蘑菇分类案例学习

目录

1.摘要

蘑菇的分类一直以来都是一个实用且与生活息息相关的问题，如何分辨出一种蘑菇是否是可食用的一直困扰着生活中的大家。在本例中，通过对蘑菇数据集的 22 个属性进行可视化分析、相关性分析等特征工程，选择出 13 个属性作为主要的分类特征，使用 SVM、决策树、朴素贝叶斯三种算法对蘑菇是否有毒的分类任务进行了训练和测试，最终得到了准确率100%的可靠分类结果。

2.背景

通过对蘑菇的外观特征进行识别，从而判断蘑菇是否可食用，是从我们的祖先开始就使用的方法，直至今天，我国云南省等地仍然有着吃野菌菇的习惯，尽管人们已经总结了许多识别野菌菇的方法，比如颜色、褶皱、菌盖的大小等等，但是在每年在云南仍然有大量的因为吃了有毒的野蘑菇而中毒致幻的受害者，因此仅仅根据人工总结出的，比较单一结构进行评判的经验，并不能保证对蘑菇进行分类的准确性。因此我们考虑使用机器学习的方法来解决蘑菇分类的问题。

3.数据集

3.1 数据集概览

蘑菇分类数据集来自于 UCI 机器学习数据集网站在 1987 年发布的一个数据集，之后迅速成为机器学习分类案例的经典之作，该数据集的原始属于来源于美国在 1981 年出版的一本指南手册——The Audubon Society Field Guide to North American Mushrooms (1981)，其中涵盖了 23 个大类的有帽蘑菇的描述以及对他们是否可食用的标记，每一种蘑菇都被标注为了 edible 或者 poisonous 着两种标签。

该数据集总共包含 8124 条数据记录，每条数据记录对应着对一种蘑菇的描述，共有 23 个属性列，其中包括 22 个对其外观、气味等因素进行描述的特征，以及一个分类标签，标记该种类的蘑菇是否有毒。数据集为了尽量保持建议，所有的属性的属性值都用缩写的形式来代替，从下面的统计也可以看出，可食用的蘑菇和有毒的蘑菇基本上数量是相等的，因此可以说明我们这个数据集是平衡的。

```
dataset.describe()
```

	class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color
count	8124	8124	8124	8124	8124	8124	8124	8124	8124	8124
unique	2	6	4	10	2	9	2	2	2	12
top	e	x	y	n	f	n	f	c	b	b
freq	4208	3656	3244	2284	4748	3528	7914	6812	5612	1728

尽管我们从数据集的描述层面发现数据集不包含空值，在查看具体的取值情况时，我们发现 stalk-root 属性有 2481 条数据记录存在取值为"?"的情况，因此在数据预处理时可能需要进行一定的注意。

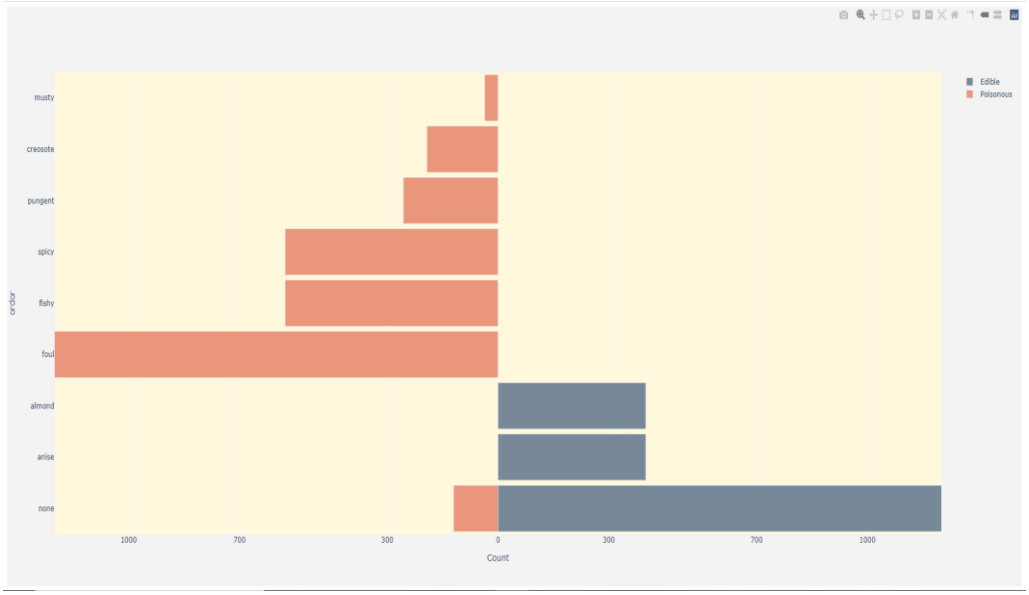
3.2 探索性数据分析及特征工程

3.2.1 可视化分析

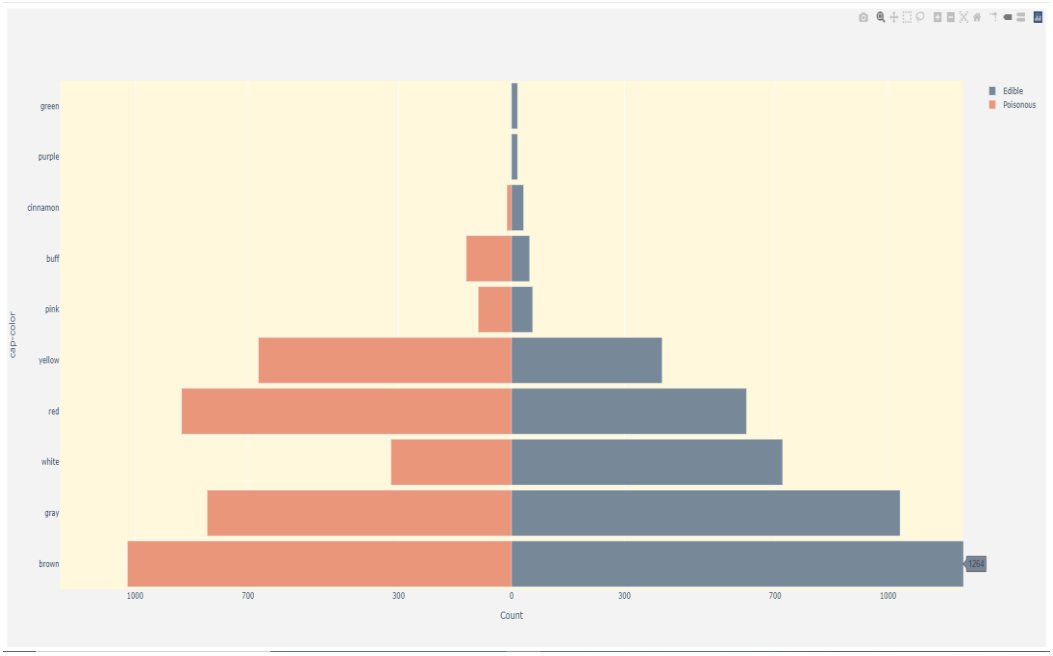
在进行完数据集的概览之后，在进行分类任务之前，我们可以使用可视化的工具对各个属性进行分析，对于一个好的分类特征，有毒和无毒的蘑菇应该在该属性的取值方面有较大的差异。

我们通过 python 的 plotly 进行数据的可视化，在数据可视化之前，将数据的属性取值，从简写替换为全称，便于之后的分析，我们主要使用两种图表形式，对所有的 22 个属性列都进行了分析。

限于篇幅，不再一一列举，接下来用几个具体属性作为呈现：



上图表示呈现的是气味（odor）这个属性各个取值的情况，我们可以看出，前面六种气味全部都是可食用的，而后面三种气味的蘑菇基本上全部是不可食用的。作为对比，我们对菌帽的颜色进行同样的可视化呈现：



可以发现，菌帽的颜色基本上基本上区分度不大，在每一种颜色的取值上，基本上可食用和有毒的蘑菇的数量基本上是一样的，因此菌帽的颜色并不是一个好特征，我们在进行特征选择时可以将其删去。类似的还采用了另外一种图表的形式进行可视化呈现：

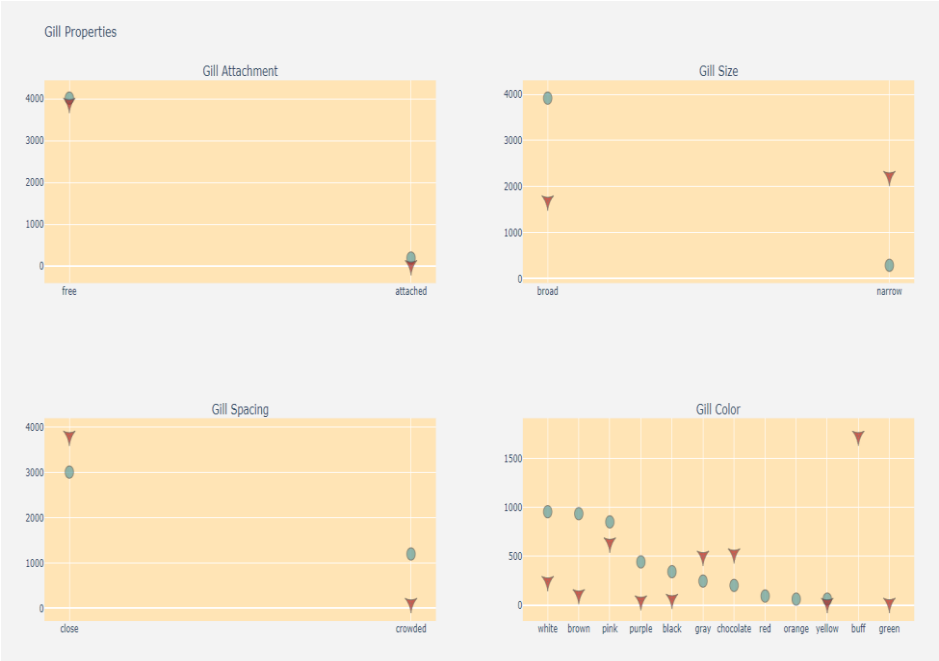
同样的可以发现 gill attachment 这个属性的取值方面，有毒和无毒基本上没有差别，因此也并不是一个进行分类的好的特征。

我们按照上述的方法，对所有的 22 个属性都进行了可视化的特征分析，从其中剔除掉了,“cap-shape”,“cap-surface”,“cap-color”,“gill-attachment”等 9 个特征，使用其余的 13 个特征作为我们进行分类的依据属性。

3.2.2 相关性分析

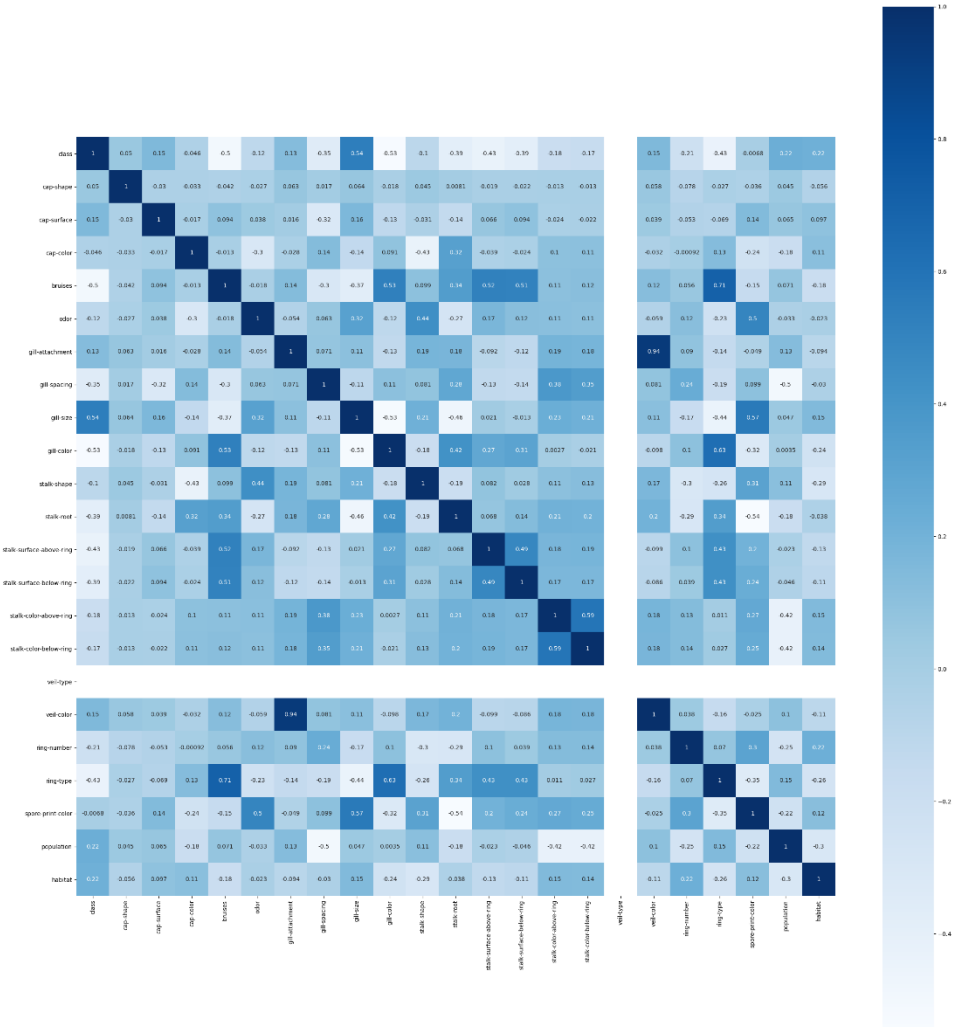
因为该变量并非连续型的变量，因此在做相关性分析的时候我们可以采用定类变量的相关性分析方法或者定序变量的相关性分析方法。首先我们使用定类变量的相关性分析方法——卡方检验，来判断某个属性在有毒和无毒的取值上是否有显著的差异，我们返回卡方检验的 p 值大于 0.05 的变量，结果只有一个变量不满足：

```
[[4208]
 [3916]]
veil-type
1.0000
```



我们可以看出，这个变量的取值只有一个，也就是说，所有的蘑菇在这个变量的取值上是相同的。但是与此同时，我们可以发现，在上文的可视化分析中，我们找出了 9 个区分度不大的特征，然而使用卡方检验的结果却并不好，因此卡方检验的方法并非有效的。

其次我们可以使用斯皮尔曼相关性系数，用定序变量的方法来分析相关性，相关性热力图如下所示（该图较大，截图可能看不清楚，可以在附件中找到原图）：



可以看出使用斯皮尔曼相关性系数，我们能剔除一些相关性较低的属性值，如 veil-type 属性，但我们同时可以发现，几乎所有的变量的相关性系数都比较低，并且基本差不多大小，所以根据相关性系数来进行属性的剔除和选择，仍然会存在较大的误差，对比于上文可视化分析中我们剔除的九个变量，这些变量基本上相关性系数都较低，因此我们选择可视化分析中得到的结果作为我们最终模型输入的属性值。

4.数据预处理方法

4.1 数据编码

在数据编码部分，在进行相关性分析前，我们就需要将我们的数据进行初步的编码，将属性的取值转化为数字，进行数字编码。

同时我们注意到上文曾经提出，stalk-root 属性有 2481 条数据记录存在取值为"?"的情况，这部分数据需要进行空值处理，但是 stalk-root 属性的区分度本身并不大，因此在特征选择时并没有作为一个分类属性作为分类的依据，所以我们直接将该属性剔除即可。同时我

们还要将特征工程里所有的不符合要求的特征剔除，最终保留 13 个属性列。

观察这 13 个属性列的取值情况，都不是连续的变量，而是有着多种取值可能的定类变量，在进行相关性计算时，将其当作定序的变量进行处理计算相关性系数无伤大雅，但是在机器学习任务中，直接使用编码的结果很容易出现问题，例如用数字 1-12 表示 1-12 月，那么就潜在表示了 12 月和 1 月差的很远，其实 12 月和 1 月的距离很近，同时，例如对于 gill-color 的属性来说，黄色、红色和黑色可能会被编码为 0、1、2，那么就隐含着黄色和黑色的差距要比红色和黑色要大，其实并没有这种差距，因此哑变量（虚拟变量）编码就是必要的，我们需要根据有多少种取值的情况，将每一种取值都划分为一个特定的列，最终我们的 13 个属性列就转变为了 67 个属性列，如下所示：

	class	cap-shape	cap-surface	...	spore-print-color	population	habitat
0	1	5	2	...	2	3	5
1	0	5	2	...	3	2	1
2	0	0	2	...	3	2	3
3	1	5	3	...	2	3	5
4	0	5	2	...	3	0	1
...	...	...	...	...	...	...	...
8119	0	3	2	...	0	1	2
8120	0	5	2	...	0	4	2
8121	0	2	2	...	0	1	2

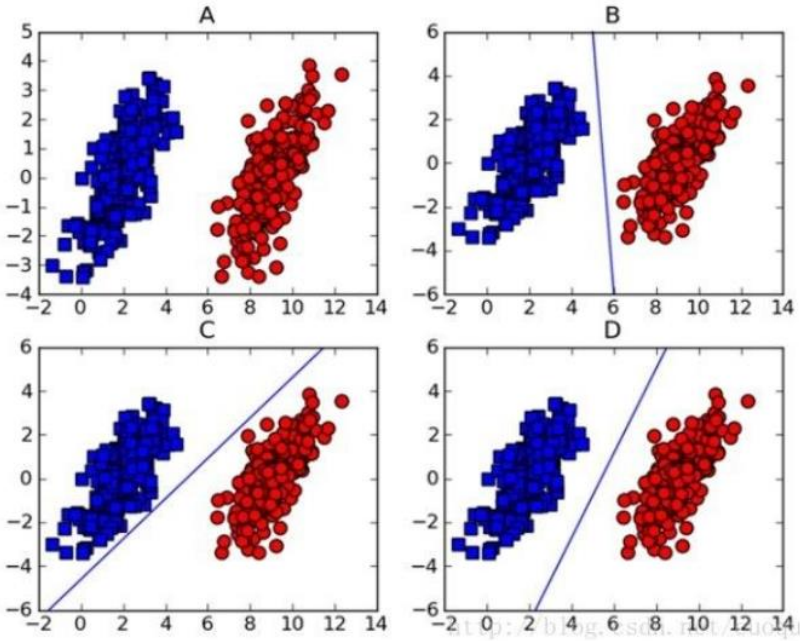
	bruises_1	odor_1	odor_2	...	habitat_4	habitat_5	habitat_6
0	1	0	0	...	0	1	0
1	1	0	0	...	0	0	0
2	1	0	0	...	0	0	0
3	1	0	0	...	0	1	0
4	0	0	0	...	0	0	0
...	...	...	...	...	...	...	...
8119	0	0	0	...	0	0	0
8120	0	0	0	...	0	0	0
8121	0	0	0	...	0	0	0

4.2 训练集测试集的划分

在将数据投入到分类器之前，我们需要对数据进行训练集和测试集的划分，我们的数据质量比较高，同时根据最后机器学习算法分类结果的高准确率，我们决定将原数据按照 1:1 的比例划分为训练集和测试集。

5.机器学习算法

在机器学习的算法选择方面，我主要选择了 SVM 算法以及作为基础模型的朴素贝叶斯算法、决策树算法。



SVM 算法的主要思想是使用一个超平面将数据点进行分类，主要的优化的方法是首先找到各类数据点中距离分割超平面最近的点，然后计算出这些点到超平面的距离，最后不断调整超平面的位置（也就是相关的参数）使得这些点的距离总和最大，对于线性分类问题，主要的目标函数如下所示：

$$arg\max_{w,b}\{\min_{n}(y_i \cdot (w^T + b)) \cdot \frac{1}{\|w\|}\}$$

具体的优化求解方法使用的是拉格朗日乘数法。对于某些线性不可分的问题，也可以使用 SVM 算法来解决，这是需要将核函数换为非线性的函数，其中使用的较多的是高斯核函数：

$$K(x,z) = \exp\left(-\frac{\|x-z\|^2}{2\sigma^2}\right)$$

对于朴素贝叶斯算法和决策树算法，课上已经详细介绍，在此不再赘述。

6.结果及讨论

6.1 模型的结果

首先在结果方面，我们首先对比使用哑变量编码和不使用哑变量编码的差异，不适用哑变量进行编码时，SVM 模型的结果如下所示：

Classification Report:

	precision	recall	f1-score	support
-1	0.97	0.98	0.97	2085
1	0.97	0.96	0.97	1977

混淆矩阵如下所示：

我们可以看出，分类为无毒的蘑菇中其中 2107 个蘑菇中有 72 个蘑菇都是有毒的，因

```
Confusion Matrix:
[[2035   50]
 [  72 1905]]
```

此不进行哑变量编码，我们的模型进行分类，还是存在一定的危险性的。

下面是进行哑变量编码之后，SVM 模型在测试集上的分类结果：

Classification Report:

	precision	recall	f1-score	support
-1	1.00	1.00	1.00	2085
1	1.00	1.00	1.00	1977

可以看到，模型在测试集上的分类结果达到了 100%的准确率，这说明 SVM 模型采用哑变量编码之后，分类的结果非常可靠，我们可以直接将其结果作为可信的蘑菇分类的依据。

同时，使用朴素贝叶斯算法和决策树算法的结果如下所示：

决策树：

Classification Report:

	precision	recall	f1-score	support
-1	1.00	1.00	1.00	2085
1	1.00	1.00	1.00	1977

朴素贝叶斯:

```
Classification Report:
              precision    recall  f1-score   support

          -1       1.00      0.97      0.99      2085
           1       0.97      1.00      0.99      1977
```

可以看出，二者的结果准确率都比较高。

6.2 针对结果的讨论

首先讨论模型的准确率较高的问题，准确率较高并不是过拟合问题，我在检查完代码之后，有足够的证据证明，我们的训练集和测试集是完全随机划分的，并且模型不仅仅在训练集上表现出了高准确率的结果，同时也在测试集上表现出了高准确率的结果，并且我们最终使用 k 折交叉验证的方法来判断，发现准确率均较高，因此并不存在过拟合的问题。

其次对模型参数的选择上，因为我们采用的三个算法，模型的结果都比较好，准确率都较高，所以使用默认参数都能完成要求，但是我们要指出，如果模型的表现不好，可以考虑调参来调优，可以使用 k 折交叉验证的方法，先在训练集上找到最优的参数，然后再到测试集中进行验证（相关的代码在 ML.py 文件中有所展示）同时也可以使用绘制学习曲线的方法来选择最优的参数，一般可以使用 auc 值作为评判的标准，学习曲线可以有效的避免模型的过拟合问题，当然本例中不涉及过拟合问题，因此不再详细介绍。

7.小结

对于特征的选择方面，一定要根据数据的特点来进行特征选择，而不要最求高大上，例如本例中使用相关性热力图分析得到的效果并不直观，反倒是使用可视化图标进行分析的方法进行特征选择更为有效。

对于数据的预处理方面，使用的数据编码的方式应该要与选取的机器学习算法保持一致，例如本例中 SVM 算法一般要求标签值为+1 和-1，所以需要在编码时进行一定的调整。而且要注意哑变量编码的问题，在某属性列有多个取值可能的时候，考虑哑变量编码来消除数据结构上的误差。

对于机器学习算法的选择，有时不能简简单单只选用一种方法，有时要综合多种算法来选取最优的，例如本例中朴素贝叶斯算法的效果就不如 SVM 和决策树，同时对于模型的结果可以考虑进行可视化。同时，有时算法表现不良好，可能是数据预处理阶段的失误，而并非是模型参数的问题，例如本例中没有进行哑变量编码时，SVM 算法分类结果的准确性有所降低。