**北京邮电大学软件学院**

**2019 -2020 学年第 一 学期实验报告**

**课程名称： 数据挖掘**

**实验名称： 实验二分类实验**

**实验完成人：**

**姓名：**\_\_黄莹\_\_**学号：**\_\_2017211953\_\_**成绩：**\_\_\_\_\_\_\_\_

**指导教师：**\_\_\_\_\_\_\_\_\_\_\_\_\_牛琨\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**日 期： 2019 年 10 月 28 日**

1. **所要进行分类的属性**

我选取了**触摸屏**（0否，1电阻，2电容）作为**分类标签**，原因如下。

首先，对数据进行**预处理。**由于产品ID与产品型号都无法对数据集进行有效区分，因此我们将产品中的ID与型号属性**删除**，接着对剩余的属性中，不包含数值意义的属性进行**离散化**，即颜色数、市场定位、芯片平台、G网、AP、外观类型、主屏幕尺寸、触摸屏、键盘类型、定位、FM广播、电视、Modem、红外、蓝牙、WLAN、重力感应器、方向感应器、文字输入方法数、智能系统等属性。

通过对所有Nominal（标称）属性的**试分类**后，我发现部分属性由于数据集**不同类别样本数量差距过大**，导致难以进行分类，比如电视属性为0的样本占了总量的98.6%，因此导致算法倾向于预测该属性为0且准确率达到99%；另外还有一些属性**只与数据集中的少数几个属性呈线性关系**，例如G网属性，它的分类结果与频段数量几乎线性相关，当频段数量少于2时，G网属性为0，否则为1，预测准确率达到了99.2%。

而触摸屏属性，三个类别数据集的比例分别为55%，33%，12%，样本比例合适，同时，它的分类决策树与超过15个决策变量相关，增强了对于特殊样本的分类识别能力

1. **不同分类器及其不同分类指标结果**

我没有专门设置检验数据集，为了保证生成的模型的准确性而不至于出现**过拟合**（overfitting）的现象，我采用了**10折交叉验证**（10-fold cross validation）来选择和评估模型。

即将数据集分成**十**份，轮流将其中**九**份作为**训练数据**，**一**份作为**测试数据**，进行试验。每次试验都会得出相应的正确率（或差错率），10次的结果的正确率（或差错率）的平均值作为对算法精度的估计，一般还需要进行多次10折交叉验证，再求其均值，作为对算法准确性的估计。

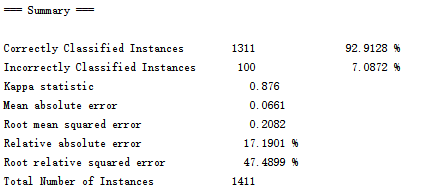
分类结果如下表所示（按正确率由高到低排列）：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 方法名 | weka | 正确率 | kappa系数 | MAE | RMSE | 相对误差 | rrse |
| **Simple Logistic** | **functions.SimpleLogistic** | **94.33%** | **0.9012** | **0.0603** | **0.1735** | **15.66%** | **39.55%** |
| **Random Forest** | **trees.RandomForest** | **94.33%** | **0.9008** | **0.1297** | **0.2035** | **33.71%** | **46.41%** |
| **SMO** | **functions.SMO** | **94.05%** | **0.8963** | **0.2369** | **0.2978** | **61.57%** | **67.91%** |
| **MultilayerPerceptron** | **functions.MultilayerPerceptron** | **93.48%** | **0.8863** | **0.0467** | **0.2** | **12.14%** | **45.60%** |
| **C4.5改进J48** | **trees.J48** | **92.91%** | **0.876** | **0.0661** | **0.2082** | **17.19%** | **47.49%** |
| Bayes Net | bayes.BayesNet | 89.30% | 0.8162 | 0.0778 | 0.2535 | 20.23% | 57.82% |
| 3-NN | lazy.lBk(k=3) | 88.52% | 0.7959 | 0.0996 | 0.243 | 25.90% | 55.41% |
| Logistic | functions.Logistic | 88.09% | 0.7934 | 0.085 | 0.2711 | 22.11% | 61.83% |
| Naive Bayes | bayes.NaiveBayes | 87.53% | 0.7833 | 0.0856 | 0.2623 | 22.24% | 59.82% |
| NN | lazy.lB1 | 86.18% | 0.7565 | 0.093 | 0.3032 | 24.16% | 69.14% |
| 2-NN | lazy.lBk(k=2) | 85.33% | 0.7327 | 0.0944 | 0.2548 | 24.55% | 58.11% |
| KStar | lazy.KStar | 80.30% | 0.6514 | 0.1353 | 0.3358 | 35.16% | 76.57% |

本次实验，我一共尝试了**十二**种分类器，其中分类准确率达到90%以上的有**Simple Logistic**（简单逻辑斯特）,**Random Forest**（随机森林），**SMO**（序列最小优化算法），**MultilayrPerceptron**（多层感知器）,**J48**（决策树）这五个分类器。

1. **不同分类指标的意义**

以J48决策树为例，其输出如下图所示：



其中，第一行为**识别准确率**，即将测试集分类准确的概率，第二行为**差错率**，与识别正确率之和为1，准确率越高，模型也就越好。

第三行为**Kappa系数**，这是一种衡量分类精度的指标。kappa计算结果为-1-1，但通常kappa是落在 0-1 间，可分为五组来表示不同级别的一致性：0.0-0.20为极低的一致性(slight)、0.21-0.40为一般的一致性(fair)、0.41-0.60为中等的一致性(moderate)、0.61-0.80为高度的一致性(substantial)、0.81-1几乎完全一致(almost perfect)。

在本次实验中，我们发现**Kappa系数**与**识别准确率**呈现出了**正相关**关系，识别准确率越高往往Kappa系数值越大，分类精度越高，模型也就越好。

第四行是平**均绝对误差**，又叫平均绝对离差，是所有单个观测值与算术平均值的偏差的绝对值的平均,可以更好地反映预测值误差的实际情况。MAE越小，模型越好。

第五行是**均方根误差**，它衡量观测值与真实值之间的偏差和样本的离散程度的。尽管RMSE更复杂且偏向更高的误差，它仍然是许多模型的默认度量标准，因为用RMSE来定义损失函数是平滑可微的，且更容易进行数学运算。常用来作为机器学习模型预测结果衡量的标准。RMSE越小，模型越好。

第六行是把N次实验的**绝对误差求和**,然后除以实际值与均值之差的求和. 此值越小实验越准确。

第七行为第六行的结果**开根号**，越小结果越准确。

1. **最佳模型的分析及其选择**

由上表的所有数据可知，当准确率、Kappa系数越高，且误差都越低时，模型是越优秀的，因此我将准确率和Kappa系数分别赋予了0.5的权值，赋给四个不同误差的权值为-0.25，最终的结果应该在-1到1之间，可知其结果值越大，结果越优秀。

结果如下表所示：

|  |  |
| --- | --- |
| Simple Logistic | 0.7257705 |
| Random Forest | 0.63846225 |
| SMO | 0.461018 |
| MultilayerPerceptron | 0.704522 |
| C4.5改进J48 | 0.672289 |
| Bayes Net | 0.576627 |

|  |  |
| --- | --- |
| 3-NN | 0.55160425 |
| Logistic | 0.5382975 |
| Naive Bayes | 0.53715225 |
| NN | 0.4768415 |
| 2-NN | 0.49906325 |
| KStar | 0.33009 |

于是我最终选取**简单逻辑斯特（Simple Logistic)**作为**最优模型。**

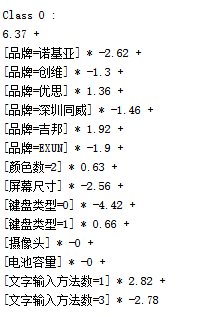
在这个模型中，模型**准确率**达到十二个分类器中的**最高**94.33%，其**kappa系数**也是十二个模型中**最高**的，其**RMSE和RRSE**为十二个模型中**最小的**，其**MSE和相对误差**是十二个模型中**仅次于**多层感知器的。

1. **模型解释**

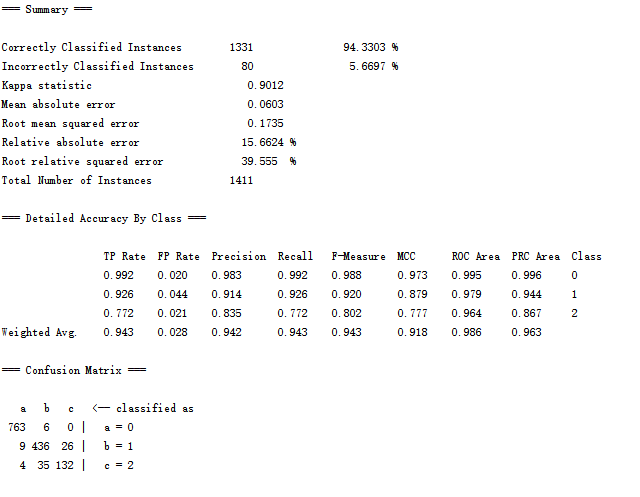
简单逻辑斯特回归解决多分类问题是靠创建多个二分类器，针对一个测试样本，我们需要找到这些分类函数输出值最大的那一个，即为测试样本的标记。

例如在本实验中，触摸屏被分为了三个类别，那么算法会创建三个多分类器，对每一个测试样本，分别计算它在不同分类器下的取值，并以取值最大的分类器作为分类结果。

三个分类器如下所示：



该模型具体指标如下图所示：



其中**Summary**部分已经在前面解释过。

**Detailed**部分则包括了三个分类的不同准确率信息。

**TP rate**是正样本预测结果数/正样本实际数；**FP rate**是被预测为正的负样本结果数/负样本实际数；**Precision**是预测为正的正样本结果数/被预测为正的样本数；**Recall**是被预测为正的正样本结果数/正样本实际数；**F-Measure**为精确值和召回率的调和均值。

**MCC**是马修斯相关系数，衡量不平衡数据集的指标比较好，公式是

IMG_256

**ROC**曲线的横坐标为false positive rate（FPR），纵坐标为 true positive rate（TPR），当测试集中的正负样本的分布变化的时候，ROC曲线能够保持不变，ROC曲线下方面积越大越好。

**PRC**曲线，在正负样本分布得极不均匀(highly skewed datasets)的情况下，PRC比ROC能更有效地反应分类器的好坏，同样其曲线下方区域越大越好。

**Confusion Metrix即混淆矩阵，**其中第一行的样本真实值均为a，第二行样本真实值均为b，第三行样本真实值均为c，第一列是被预测为a的样本数量，第二列是被预测为b的样本数量，第三列是被预测为c的样本数量。