汽车价值评估

问题

根据汽车的属性评估汽车的价值等级（接受度）。该问题在现实生活中具有一定的研究意义，如在二手车市场，当销售方为一大批的二手车定价时，可以先根据汽车的属性做价值评估，为具体的定价提供参考。

当然，限于各方面的因素，本人所做的工作简化了汽车评估中的复杂性，只考虑了汽车的某几个属性，而实际上汽车的价值评估要考虑的远远不止这几个属性。在此工作中，期望能够获得较高的准确率和较好的评估性能。

形式化

该问题是典型的分类问题。根据不同属性将汽车的价值分类到几个固定的标签中。

数据集

本人使用汽车评估数据集是从 UCI 的机器学习仓库中获得的。在这份汽车评估数据集中，汽车类标号如下：

class	unacc, acc, good, vgood

unacc, acc, good, vgood 表示汽车的价值从低到高。

使用的评估属性共 6 个，属性名及可能的属性值如下：

属性名	属性值
buying	vhigh, high, med, low
maint	vhigh, high, med, low
doors	2, 3, 4, 5more
persons	2, 4, more
lug_boot	small, med, big
safety	low, med, high

buying、maint、doors、persons、lug_boot、safety 依次表示汽车的：购买价格、维护费用、车门数量、可载人数、后备箱大小、安全性。本人使用的该汽车评估数据集共有 1728 条数据，覆盖各种情况，不含缺失值，具体见下表：

类标签	数量	比率
unacc	1210	70.023 %
acc	384	22.222 %
good	69	3.993 %
vgood	65	3.762 %

算法

由于该数据集的原始数据基本上是标称属性，本人在将数据集输入到算法前先做了预处理，由于 Python 在处理字符串时的极大方便，使用 Python 将原始数据集中的不同标称属性处理为整型类型之后再输入到算法中。

结合课本所学分类模型分析该数据集，数据集 size 较小，用人工神经网络实现难度较大；而朴素贝叶斯算法要求属性之间的条件独立，贝叶斯网络的网络拓扑结构需要主观的知识编码；线性支持向量机只区分开线性可分情况…鉴于以上情况，本人采用决策树和 KNN 算法来做汽车评估。训练完分类模型之后将检验集输入模型即可得到分类结果。由于环境所限，不提供数据可视化。由于决策树和 KNN 都是经典的分类方法，在此不再对决策树和 KNN 的原理进行赘述。

简要介绍本算法中的一些设计：

数据集划分：由于只有一份数据集，没有专用测试集，需要手动划分数据集，鉴于数据集本身只有 1728 条数据，采用传统留出法会使得训练集较小，这里采用了 bootstrap 方法将数据集划分成训练集（size：1728）和检验集（size：600~650）。

不纯性度量：决策树节点的分裂需要寻找最佳分裂属性，不纯性度量用来比较不同属性划分结果，传统的不纯性度量熵、Gini 系数和分类误差，在本算法中都尝试过，最后采用了熵作为不纯性度量且用于信息增益的计算。

预剪枝：为了避免模型过拟合，往往需要对决策树进行先剪枝或者后剪枝，由于在算法设计中本人采用的数据结构因素，对决策树进行后剪枝较为复杂，所以采用预剪枝技术，通过对不同参数进行

实验评估，最后决定当节点中的数据集大小不超过 7 时，停止分裂。

K 值决定：在 KNN 算法中，对于不同的 K 值，得到的算法效果也不一样，在本算法中，本人比较了不同的 K 值对于算法效果的影响，但是由于数据集采用的自助法，不同的训练集对应的最佳 K 值也不一样。

距离函数：在 KNN 中距离函数用于计算两个数据之间的距离，针对汽车评估数据集，采用欧氏距离。

评估

下面分别评估决策树和 KNN 两个模型：

1. 错误率

分类模型	决策树	KNN
错误率	0.0999	0.0607

因为每次数据集划分出的训练集和检验集都不同，所以每次得到的决策树和 KNN 模型的错误率也不同，为了便于比较，这里去多次算法运行得到的平均值作为决策树和 KNN 的错误率来进行比较。但是在相同的训练集和检验集中，适当的 K 值总能使得 KNN 的错误率低于决策树 3 到 5 个百分点，说明在该特定的汽车评估数据集中，简单的 KNN 模型反而优于稍复杂的决策树模型。

2. 混淆矩阵

如果把兴趣放在类标号为 unacc（表示汽车不被接受，价值过低），则认为实际 unacc 预测为 unacc 的实例数记为 TP，实际 unacc 的预测为非 unacc 的实例数记为 FP，实际非 unacc 预测为 unacc 的实例数记为 FN，实际非 unacc 预测为非 unacc 记为 TN。

决策树混淆矩阵

检验集 size：Total = 631

		预测的类	
		unacc	非 unacc
实际的类	unacc	417	20
	非 unacc	15	179

度量如下表：

度量	公式	大小
准确率	$\frac{TP+TN}{Total}$	0.9445
精度	$\frac{TP}{TP+FP}$	0.9652
召回率	$\frac{TP}{TP+FN}$	0.9542
F1 分数	$\frac{2*p*r}{p+r}$	0.9597

KNN 混淆矩阵如下：

检验集 size：Total = 663

		预测的类	
		unacc	非 unacc
实际的类	unacc	455	7
	非 unacc	26	175

度量如下表：

度量	公式	大小
准确率	$\frac{TP+TN}{Total}$	0.9502
精度	$\frac{TP}{TP+FP}$	0.9459
召回率	$\frac{TP}{TP+FN}$	0.9848
F1 分数	$\frac{2*p*r}{p+r}$	0.9650

从决策树和 KNN 模型的混淆矩阵来看，当把 unacc 作为正样本，非 unacc 作为负样本时，决策树和 KNN 模型都能较好的区分开正样本和负样本。

以上评估只是某次实验结果或者多次实验结果的平均效果，因为采用的自助法划分数据集，每次得到不同的训练集与测试集，故每次实验结果有差别，但差异不大。

总结

对于汽车评估数据集，本人采用决策树和 KNN 算法分别对其进行训练和预测，对于影响算法的因素，通过多次实验得到较优参数。在两个模型中，光从错误率来看 KNN 的性能较决策树好，作为四分类问题，决策树错误率基本低于 10%，KNN 错误率大约 6%，都还算令人满意。如果将其简化为二分类问题，将 unacc 作为正样本，则两个模型的准确率都得到较大提升，在 95%以上，且精度和召回率都较高，说明在整个预测中，即使预测错误，预测结果和实际结果差别也不太大。