《数据挖掘》实验

班级 计科11606 姓名 张仁兵 成绩 \_\_\_\_\_\_\_

2. 实验二、分类器构建与评测

2.0 实验内容与目的

（1）下载并安装Weka的稳定版本，例如Weka 3.8，可选择带JRE或者不带JRE的版本（http://www.cs.waikato.ac.nz/~ml/weka/downloading.html）。

（2）以weather.nominal数据集为基础，构建k近邻、决策树、朴素贝叶斯分类器，从训练数据中生成预测模型，采用交叉验证方式对其生成模型进行评测，分析评测结果，并将决策树模型可视化。

操作方法与要点如下：

2.1 安装并运行Weka：

* 运行Weka安装包，进行安装。
* 运行Weka：“开始-程序-Weka”，将启动Weka图形界面选择程序（GUI chooser）。
* 点击 “Explorer”打开主界面。

2.2 装载数据集：

* Preprocess面板中，点击<Open file…>，选择Weka安装目录中data文件夹下的weather.nominal.arff文件。
* 观察“Current relation”中显示的数据集信息，回答下面的问题：

**该数据集的关系名称是：weather.symbolic\_。该数据集有 \_\_\_14\_\_\_\_ 条实例，\_\_\_\_5\_\_\_ 个属性。**

* 在Attributes部分中选择temperature属性，观察Selected attribute版块中的信息，回答下面的问题：

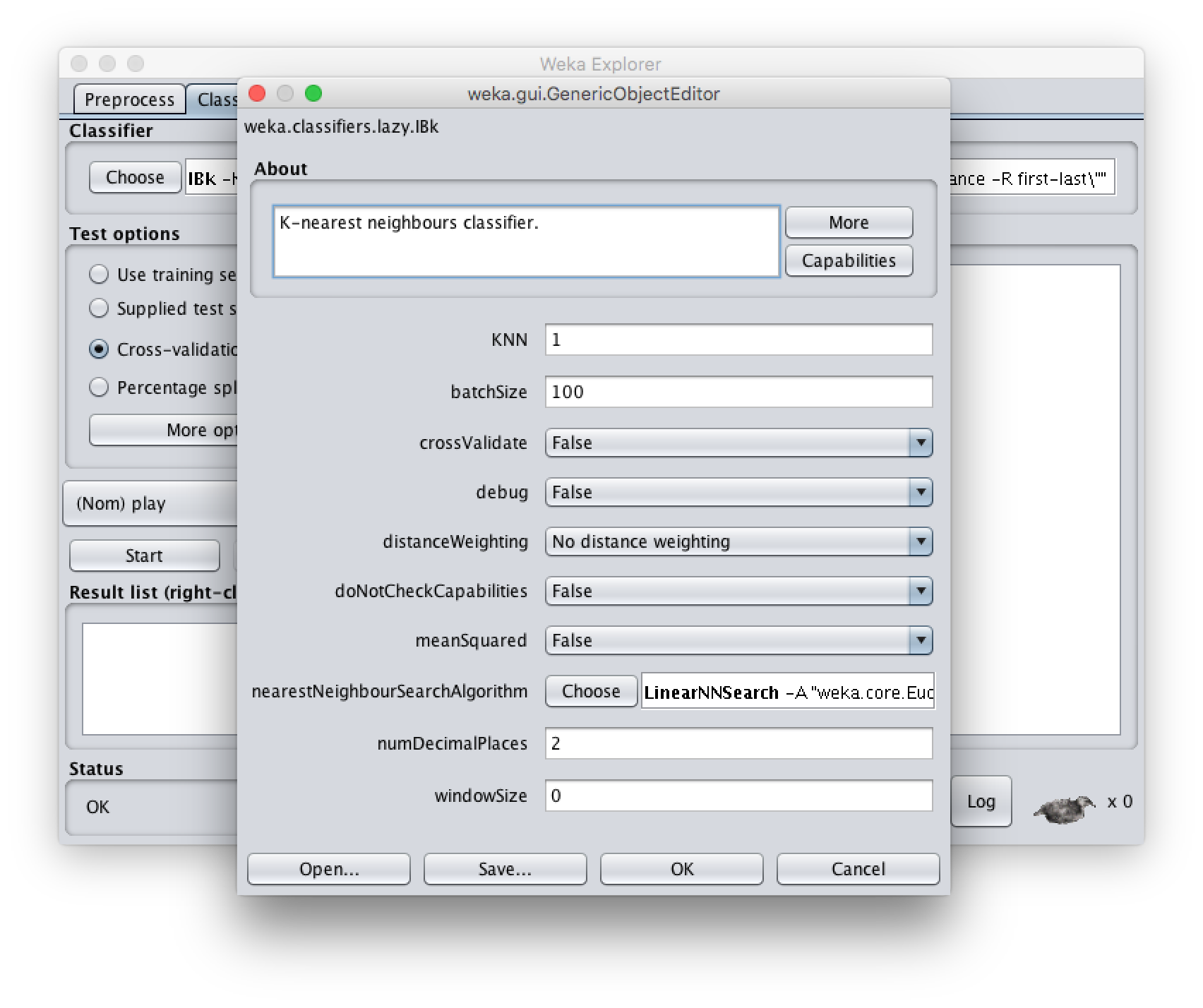
**该属性的类型是：\_\_Nominal\_\_\_\_\_\_，有\_\_3\_\_\_\_\_ 种取值，分别是\_\_hot、mild、cool\_\_\_\_\_\_\_\_。**

2.3 使用分类面版：

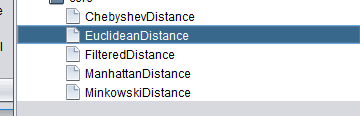
* 点击Classify面板，在Classifier部分点击Choose，将显示分类器列表。Weka中的分类器已按照其特性进行了分组。接下来要构建的三种分类器的分组分别是：lazy->IBk（k近邻）、trees->J48（决策树）、bayes->NaiveBayes（朴素贝叶斯）。
* 左侧的Test options版块中提供了各种测评方法的选项。右侧的 Classifier output版块会显示分类器构建过程的输出信息。

2.4 构建k近邻分类器

* 选择lazy->IBk作为分类器，点击IBk的参数部分将打开Weka的通用对象编辑器（GenericObjectEditor），如下图所示。点击“More”将显示当前算法的详细信息，点击“Capabilities”将显示该分类器的适用范围。



* k近邻分类器的核心参数有两个：搜索方法和距离公式。搜索方法决定了寻找k近邻的效率，距离公式则定义了实例之间相似度（或者距离）的计算方法。在kNN中，这两个参数都是通过neareastNeighbourSearchAlgorithm这个参数来设定的。缺省情况下，kNN将采用线性搜索（即LinearNNSearch）和欧式距离（EucledianDistance），其中距离公式作为搜索方法的参数进行设定。点击LinearNNSearch的参数配置，点击distanceFunction参数。**请回答：除了欧式距离之外，Weka还提供了哪些距离公式？（可以截图或者文字回答）**



* 构建基础kNN：点击Start，以缺省参数构建kNN，观察右侧的Classifier output，**此时kNN的准确率是多少？**

57.1429 %

* 不同测试方式的影响：注意，此时我们采用的是交叉验证（Cross-validation）的测试方式。换成“Use training set”的测试方式，再构建kNN，**此时的准确率是多少？请问该准确率可信吗？为什么？**

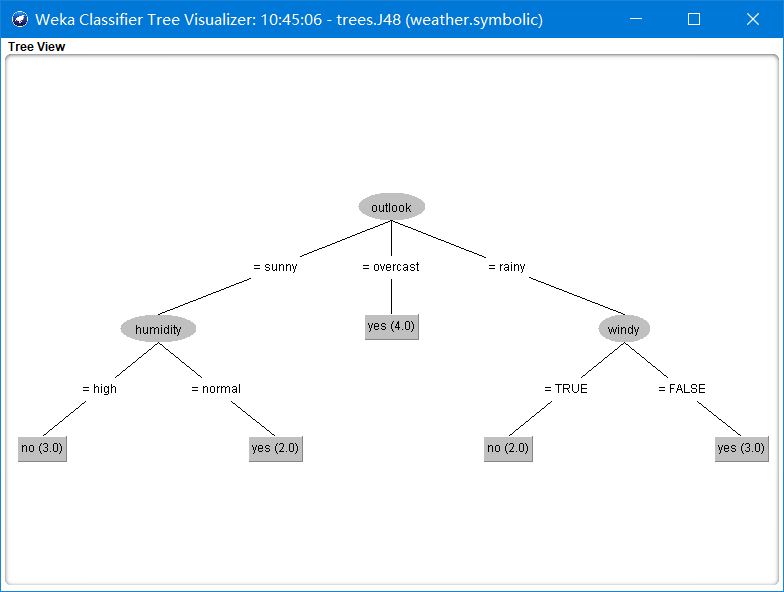
100 % 不可信，这种方式得到的结果往往好于对未知实例的测试结果，不能反映分类器的泛化能力。

* + 不同k值的影响：切换回交叉验证的测试方式，点击IBk，对其中的kNN选项（即k值）进行设置，尝试不同的k值，**准确率最高的k值是\_\_\_k>1\_\_\_\_\_\_，准确率（Correctly classified instances）是\_\_\_ 64.2857 % \_\_\_。**

2.5 构建决策树分类器

* 选择trees>J48作为分类器，点击Start，以缺省参数构建基础分类器。观察分类器输出，并与kNN分类器的输出进行对比。
* 决策树最大的优势之一在于其可视化和易于理解。在Result list版块中右键点击J48的结果集，选择“Visualize tree”，**将可视化后的决策树截图粘贴到下面。**请回答：基于该决策树的结构，**下面实例的预测类别标识将是\_\_no（3.0）\_\_\_\_**。

outlook = sunny, temperature = cool, humidity = high, windy = TRUE



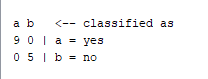
* 打开J48的参数编辑窗口，更改叶子节点的最小实例数量（即minNumObj）参数。该参数的取值约大，意味着叶子节点中必须包含的实例数量越多。缺省取值为2，尝试不同的取值，观察产生的树结构发生了哪些变化。

2.6 构建朴素贝叶斯分类器

* 选择bayes->NaiveBayes作为分类器，点击Start，以缺省参数构建基础的朴素贝叶斯分类器。观察分类器输出，并与kNN分类器和决策树分类器的输出进行对比。
* 在分类器输出的Classifier model部分，显示了各个属性及其取值在各个类别取值中的分布情况。

2.7 多个分类器的比较

* 每个分类器的测评结果中都有混淆矩阵（Confusion Matrix），其中的行代表实例的原始类别，列代表实例的预测类别。**那么，理想情况下的混淆矩阵应该是什么样的？**



最理想的结果就是所有的数据都在对角线上，那么说明分类精度最高，为100%

* 以交叉验证为测试方法，以缺省参数构建kNN分类器和朴素贝叶斯分类器，以minNumObj=1，其他参数均为默认取值的配置构建决策树分类器，分别得到三个分类器。
* 比较这三个分类器的混淆矩阵。**对类别yes而言，最准确的分类器是哪一个？对类别no而言，最准确的分类器是哪一个？**

对于类别yes：tree.J48的准确率为6/9

Lazy.IBK的准确率为6/9

Bayes.NaiveBayes的准确率为7/9

所以最准确的为Bayes.NaiveBayes最精准

对于类别no：tree.J48d 准确率为3/5

Lazy.IBK的准确率为2/5

Bayes.NaiveBayes的准确率为1/5

所以最准确的是tree.J48,准确率为3/5