人工智能第二次大作业

2014011575　　自 41　　郭畅

1. 作业要求——蘑菇分类

你需要设计一个机器学习的算法，该算法的输入是蘑菇的特征，算法需要判断该蘑菇是否有毒（二分类）。蘑菇的特征总有 22 维度，每一维都是离散型变量，取值用特定的字母来表示。比如，特征“cap-shape”的取值有“x，b，s，f”。每一维特征的取值字母代表的含义见文件 mushroom_attr.txt。给定的数据集有 8124 个样例，在数据集文件 mushroom.csv 中第一行为表头，其他每一行代表一个样例，第一列为类别，包括“p”（有毒）和“e”（无毒），剩下每一列代表一个特征。你需要实现一个机器学习算法，利用第 2 列到第 23 列的特征去预测第 1 列的类别。具体要求如下：

1) 使用至少一种机器学习算法，可以是课上讲过的算法，也可以是其他的算法，请给出算法实现过程的描述。
2) 请画出两种类别的蘑菇在每一个特征上的分布图，并说明哪些特征在两类蘑菇中差异比较明显，即对预测类别有帮助。
3) 在数据集中，随机选取 80%的数据作为训练集，剩下 20%的数据作为测试集。请画出训练过程中训练集正确率和测试集正确率的变化曲线，还有训练完成后的分类 ROC 曲线。
4) 在数据集上做 10 倍交叉验证实验，并给出每一次实验的结果包括训练正确率和测试正确率。
5) 探讨可能提升算法效果的策略，比如数据预处理，特征提取，参数调整等。

2. 编译环境

Matlab 2017a，Win8.1 系统。

3. 算法描述

采用的方法：由多个决策树构成的随机森林。

采用该方法的理由：非数值特征。（正交编码会增加维数）

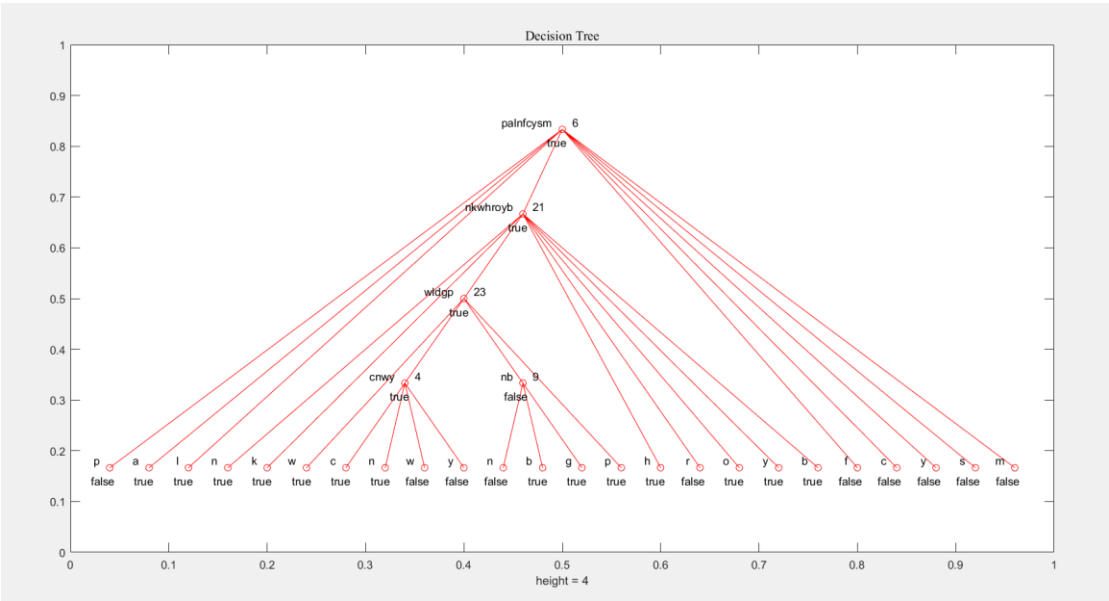
决策树的数据及结构：对训练集数据再进行又放回的随机采样，以便利用同一数据集构建不同的决策树。树的结构为二叉树，但由于样本每个特征的可能结果多于两个，故采用**孩子兄弟表示法**来构建决策树。

决策树的原理参照 ID3 方法（交互式二分法），即通过熵不纯度的变化作为指标，逐层选出最具分辨力的特征。先不考虑任何特征的类别，算出整体的熵不纯度；再逐一考察每个特征，判断引入一个新特征是否会使熵不纯度减少，比较哪个特征能够使不纯都减少幅度最大。

建树过程采用递归的方法。如果新节点中只有一类，则停止递归，该节点为叶子节点；否则直至终止条件结束递归。（本次算法实现中主要依赖对决策树层数的控制）由于采用孩子兄弟表示法，建树过程中，函数传递的是上一层的数据，到了新的兄弟时，根据特征列表信息 list 和位置信息 brotherNumber 重新分类出本节点的数据。

树节点中的数据主要为：被选出的特征、指向孩子节点的指针、指向父节点的指针、该特征列表、该节点判断结果。

测试机理：如果当前节点有孩子节点，则依照当前的特征属性列表，走向下一层指定位置；如果没有孩子节点，则返回当前节点的判断结果。最后将所有决策树的判断结果进行汇总投票，根据投票结果选出最终的判断结果。

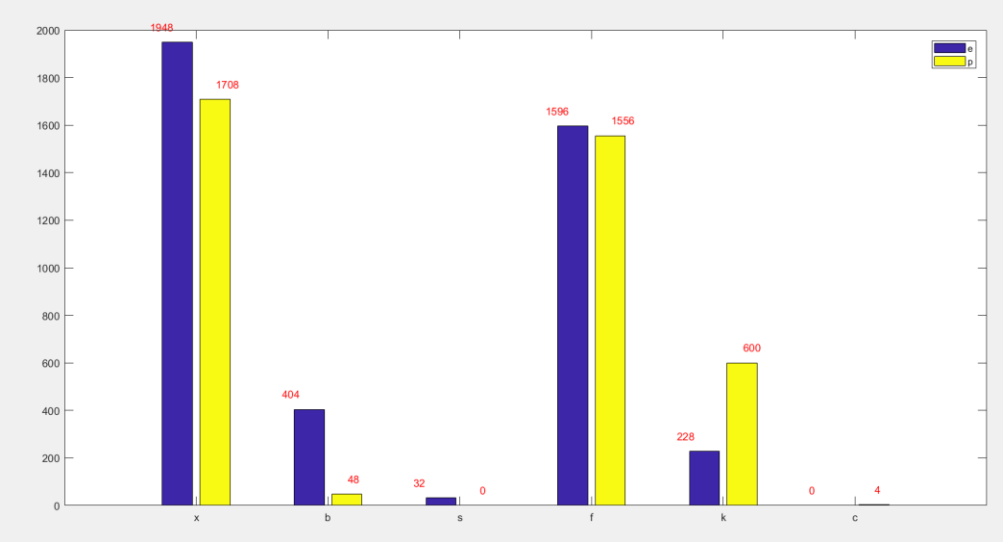


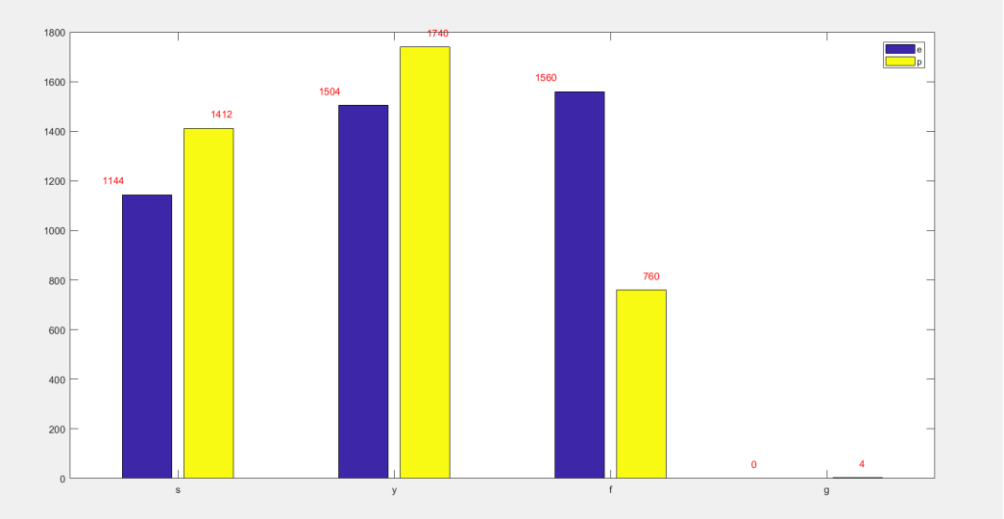
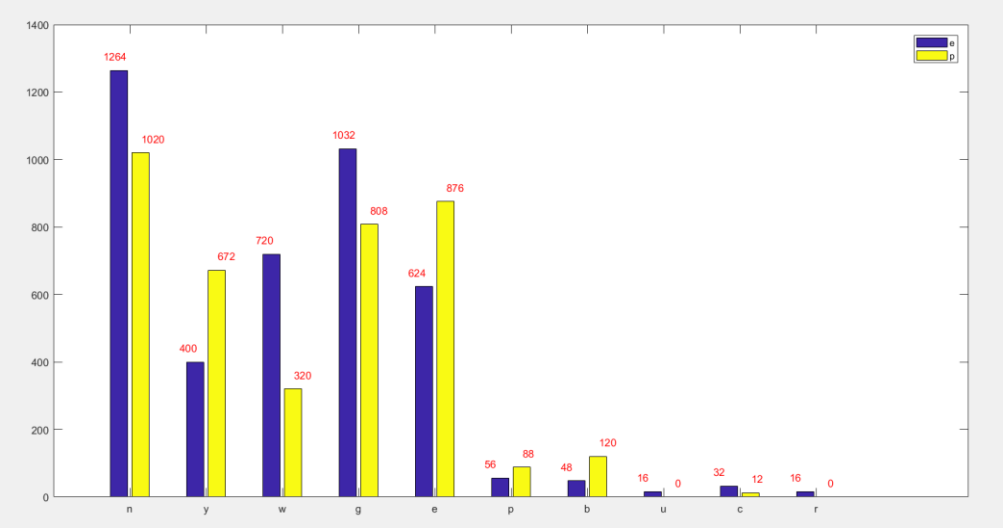
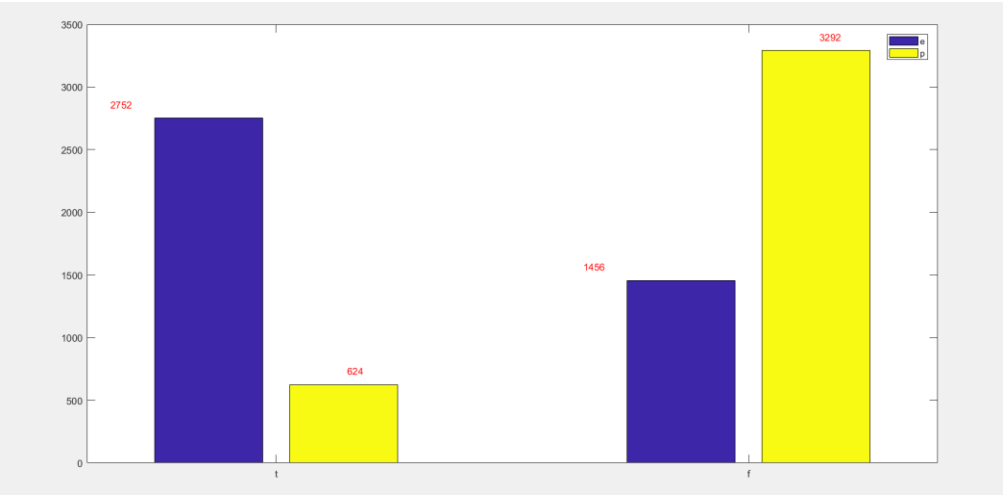
上面的决策树是根据所有数据建成的二叉树，在第 4 层完成所有数据的分类。在图中，每个节点显示了 3 个数据，当前节点被选中的划分特征（数字）；当前节点的判断结果（true\false）；当前节点中存在的属性（叶子节点只有一个属性，父亲节点为孩子属性的列表）。（图中第三层特征 9 的叶子节点只有 n、b，而 g 是由于 matlab 自带函数 treeplot 将连接线算重合了）。

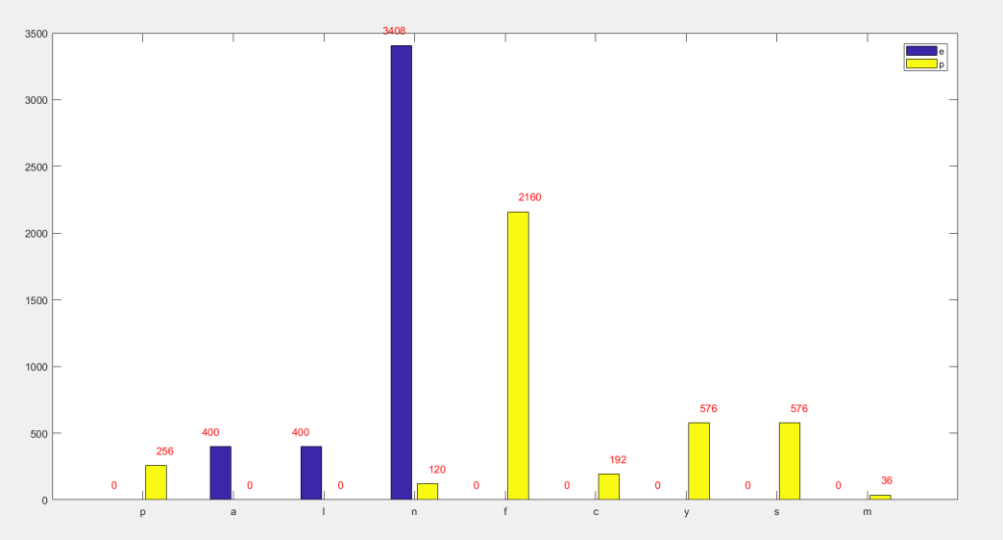
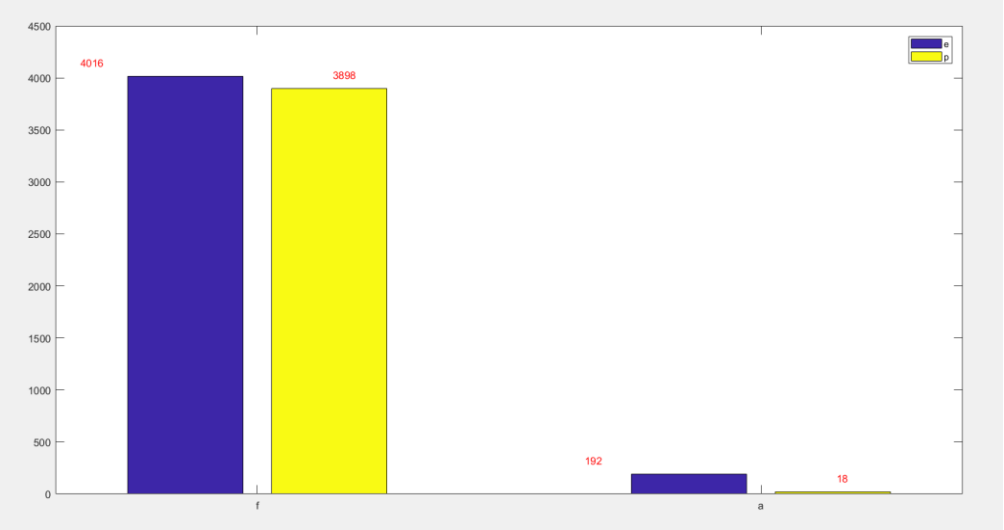
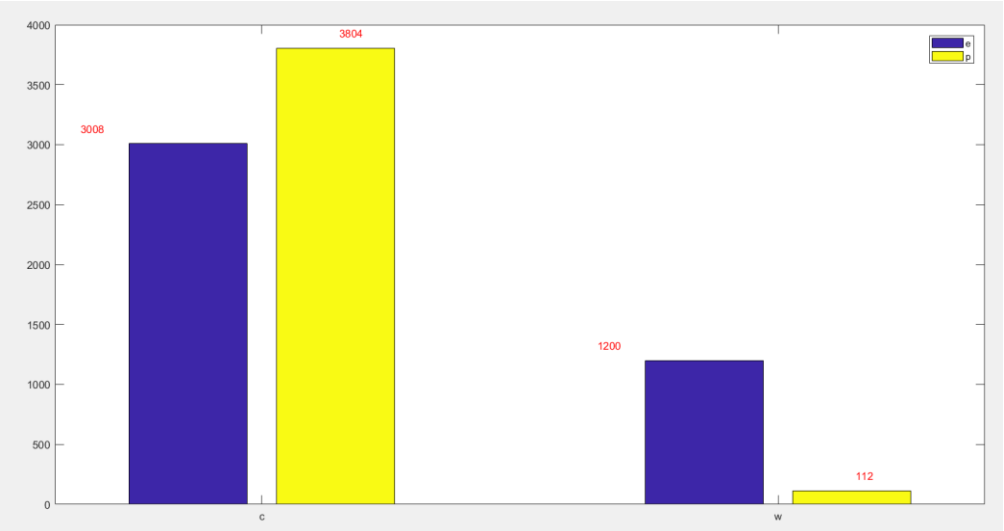
4. 特征描述

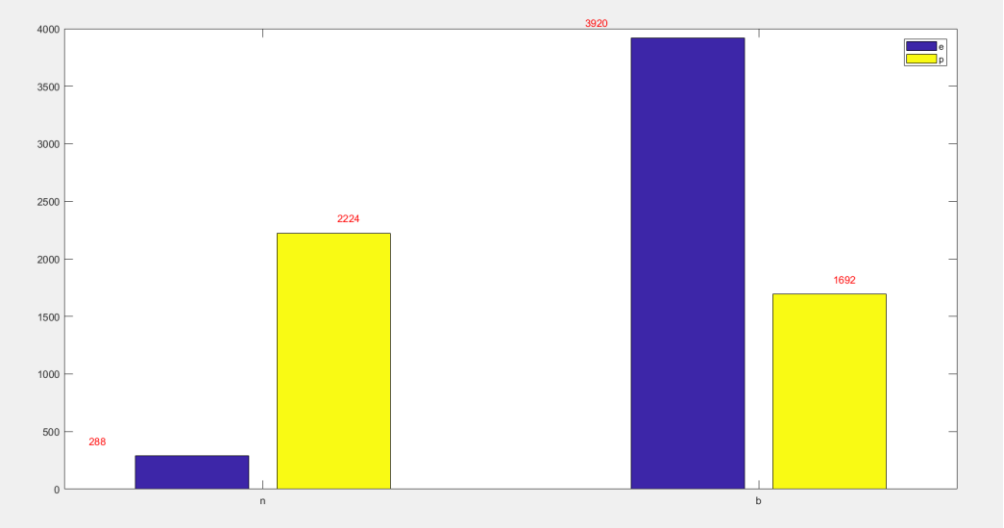
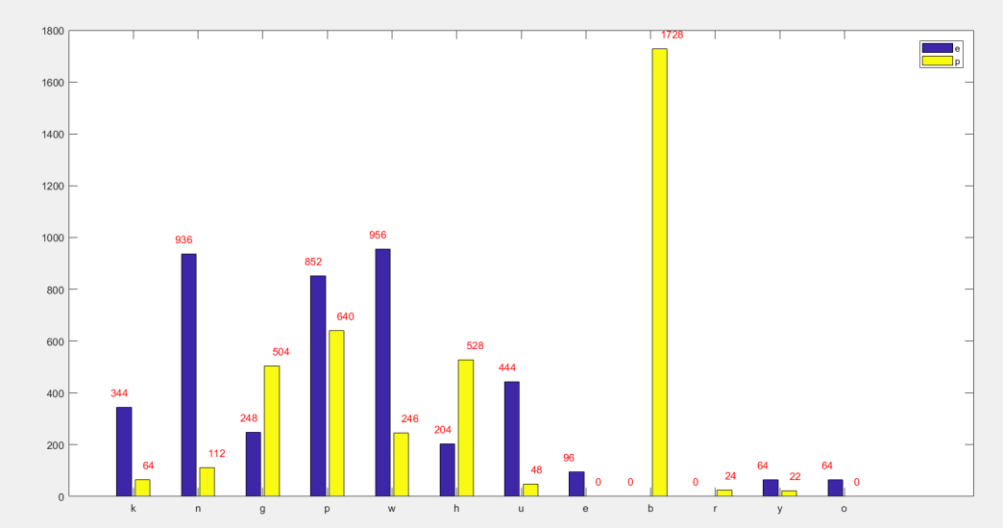
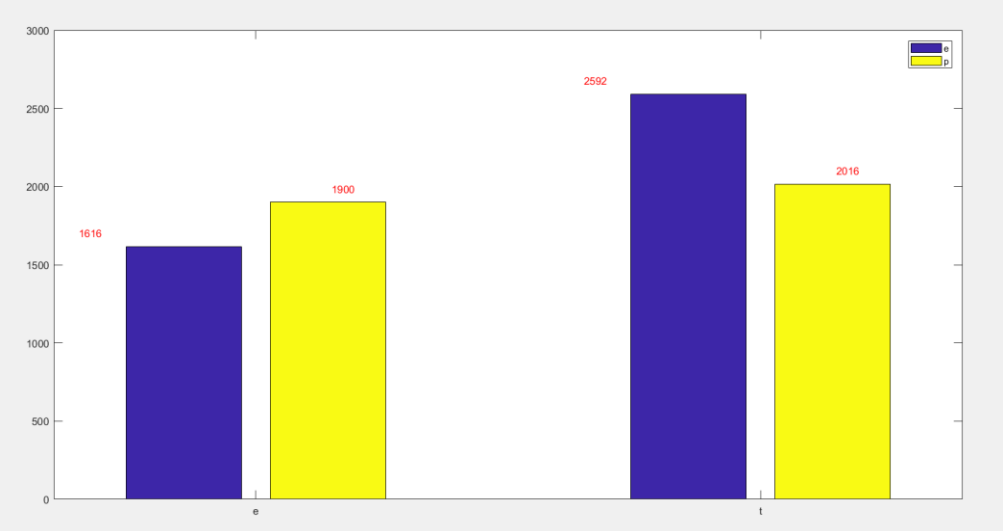
画出两种类别的蘑菇在每一个特征上的分布图：

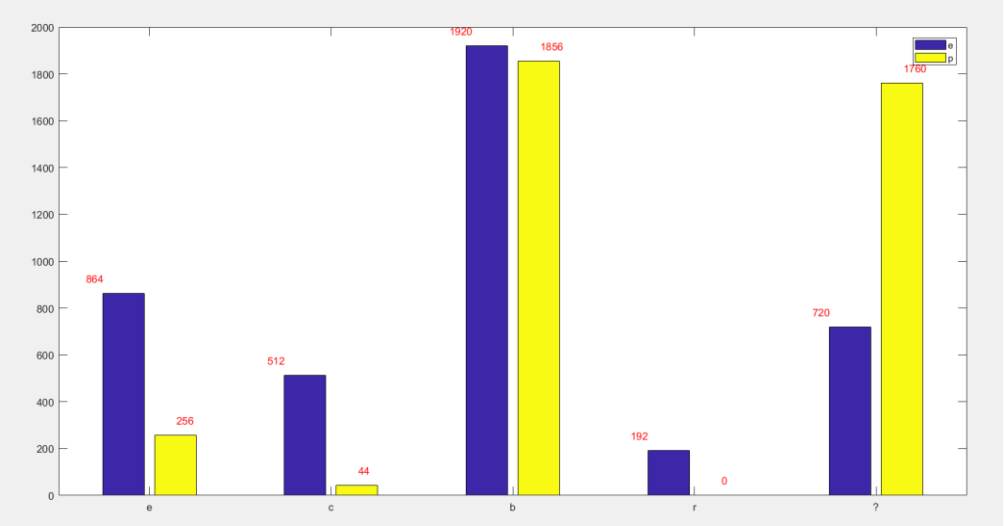
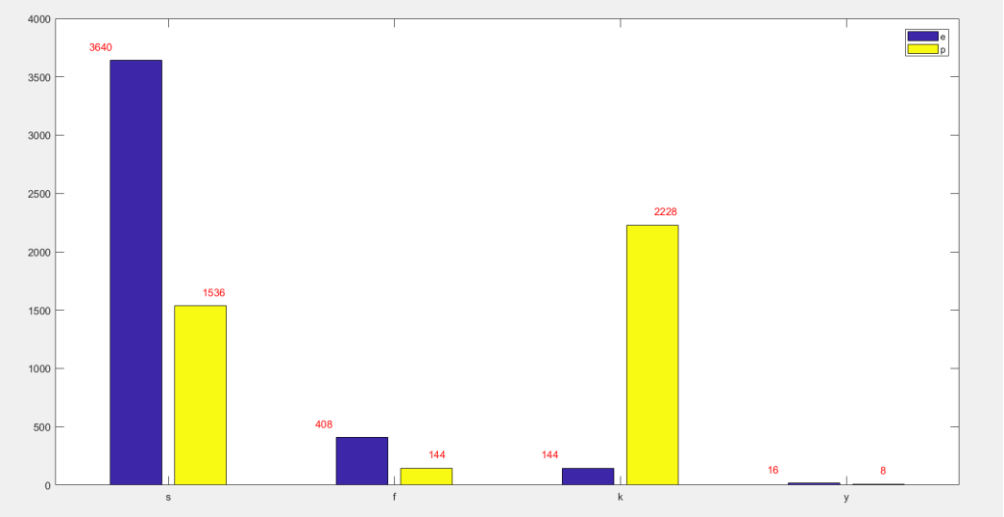
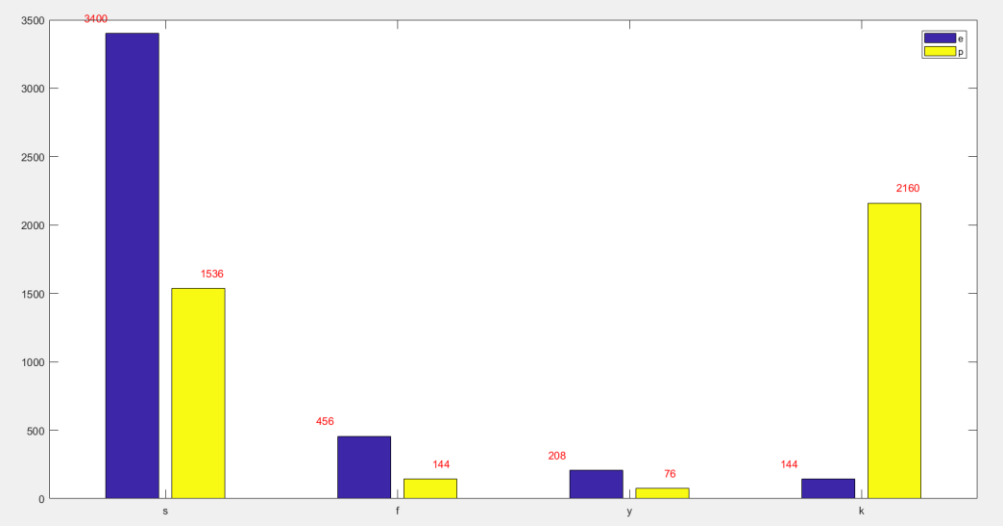
规定：为了与代码统一，不产生奇异，将 22 维特征表示为特征 2 ~ 特征 23。

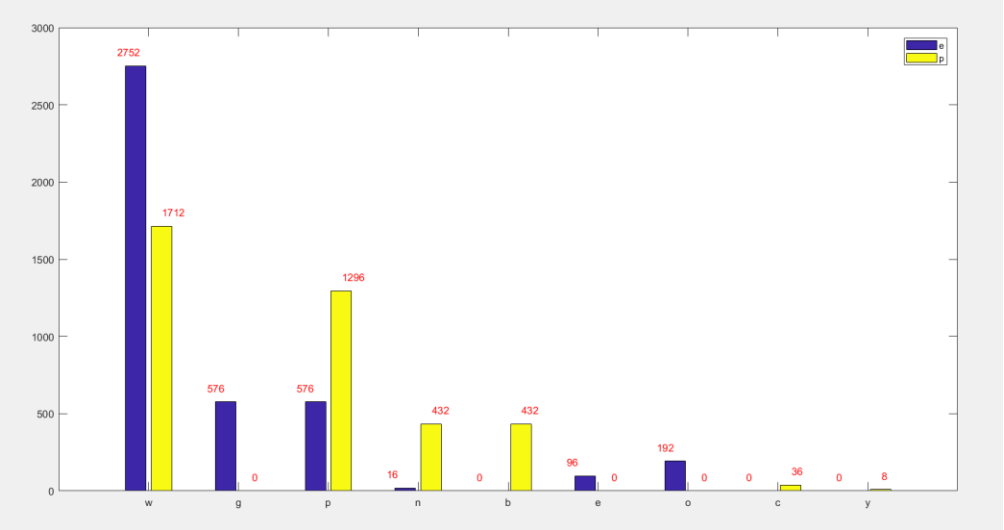
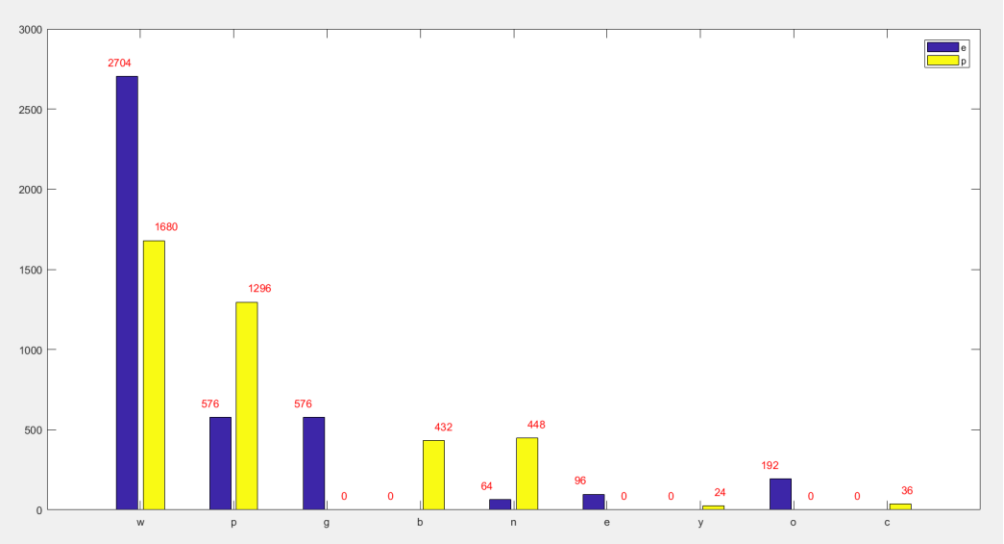
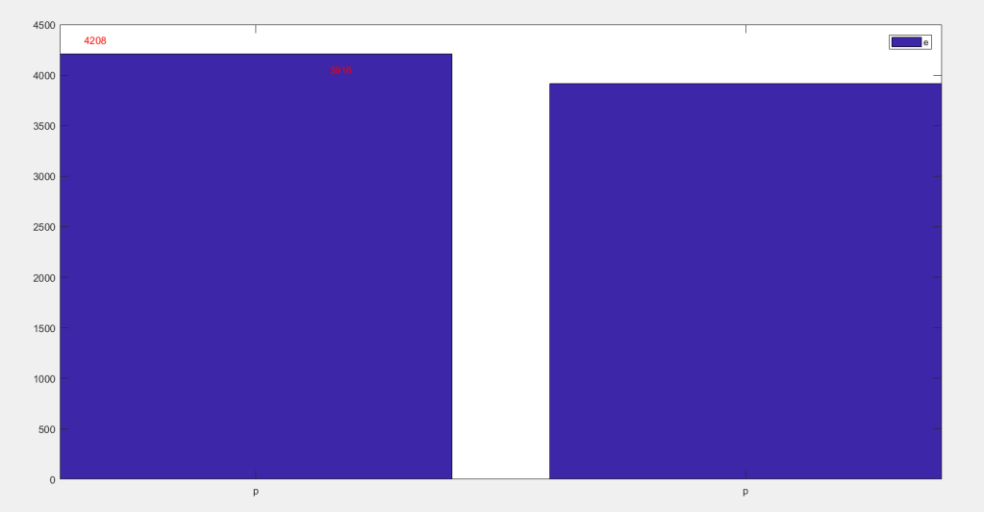
特征 2	

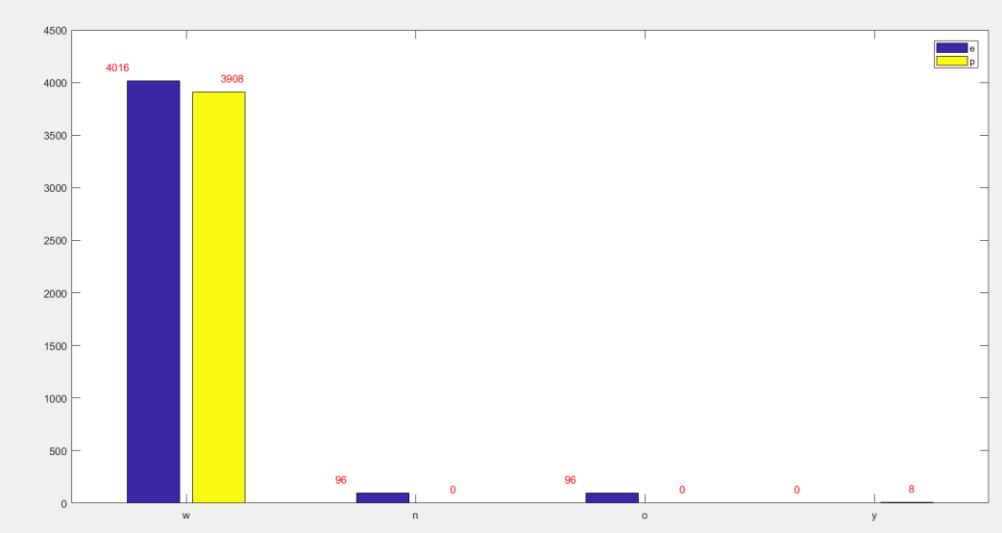
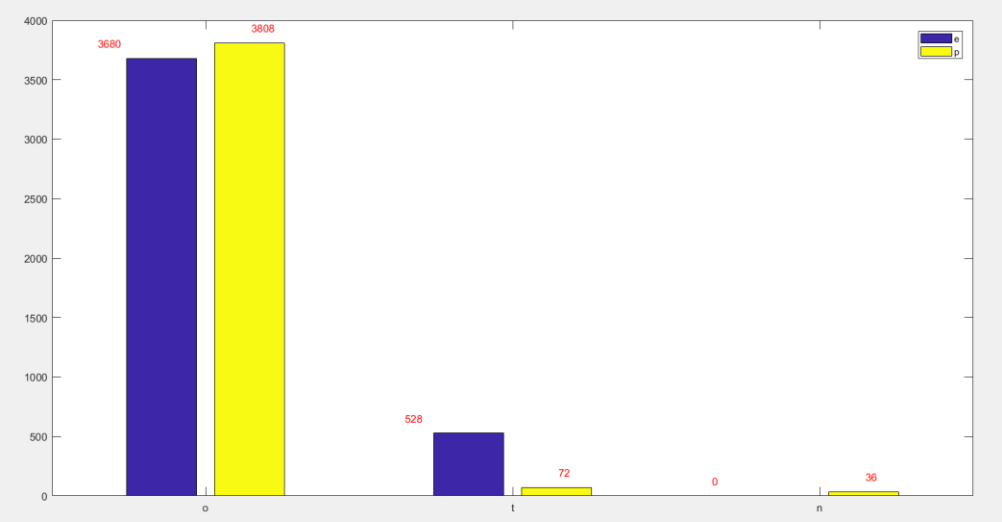
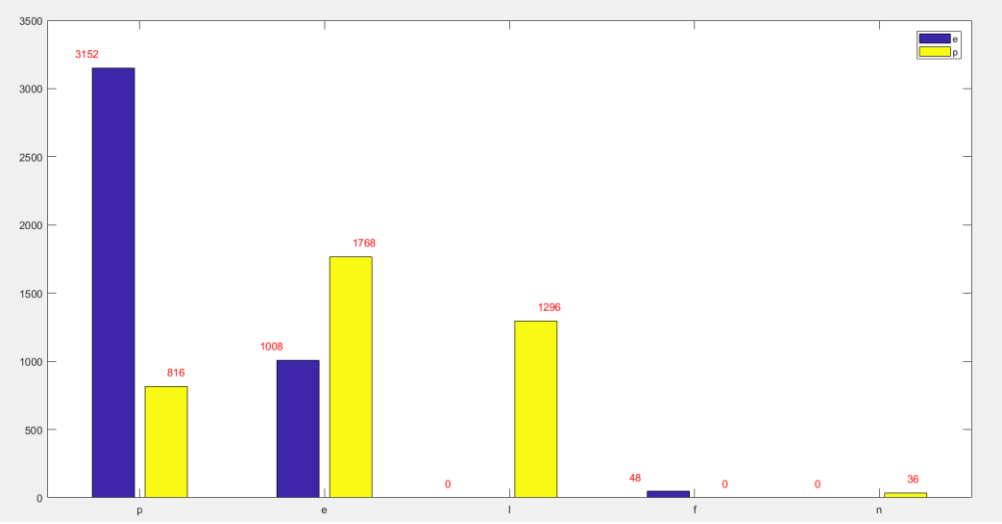
特征3	 1800 1600 1400 1200 1000 800 600 400 200 0 1144 1412 1504 1740 1560 760 0 4 s y f g e p
特征4	 1400 1200 1000 800 600 400 200 0 1264 1020 400 672 720 320 1032 808 624 876 56 88 48 120 16 0 32 12 16 0 n y w g e p b u c r e p
特征5	 3500 3000 2500 2000 1500 1000 500 0 2752 624 1456 3292 t f e p

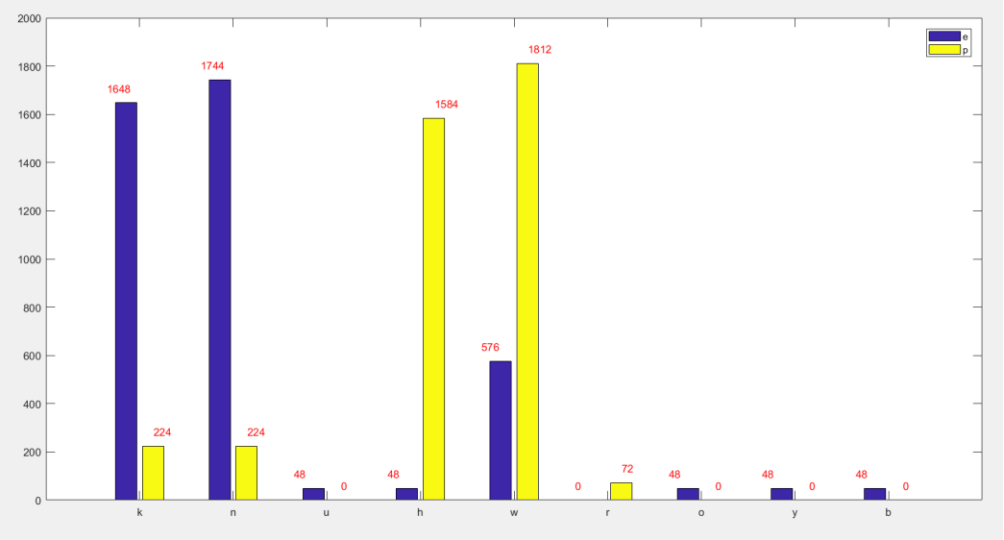
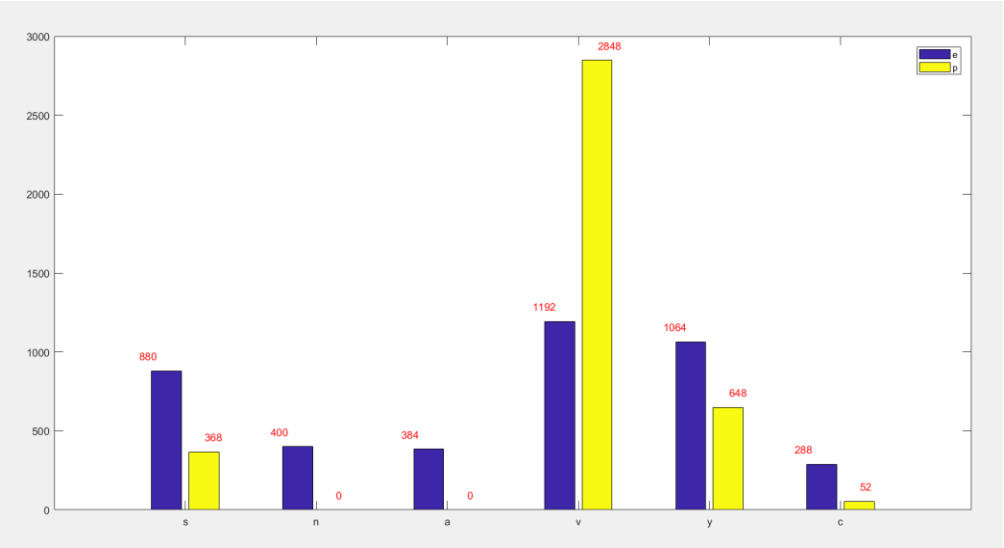
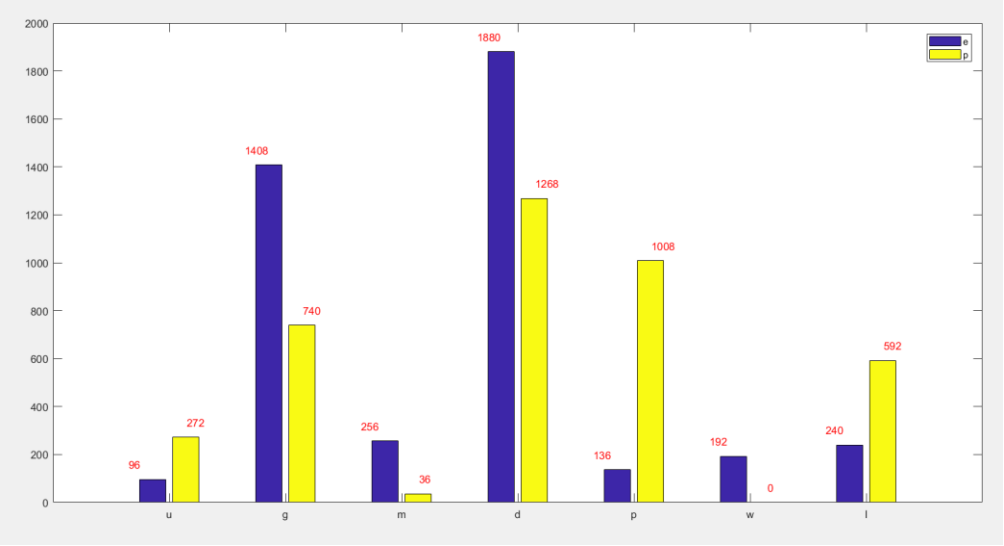
特征6	 3500 3000 2500 2000 1500 1000 500 0 e p 0 256 400 0 400 0 3408 120 0 2160 0 192 0 576 0 576 0 36 p a l n f c y s m
特征7	 4500 4000 3500 3000 2500 2000 1500 1000 500 0 e p 4016 3898 192 18 f a
特征8	 4000 3500 3000 2500 2000 1500 1000 500 0 e p 3008 3804 1200 112 c w

特征9	
特征10	
特征11	

特征12	
特征13	
特征14	

特征15	
特征16	
特征17	

特征18	
特征19	
特征20	

特征21	
特征22	
特征23	

通过对所有特征分布的直观分析，将特征主要分为以下三类: ①不同类别间差异性很强，主要有特征 6、10、13、14、15、16、21； ②不同类别间差异性相对较强，主要有特征 5、9、12、20、22、23； ③不同类别间基本无差别，主要有 2、3、4、7、8、11、17、18、19。其中特征 17 在所有样本中均为单一表现(partial=p)，属于无意义特征。

但这只是第一层观察，在决策树中，当前看似区分性不强的特征，在经过一次、两次决策后，可能会成为下一个区分特征。

5. 训练与测试

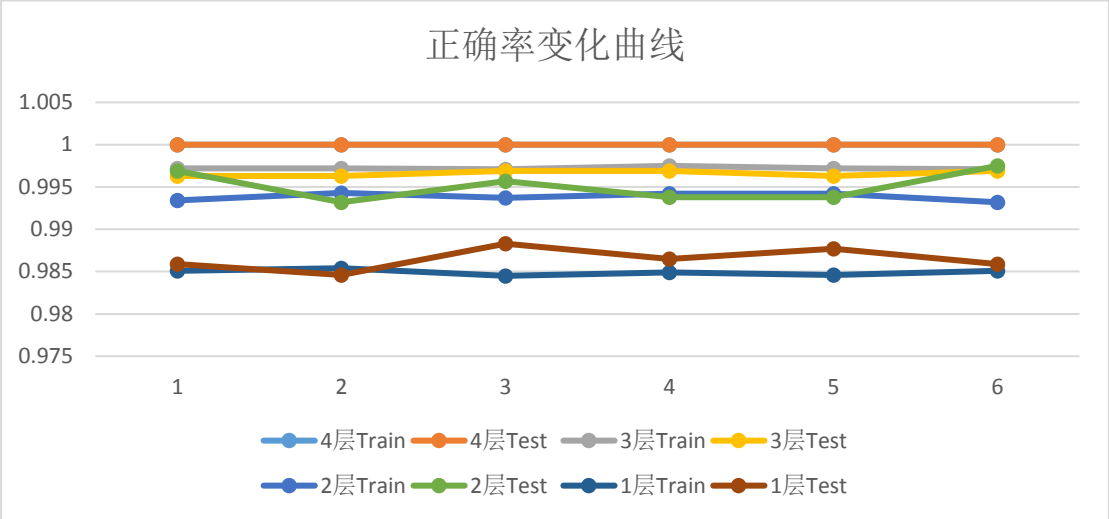
在数据集中，随机选取 80%的数据作为训练集，剩下 20%的数据作为测试集。请画出训练过程中训练集正确率和测试集正确率的变化曲线，还有训练完成后的分类 ROC 曲线。

以下表格展示了 1 棵决策树的训练与测试结果：

	4 层决策树		3 层决策树		2 层决策树		1 层决策树	
	训练集准确度	测试集准确度	训练集准确度	测试集准确度	训练集准确度	测试集准确度	训练集准确度	测试集准确度
1	1	1	0.9972	0.9963	0.9934	0.9969	0.9851	0.9859
2	1	1	0.9972	0.9963	0.9943	0.9932	0.9854	0.9846
3	1	1	0.9971	0.9969	0.9937	0.9957	0.9845	0.9883
4	1	1	0.9975	0.9969	0.9942	0.9938	0.9849	0.9865
5	1	1	0.9972	0.9963	0.9942	0.9938	0.9846	0.9877
6	1	1	0.9971	0.9969	0.9932	0.9975	0.9851	0.9859

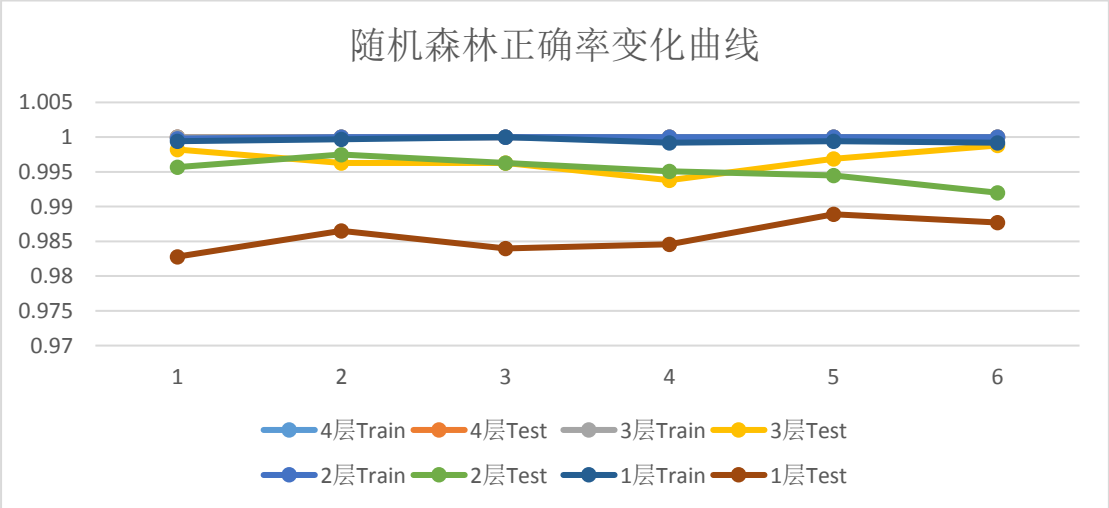
以下折线图显示了正确率变化曲线：

从折线图中，我们可清楚的看到，随着层数的增加，正确率呈现上升趋势。且当建立 4 层决策树时，所有数据全部正确分类。而且，决策树并没有出现过拟合的情况，训练正确率和测试正确率较为接近。整体来看，当采用一个特征进行分类时，即可达到 98.5%的良好效果。



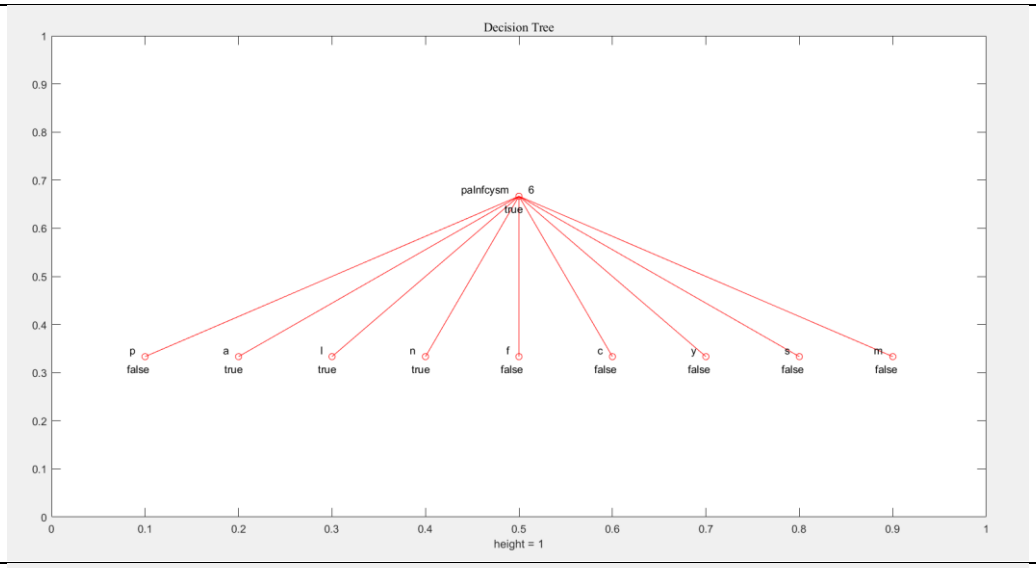
以下表格展示了 7 棵决策树构成的随机森林的训练与测试结果：

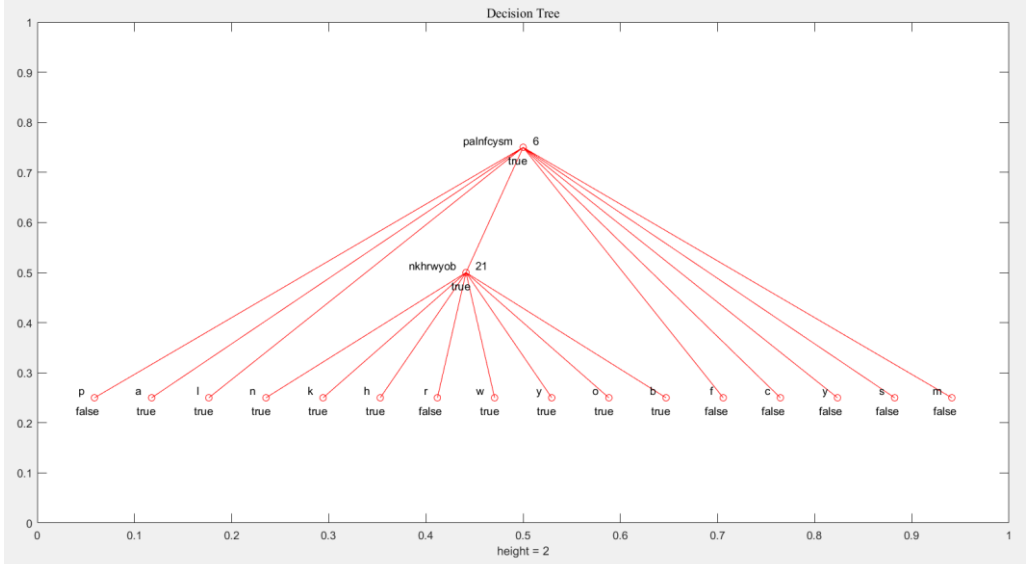
	4 层决策树		3 层决策树		2 层决策树		1 层决策树	
	训练集准确度	测试集准确度	训练集准确度	测试集准确度	训练集准确度	测试集准确度	训练集准确度	测试集准确度
1	1	1	1.0000	0.9982	0.9998	0.9957	0.9994	0.9828
2	1	1	1.0000	0.9963	1.0000	0.9975	0.9997	0.9865
3	1	1	1.0000	0.9963	1.0000	0.9963	1.0000	0.9840
4	1	1	1.0000	0.9938	1.0000	0.9951	0.9992	0.9846
5	1	1	1.0000	0.9969	1.0000	0.9945	0.9994	0.9889
6	1	1	1.0000	0.9988	1.0000	0.9920	0.9992	0.9877

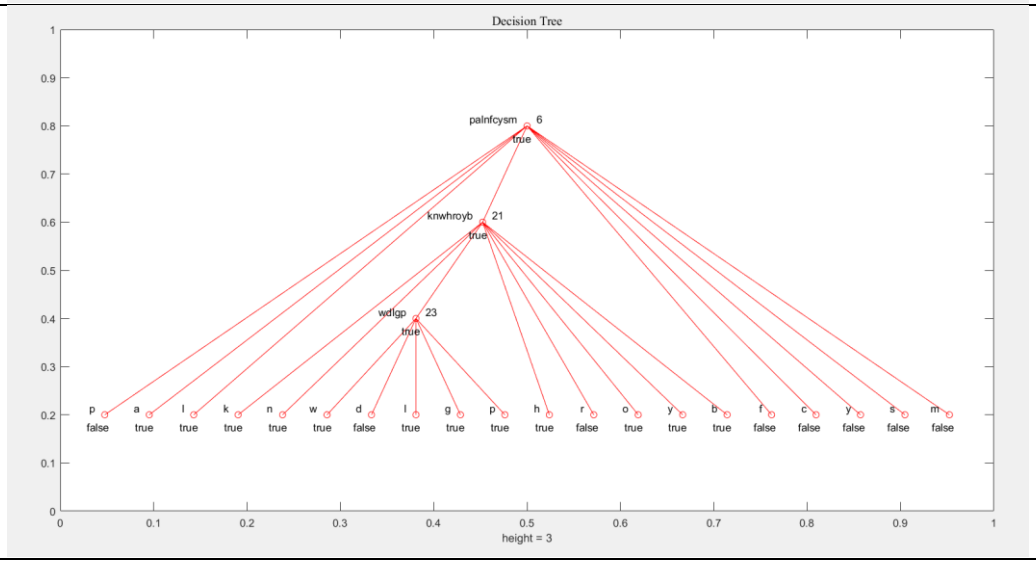


通过与单一决策树的对比，随机森林明显提升了训练集的正确率，几乎达到 1；而测试集正确率提升的并不明显。

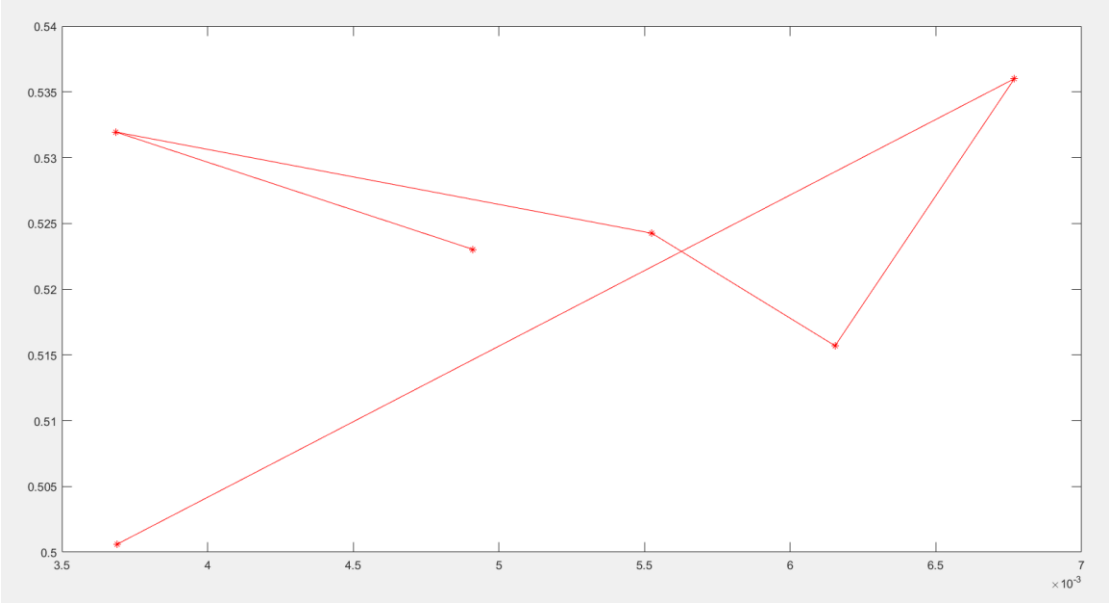
以下为不同层决策树的展示：







单个 2 层决策树的 ROC 曲线：



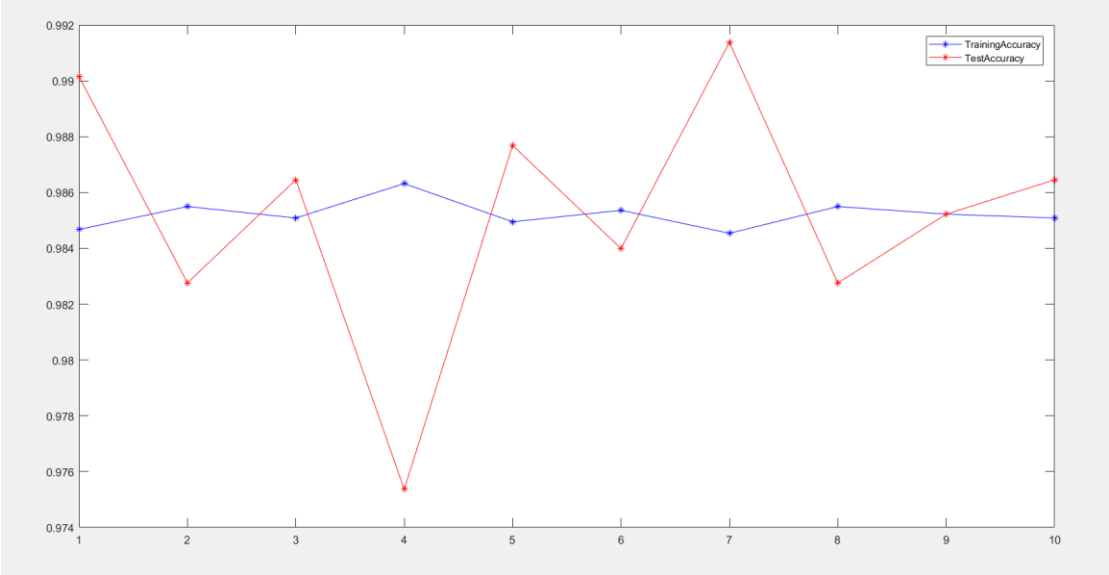
由于决策树不便于画 ROC 曲线，根据定义，画出的曲线大致为一个点，该点大概（0.005，0.525）。如果有比较好的决策树画 ROC 曲线方法，还请及时指教。

6. 交叉验证

在数据集上做 10 倍交叉验证实验，并给出每一次实验的结果包括训练正确率和测试正确率。

- 单个 1 层决策树进行测试：（height = 1）

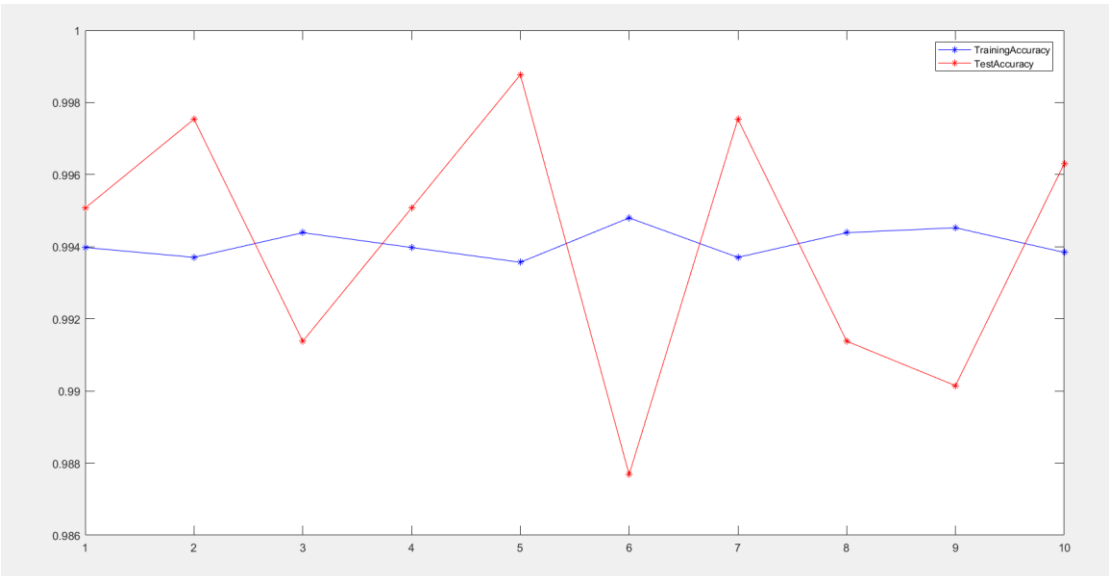
	1	2	3	4	5	6	7	8	9	10
Train	0.9847	0.9855	0.9851	0.9863	0.9849	0.9854	0.9845	0.9855	0.9852	0.9851
Test	0.9901	0.9828	0.9865	0.9754	0.9877	0.9840	0.9914	0.9828	0.9852	0.9865



由图中可见，训练准确率平衡在 98.5%左右，测试准确率略有波动，但也平衡在 98.5%左右。

- 单个 2 层决策树进行测试：（height = 2）

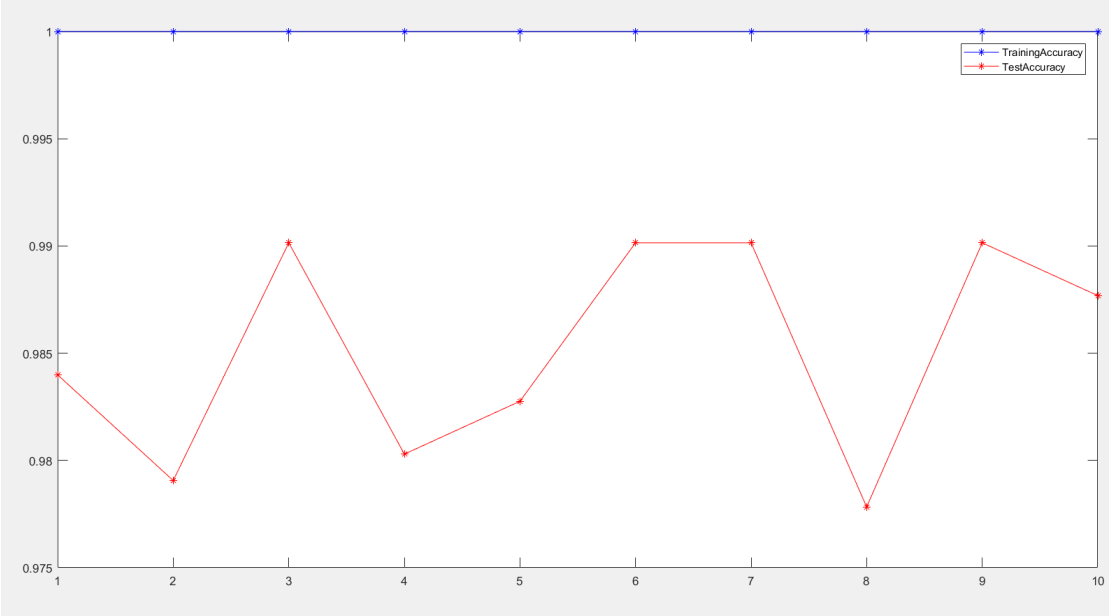
	1	2	3	4	5	6	7	8	9	10
Train	0.9940	0.9937	0.9944	0.9940	0.9936	0.9948	0.9937	0.9944	0.9945	0.9938
Test	0.9951	0.9975	0.9914	0.9951	0.9988	0.9877	0.9975	0.9914	0.9901	0.9963



由图中可见，训练准确率平衡在 99.5%左右，测试准确率略有波动，但也平衡在 99.5%左右。

- 采用随机森林： 7 棵 1 层决策树进行测试：(height = 1)

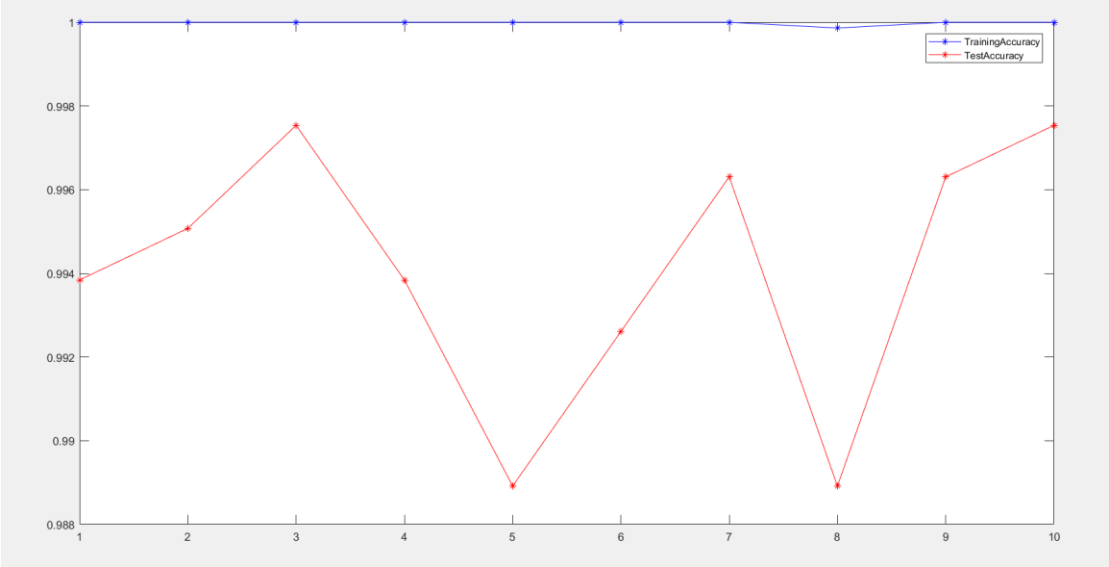
	1	2	3	4	5	6	7	8	9	10
Train	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Test	0.9840	0.9791	0.9901	0.9803	0.9828	0.9901	0.9901	0.9778	0.9901	0.9877



由图中可见，当采用随机森林时，训练准确率明显提升了，可以达到满意的效果。但测试准确率改善的并不明显，仍在 98.5%左右，且有较大波动。

- 采用随机森林： 7 棵 2 层决策树进行测试：(height = 2)

	1	2	3	4	5	6	7	8	9	10
Train	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9999	1.0000	1.0000
Test	0.9938	0.9951	0.9975	0.9938	0.9889	0.9926	0.9963	0.9889	0.9963	0.9975



同样，当采用随机森林时，训练准确率明显提升了，可以达到满意的效果。但测试准确率改善的并不明显，仍在 99.5%左右，且有较大波动。

可能是以下的原因导致：当训练集为单一决策树时，对于每一个训练样本来说，内容是固定的，导致建出来的树是相同的。训练正确率不同的原因：在 80%和 20%的随机抽样中，导致了样本的不同；而交叉验证，由于其本身的原理，轮流做测试集，其余作为训练集，得到的训练集也是不同的。

但为什么随机森林可以将训练正确率提升到 1 呢？因为随机森林会对得到的训练集进行有放回的重新随机抽样，所以针对得到的确定训练集，有放回的抽样可以得到 k 个不同的树。这些树进行投票的机制，几乎可以保证每个训练样本的正确分类。因为那些原本错误的样本在不同的有放回抽样中，权重会变化，进而会被纠正。

但对于测试集效果不明显，是因为测试集的数据并没有经过有放回的训练环节，对于原本错误的样本，森林的判断和一棵树的判断是一致的，投出的票很有可能是一致通过。也许有将其正确划分的树，但达不到半数。

7. 可能的改进

在编写一半时，发现仅仅一个特征 6 即可实现 98%的正确分类。如果可以，希望老师、助教将特征 6 的辨别性降低，以便更好的展现算法区分度。

我觉得可以从决策树的信息增益 gains 进行改进，当特征区分度不是如此明显、层数较多时，可以设定 gains 的阈值，进行先剪枝；或者利用卡方检验考察信息增益的显著性，来控制树的生长。改变当前仅仅依靠层数来控制的局面。

还可以对树进行后剪枝，来防止过学习，比如采取最小代价与复杂性的折中，在不明显改变错误率的情况下，精简树。减少程序的运行时间，当前算法的运行时间较久。

特征提取的话，首先要把非数值特征数值化，比较理想的是正交处理。正交处理后，个人觉得采用其他方法说不定可以取得更好的效果，比如神经网络。决策树的优势在于非数值特征。

8. 总结

本次大作业历时一周，终于完成了所有代码。算法的思路、细节处理在代码中均有详细的注释进行阐述。不得不说，这个作业还是很有挑战的，非数值特征、多个可能的属性、树的绘制、原始数据的统计处理、以及 Matlab 在 cell、char、double 之间的转换处理，均为这次作业带来了不小的难度。当然，这个代码仍有很大的改进之处，有些细节还需调整，以便具有更好的普适性。

通过本次作业，也算是对《人工智能导论》课程的总结提高，系统的学习、实践了决策树以及随机森林的全过程，收获颇丰。