv')

# 将字符型数据做因子化处理，将其转换为整数型数据
columns = mushrooms.columns[1:]
for column in columns:
    mushrooms[column] = pd.factorize(mushrooms[column])[0]
```

	type	cap_shape	cap_surface	cap_color	bruises	odor	gill_attachment	gill_spacing	gill_size	gill_color	...	stalk_surface_above
0	poisonous	convex	smooth	brown	yes	pungent	free	close	narrow	black	...	smooth
1	edible	convex	smooth	yellow	yes	almond	free	close	broad	black	...	smooth
2	edible	bell	smooth	white	yes	anise	free	close	broad	brown	...	smooth
3	poisonous	convex	scaly	white	yes	pungent	free	close	narrow	brown	...	smooth
4	edible	convex	smooth	gray	no	none	free	crowded	broad	black	...	smooth

5 rows × 22 columns

	type	cap_shape	cap_surface	cap_color	bruises	odor	gill_attachment	gill_spacing	gill_size	gill_color	...	stalk_surface_above_r
0	poisonous	0	0	0	0	0	0	0	0	0	...	0
1	edible	0	0	1	0	1	0	0	1	0	...	0
2	edible	1	0	2	0	2	0	0	1	1	...	0
3	poisonous	0	1	2	0	0	0	0	0	1	...	0
4	edible	0	0	3	1	3	0	1	1	0	...	0

5 rows × 22 columns

多项式贝叶斯—毒蘑菇识别

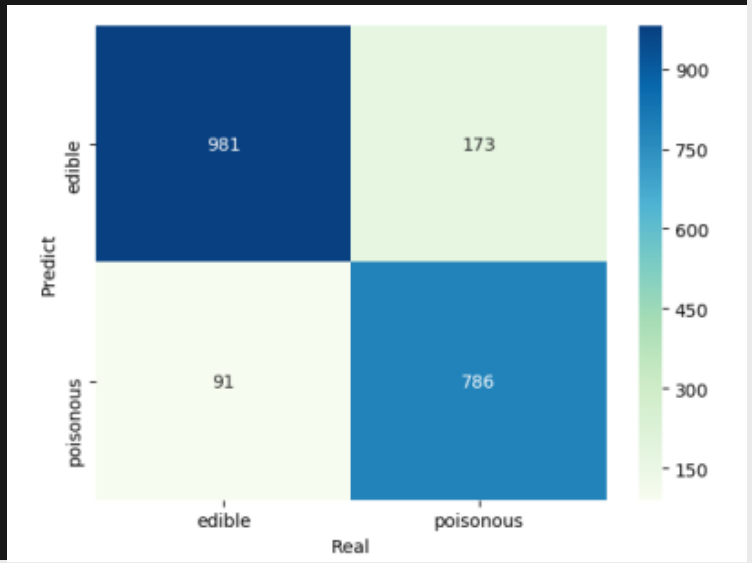
```
# 将数据集拆分为训练集合测试集
Predictors = mushrooms.columns[1:]
X_train,X_test,y_train,y_test = model_selection.train_test_split(mushrooms[Predictors],
                                                                 mushrooms['type'],
                                                    test_size = 0.25, random_state = 10)
# 构建多项式贝叶斯分类器的“类”
mnb = naive_bayes.MultinomialNB()
# 基于训练数据集的拟合
mnb.fit(X_train, y_train)
# 基于测试数据集的预测
mnb_pred = mnb.predict(X_test)
```

多项式贝叶斯—毒蘑菇识别

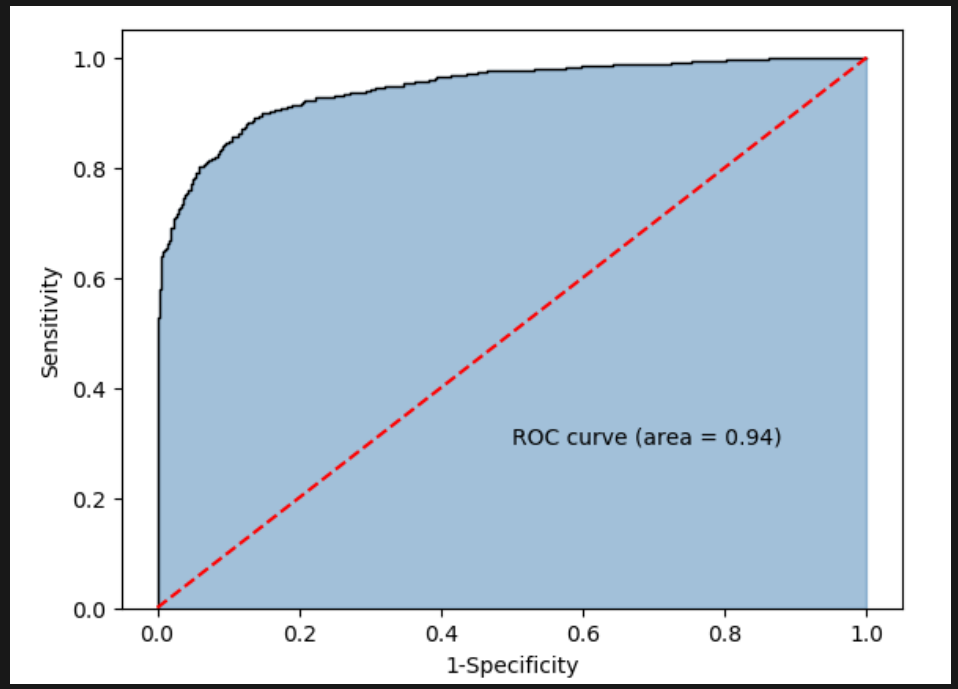
```
# 构建混淆矩阵
cm = pd.crosstab(mnb_pred,y_test)
# 绘制混淆矩阵图
sns.heatmap(cm, annot = True, cmap = 'GnBu', fmt = 'd')
# 去除x轴和y轴标签
plt.xlabel('')
plt.ylabel('')
# 显示图形
plt.show()

# 模型的预测准确率
print('模型的准确率为：\n',metrics.accuracy_score(y_test, mnb_pred))

模型的准确率为：
0.870014771049
```



多项式贝叶斯—毒蘑菇识别



朴素贝叶斯模型

伯努利贝叶斯分类器

BernoulliNB (alpha = 1.0, binarize=0.0, fit_prior = True, class_prior = None)

alpha：用于指定平滑系数a的值，默认为1.0
binarize：如果该参数为浮点型数值，则将以该值为界限，当自变量的值大于该值时，自变量的值将被转换为1，否则被转换为0；如果该参数为None时，则默认训练数据集的自变量均为0-1值
fit_prior：bool类型参数，是否以数据集中各类别的比例作为$P(C_i)$的先验概率，默认为True
class_prior：用于人工指定各类别的先验概率$P(C_i)$，如果指定该参数，则参数fit_prior不再有效

伯努利贝叶斯—情感分析

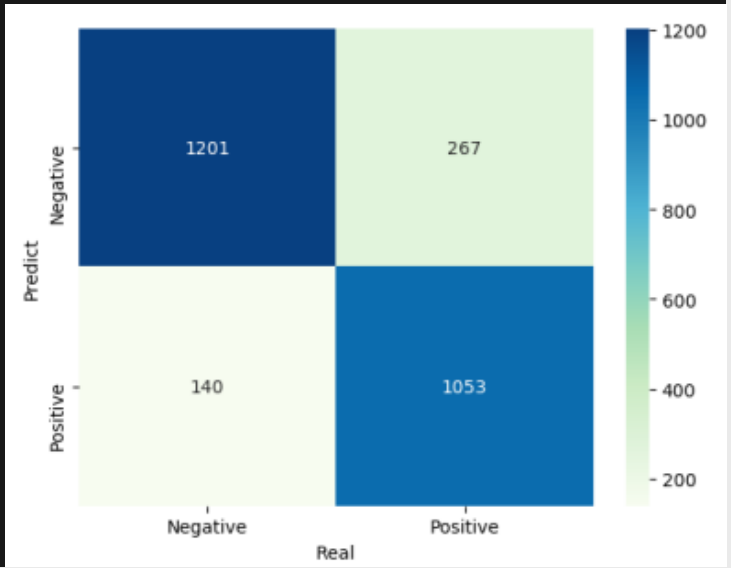
```
# 读入评论数据
evaluation = pd.read_excel(r'C:\Users\Administrator\Desktop\Contents.xlsx',sheetname=0)
# 运用正则表达式，将评论中的数字和英文去除
evaluation.Content = evaluation.Content.str.replace('[0-9a-zA-Z]','')
# 加载自定义词库
jieba.load_userdict(r'C:\Users\Administrator\Desktop\all_words.txt')
# 读入停止词
with open(r'C:\Users\Administrator\Desktop\mystopwords.txt', encoding='UTF-8') as words:
    stop_words = [i.strip() for i in words.readlines()]
# 构造切词的自定义函数，并在切词过程中删除停止词
def cut_word(sentence):
    words = [i for i in jieba.lcut(sentence) if i not in stop_words]
    # 切完的词用空格隔开
    result = ' '.join(words)
    return(result)
# 调用自定义函数，并对评论内容进行批量切词
words = evaluation.Content.apply(cut_word)
```

伯努利贝叶斯—情感分析

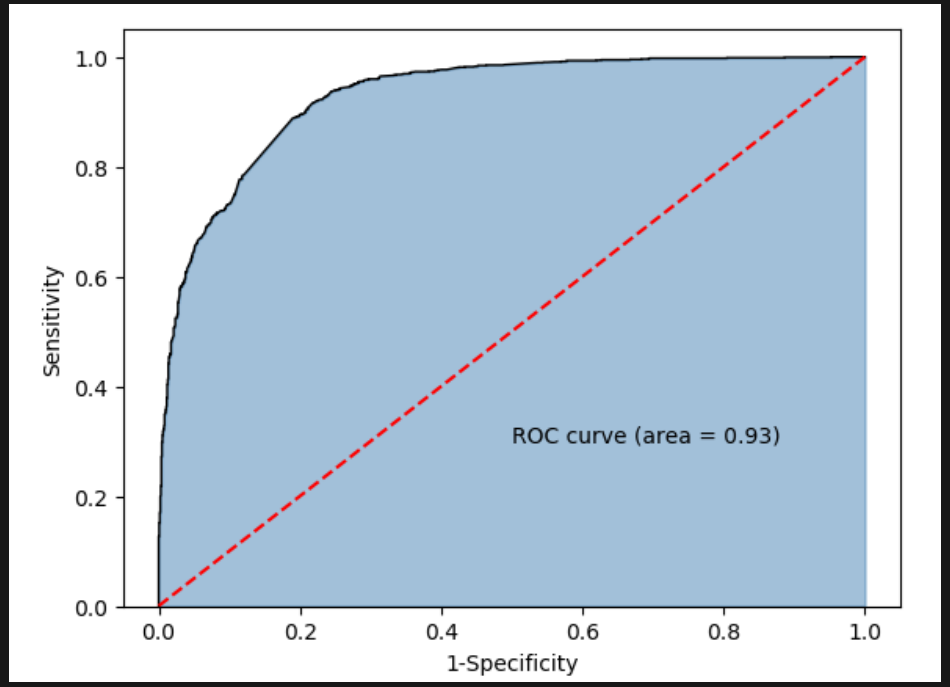
```
# 计算每个词在各评论内容中的次数，并将稀疏度为99%以上的词删除
counts = CountVectorizer(min_df = 0.01)
# 文档词条矩阵
dtm_counts = counts.fit_transform(words).toarray()
# 矩阵的列名称
columns = counts.get_feature_names()
# 将矩阵转换为数据框，即X变量
X = pd.DataFrame(dtm_counts, columns=columns)
# 情感标签变量
y = evaluation.Type
```

伯努利贝叶斯—情感分析

```
# 将数据集拆分为训练集和测试集
X_train,X_test,y_train,y_test = model_selection.train_test_split(X,y,test_size = 0.25,random_state=1)
# 构建伯努利贝叶斯分类器
bnb = naive_bayes.BernoulliNB()
# 模型在训练数据集上的拟合
bnb.fit(X_train,y_train)
# 模型在测试数据集上的预测
bnb_pred = bnb.predict(X_test)
# 构建混淆矩阵
cm = pd.crosstab(bnb_pred,y_test)
# 绘制混淆矩阵图
sns.heatmap(cm, annot = True, cmap = 'GnBu', fmt = 'd')
# 去除x轴和y轴标签
plt.xlabel('Real')
plt.ylabel('Predict')
# 显示图形
plt.show()
```



伯努利贝叶斯—情感分析



EDU

CSDN学院　IT实战派