任务一实现了两种分类方法：决策树归纳法和朴素贝叶斯算法，在算法描述中进行了叙述。在实验对比中采用10-fold cross-validcation进行了正确率比对。在降维处理中讨论了降维的方式与结果。

# 算法描述：

任务一实现了两种分类方法：决策树归纳法和朴素贝叶斯算法。所有的分类维度均预先根据其性质标记为连续维度与离散维度。

## 决策树

采用经典的决策树算法。

对于训练数据的各个维度，首先根据一种属性选择度量（选择分裂准则），对当前所有的维度进行计算，根据分裂准则选出效果最好的一种（例如属性A），将数据集按照A为2类（连续值）或n类（离散值，n为该类下的所有取值种类数）。删除属性A。

对于分出的各分区D，递归的进行以上操作，直到满足以下条件：

1. 数据集中所有项的类别均相同：返回该类别；
2. 维度已经为空（已删掉所有维度）：返回该分区下较多的类。

具体的属性选择度量，这里选的是信息增益：

对D中元组分类所需要的期望信息如下：



其中pi是D中任意元组属于类Ci的非零概率。我们还需要的信息由下式度量：



信息增益的定义即为元信息需求与新的信息需求之间的差：



因此，每一次划分都要选出最高信息增益的属性作为分裂属性，这等价于选择最小的

对于离散值，的计算中Dj即为该维度下所有可能的取值各自的集合；而对于连续值的维度，需要将所有出现的值排序（例如，排序为a0,a1,…an），选取两点间的中点bi（bi=ai+1-ai）,对所有bi，将分区D划分为：as小于等于bi，和as大于bi两部分。均进行划分尝试，选出最小的取值bi作为分裂点。

算法的伪代码如下：

**Algorithm** decision:

输入：

·数据分区D

·候选属性集合；

输出：决策树。

方法：

1.创建节点N；

2.如果D中元组都属于类别Ci或候选属性集合为空：

3. 返回D中较多的类别C

4.选出最好的属性的最好的分裂点，将D分为多路

5.删除选出的属性

6.对每一块分区，递归调用decision算法。

## 朴素贝叶斯分类器

贝叶斯分类器基于贝叶斯定理实现，即：



设D为训练元组，元组有n维属性。其中X为某项的各个维度取值，而H为所属类别。显然，P(H|X)即为基于维度做出的预测（后验概率），我们要做的就是利用先验概率（P(X|H)为当前训练集下某类别中X属性所占的比例）建模。先验概率可以用频率计算概率，而P(H)和P(X)也可以计算。

由于X为多种属性的组合，计算联合概率比较困难。但是可以假设各个属性之间互相独立（朴素贝叶斯算法），则对X(x1,x2,…xt)有：



对于离散值，P(xi|H)可以由D中该属性取值为xi的数量占全部D数量的比例求得，对于连续值，计算D中该属性的均值与方差（方差采用无偏估计），假设属性服从均值为μ，方差为σ的高斯分布：



则对于P(xk|C)有：



因此bayes算法对于训练数据所做的工作是：对于所有的分类种类（y/no），计算连续值属性的均值和方差，计算离散值属性的各取值概率（频率）分布。

算法的伪代码如下：

**Algorithm** bayesian:

输入：

·训练数据D

·候选属性集合；

输出：贝叶斯模型。

方法：

1.按照最终类别将D划分为多个分区Di

2.对于所有属性Fi，所有Di，依次进行3~步

3.如果Fi为连续值，计算Di上该属性的均值和方差

4.如果Fi为离散值，计算Di上该属性的各个取值所占Di总数的比例

# 交叉验证

采用10 fold cross validation进行分类结果的验证。

决策树耗时（10次建树+所有数据检验，以下皆同）：2311.34 s，准确率：78.51%,如下：

|  |  |  |  |
| --- | --- | --- | --- |
| fold | right | wrong | rate |
| 0 | 3841 | 277 | 93.27% |
| 1 | 3872 | 246 | 94.03% |
| 2 | 3706 | 412 | 90.00% |
| 3 | 3682 | 436 | 89.41% |
| 4 | 3740 | 378 | 90.82% |
| 5 | 3694 | 424 | 89.70% |
| 6 | 3111 | 1007 | 75.55% |
| 7 | 3134 | 984 | 76.10% |
| 8 | 3345 | 773 | 81.23% |
| 9 | 210 | 3915 | 5.09% |
| total | 32335 | 8852 | 78.51% |

表 0‑1 10-fold 决策树验证结果

贝叶斯耗时：6.69 s，准确率：78.78%，其中建树耗时不足1毫秒。

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 3999 | 119 | 97.11% |
| 1 | 3981 | 137 | 96.67% |
| 2 | 3959 | 159 | 96.14% |
| 3 | 3862 | 256 | 93.78% |
| 4 | 3893 | 225 | 94.54% |
| 5 | 3898 | 220 | 94.66% |
| 6 | 3225 | 893 | 78.31% |
| 7 | 1616 | 2502 | 39.24% |
| 8 | 2119 | 1999 | 51.46% |
| 9 | 1900 | 2225 | 46.06% |
| total | 32452 | 8735 | 78.78% |

表 0‑2 10-fold beyes验证结果

由于不需要递归调用，贝叶斯算法的复杂度远低于决策树，而最终效果持平，显示了贝叶斯算法的强大之处。

总体来看，最后一部分实验数据的性质与前面的数据差别较大，具体表现为后三组。二者的准确率总体接近，但可以看出，决策树在fold-9，贝叶斯在fold-7上二者差别较大。两者的区别还是有的。

# 装袋优化

由装袋的思想，我们进行了以下优化：对于十组数据训练得到了十个模型，验证分区Di时，训练数据是其他九个分区形成的九个模型分别验证，按照表决多数选取最终结果，优化后的结果如下：

决策树耗时：180.40 s，准确率：87.64%。决策树的建树时间与训练数据量关系明显，装袋优化极大的提升了时间，同时明显提高了准确率。

贝叶斯耗时：14.37 s，准确率：88.37%。多出的耗时主要是多次检验的耗时。装袋优化对于贝叶斯的提升非常明显。

# 降维优化

在格式说明中提到duration对结果的影响较大，因为如果为duration为0的话那么一定是no（未通话），为了得到更实际的结果应该去掉它，我们对此进行了分析：

duration与类别为“y”占比关系图如上，所有数据中并没有出现duration=0的项，这一决定性因素并没有出现，因此暂时没有删除duration维度。

对于离散纬度，统计其unknown所占比例。Unknown不应该作为一个单独的取值，因为他对于其他所有取值均有相同的取值可能，所以不提供信息，如果unknown占比过大则可以考虑删除。

可以看到只有default的no中所占比例较大（大约1/7），我们尝试将default去掉，验证结果：

|  |  |  |
| --- | --- | --- |
| total | 82.034% | 82.153% |
| time | 23.71 s | 24.38 s |

可以看到，去除该维度减少了运行时间，而对准确率几乎没有降低。

我们进行了以下两种尝试：

1. 依次去除每个维度，计算比较准确率；（去掉属性后准确率越高，说明该属性约不影响结果）
2. 计算各个维度不同取值的频率分布。对于去掉某维度后准确率仍然最高的几个维度，观察其取值与类别的相关度验证猜想。
3. 去掉最不相关的一些属性，再次运行，验证结果。

观察euribor3m：

各种取值与类别的相关度极差，前面猜想得到验证。

去掉最不重要的前十维，耗时减少同时准确度提高。

"euribor3m","emp.var.rate","nr.employed","campaign","cons.price.idx","contact","loan"，"housing","marital"

# 结论：

最终我们的耗时和准确度为：

贝叶斯耗时：6.46s，准确度：81.40%，相比于交叉验证时的14s耗时有所提升。准确率相比于原装袋优化，在降了10维之后，有所下降，但相比于原始的78%仍提升明显。

决策树耗时：108.50s，准确率：87.86%。运行时间进一步缩短，准确率由于去掉了不好的剪枝，因此并没有受到影响。

装袋优化对于两种方法优化均较为明显。

降维优化在时间方面对于决策树的提升较为明显；在准确率方面，贝叶斯归纳的准确率稍有下降，而决策树几乎不受影响。

考虑到贝叶斯方法原本耗时就不多，因此降维优化未必是最好的选择。对于决策树，降维优化显然是很大的提升。

各个维度删除后的计算结果和直方图统计已呈现在result文件夹中。

# 代码说明：

所有函数添加了注释，以下是一些概述：

Preprocessing/ generate\_test\_data.py：测试性数据；

Preprocessing/ Preprocessing.py：一些预处理；

decision/

decision\_tree：决策树相关的生成、验证函数。同时对于无优化、装袋优化、降维优化，最下面有备注是的三行脚本；

decision\_node：决策树的节点类

bayesian/

bayes\_classification：包括两部分：1.贝叶斯模型的生成和验证；2.由于贝叶斯速度较快，因此降维尝试的各种测试均基于贝叶斯方法，这些测试函数也写在这里。

test/

fold\_cross\_validfication:包括三个性能测试函数：无优化交叉验证；装袋交叉验证；降维交叉验证。

result/

histogram：包括所有属性取值与分类的相关性统计文件

del:包括删掉各个属性后的准确率结果。

cross-validfication：包括两种方法的三种测试结果。