1. 简介

数据挖掘、机器学习这些字眼，在一些人看来，是门槛很高的东西。诚然，如果做算法实现甚至算法优化，确实需要很多背景知识。但事实是，绝大多数数据挖掘工程师，不需要去做算法层面的东西。他们的精力，集中在特征提取，算法选择和参数调优上。那么，一个可以方便地提供这些功能的工具，便是十分必要的了。而 weka，便是数据挖掘工具中的佼佼者。

Weka 的全名是怀卡托智能分析环境（ Waikato Environment for Knowledge Analysis ），是一款免费的，非商业化的，基于 JAVA 环境下开源的机器学习以及数据挖掘软件。它和它的源代码可在其官方网站下载。有趣的是，该软件的缩写 WEKA 也是 New Zealand 独有的一种鸟名， 而 Weka 的主要开发者同时恰好来自新西兰的 the University of Waikato。（本段摘自百度百科） 。

Weka 提供的功能有数据处理，特征选择、分类、回归、聚类、关联规则、可视化等。本文将对 Weka 的使用做一个简单的介绍，并通过简单的示例，使大家了解使用 weka 的流程。本文将仅对图形界面的操作做介绍，不涉及命令行和代码层面的东西。

2. 安装

Weka 的官方地址是 http://www.cs.waikato.ac.nz/ml/weka/ 。点开左侧 download 栏，可以进入下载页面，里面有 windows， mac os， linux 等平台下的版本，我们以 windows 系统作为示例。目前稳定的版本是 3.6。

如果本机没有安装 java，可以选择带有 jre 的版本。下载后是一个 exe 的可执行文件，双击进行安装即可。

安装完毕，打开启动 weka 的快捷方式，如果可以看到下面的界面，那么恭喜，安装成功了。

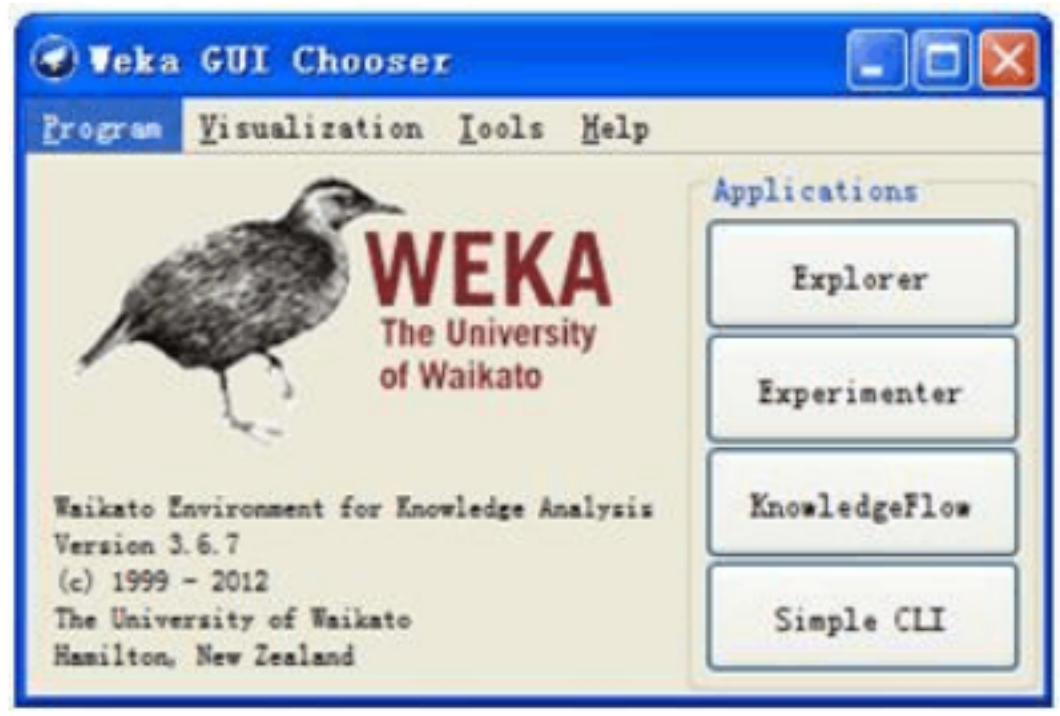

图 2.1 weka 启动界面

窗口右侧共有 4 个应用，分别是

1） Explorer

用来进行数据实验、 挖掘的环境， 它提供了分类， 聚类，关联规则， 特征选择， 数据可视化的功能。（An environment for exploring data with WEKA ）

2） Experimentor

用来进行实验，对不同学习方案进行数据测试的环境。 （An environment for performing experiments and conducting statistical tests between learning schemes.）

3） KnowledgeFlow

功能和 Explorer 差不多，不过提供的接口不同，用户可以使用拖拽的方式去建立实验方案。另外，它支持增量学习。（This environment supports essentially the same functions as the Explorer but with a drag-and-drop interface. One advantage is that it supports incremental learning. ）

4） SimpleCLI

简单的命令行界面。 （Provides a simple command-line interface that allows direct execution of WEKA commands for operating systems that do not provide their own command line interface. ）

3. 数据格式

Weka 支持很多种文件格式，包括 arff、 xrff 、 csv，甚至有 libsvm 的格式。其中， arff 是最常用的格式，我们在这里仅介绍这一种。

Arff 全称是 Attribute-Relation File Format ，以下是一个 arff 格式的文件的例子。

```
%
%   Arff file example
%
@relation   ‘ labor-neg-data ’
@attribute   ‘ duration ’   real
@attribute   ‘ wage-increase-first-year ’   real
@attribute   ‘ wage-increase-second-year ’   real
@attribute   ‘ wage-increase-third-year ’   real
@attribute   ‘ cost-of-living-adjustment   ’   { ‘ none ’ ,'tcf ’ ,'tc ’ }
@attribute   ‘ working-hours ’   real
@attribute   ‘ pension ’   { ‘ none ’ ,'ret_allw ’ ,'empl_contr ’ }
@attribute   ’ standby-pay ’   real
@attribute   ’ shift-differential   ’   real
@attribute   ‘ education-allowance ’   { ‘ yes ’ ,'no ’ }
@attribute   ’ statutory-holidays ’   real
@attribute   ‘ vacation ’   { ‘ below_average ’ ,'average ’ ,'generous ’ }
@attribute   ‘ longterm-disability-assistance   ’   { ‘ yes ’ ,'no ’ }
@attribute   ‘ contribution-to-dental-plan   ’   { ‘ none ’ ,'half ’ ,'full   ’ }
@attribute   ‘ bereavement-assistance’   { ‘ yes ’ ,'no ’ }
@attribute   ‘ contribution-to-health-plan   ’   { ‘ none ’ ,'half ’ ,'full   ’ }
@attribute   ‘ class ’   { ‘ bad ’ ,'good ’ }
@data
1,5,?,?,?,40,?,?,2,?,11,’ average ’ ,?,?, ’ yes ’ ,?, ’ good ’
2,4.5,5.8,?,?,35, ’ ret_allw   ’ ,?,?, ’ yes ’ ,11, ’ below_average ’ ,?, ’ full   ’ ,?, ’ full   ’ ,'good ’
?,?,?,?,?,38,’ empl_contr   ’ ,?,5,?,11, ’ generous ’ ,'yes ’ ,'half   ’ ,'yes ’ ,'half   ’ ,'good ’
3,3.7,4,5, ’ tc ’ ,?,?,?,?, ’ yes ’ ,?,?,?,?, ’ yes ’ ,?, ’ good ’
3,4.5,4.5,5,?,40,?,?,?,?,12, ’ average ’ ,?, ’ half   ’ ,'yes ’ ,'half   ’ ,'good ’
2,2,2.5,?,?,35,?,?,6, ’ yes ’ ,12, ’ average ’ ,?,?,?,?,’ good ’
3,4,5,5, ’ tc ’ ,?, ’ empl_contr   ’ ,?,?,?,12, ’ generous ’ ,'yes ’ ,'none ’ ,'yes ’ ,'half   ’ ,'good ’
3,6.9,4.8,2.3,?,40,?,?,3,?,12, ’ below_average ’ ,?,?,?,?,’ good ’
2,3,7,?,?,38,?,12,25, ’ yes ’ ,11, ’ below_average ’ ,'yes ’ ,'half   ’ ,'yes ’ ,?, ’ good ’
1,5.7,?,?, ’ none ’ ,40, ’ empl_contr   ’ ,?,4,?,11, ’ generous ’ ,'yes ’ ,'full   ’ ,?,?, ’ good ’
3,3.5,4,4.6, ’ none ’ ,36,?,?,3,?,13, ’ generous ’ ,?,?, ’ yes ’ ,'full   ’ ,'good ’
2,6.4,6.4,?,?,38,?,?,4,?,15,?,?,’ full   ’ ,?,?, ’ good ’
2,3.5,4,?, ’ none ’ ,40,?,?,2, ’ no ’ ,10, ’ below_average ’ ,'no ’ ,'half   ’ ,?, ’ half   ’ ,'bad ’
```

这个例子来自于 weka 安装目录 data 文件下的 labor.arff 文件， 来源于加拿大劳资谈判的案例，它根据工人的个人信息，来预测劳资谈判的最终结果。

文件中，“ % ”开头的是注释。剩余的可以分为两大部分，头信息（ header information ）和数据信息（ data information ）。

头信息中，“ @relation ”开头的行代表关系名称，在整个文件的第一行（除去注释） 。格式是

@relation <relation-name>

“ @attribute ”开头的代表特征，格式是

@attribute <attribute-name> <datatype>

attribute-name 是特征的名称，后面是数据类型，常用数据类型有以下几种

1） numeric，数字类型，包括 integer（整数）和 real（实数）

2） nominal，可以认为是枚举类型，即特征值是有限的集合，可以是字符串或数字。

3） string，字符串类型，值可以是任意的字符串。

从“ @data ”开始，是实际的数据部分。每一行代表一个实例，可以认为是一个特征向量。各个特征的顺序与头信息中的 attribute 逐个对应，特征值之间用逗号分割。在有监督分类中，最后一列是标注的结果。

某些特征的数值如果是缺失的，可以用“ ？ ”代替。

数据挖掘流程

使用 weka 进行数据挖掘的流程如下图

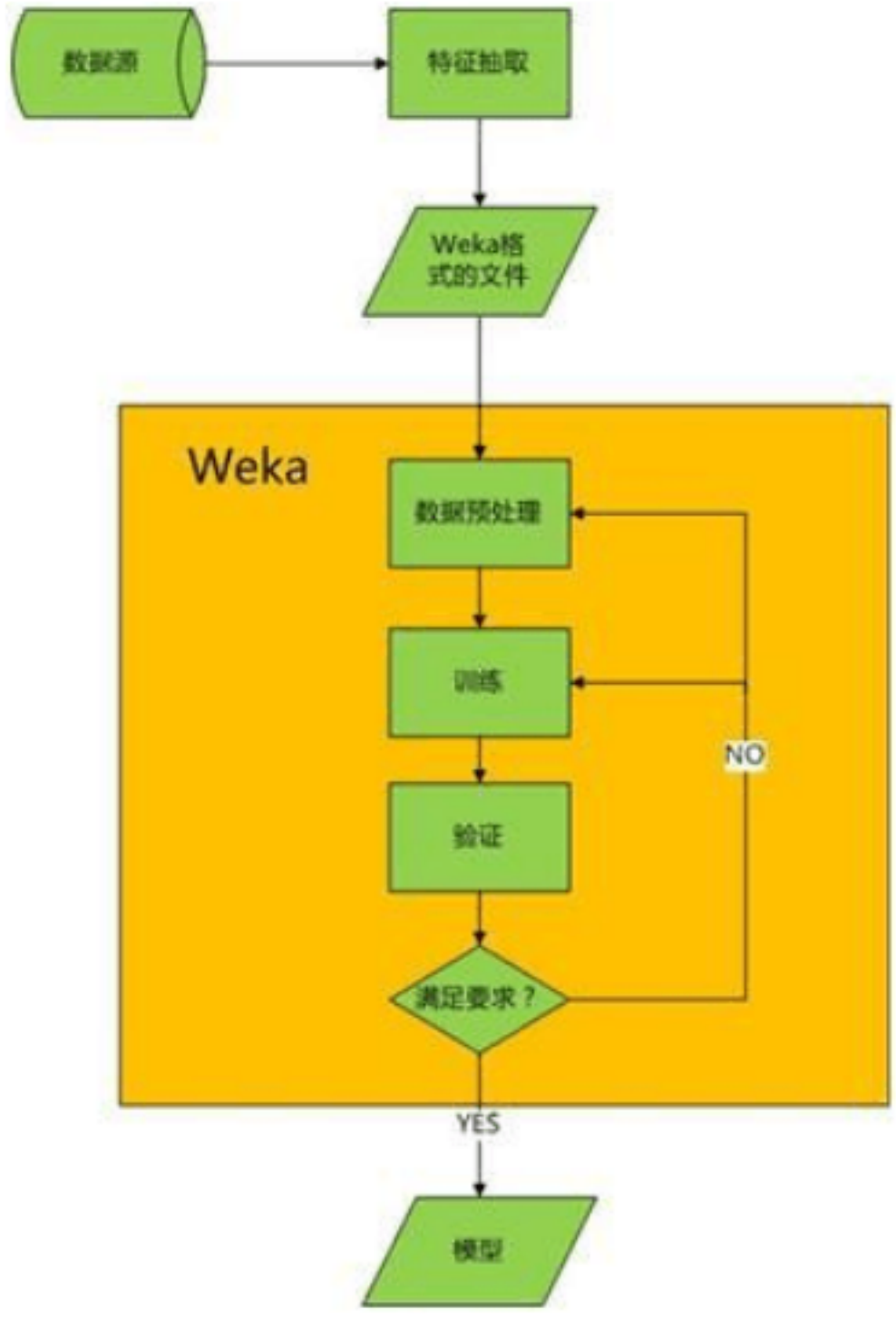

图 4.1 数据挖掘流程图

其中，在 weka 内进行的是数据预处理，训练，验证这三个步骤。

1）数据预处理

数据预处理包括特征选择，特征值处理（比如归一化） ，样本选择等操作。

2）训练

训练包括算法选择，参数调整，模型训练。

3）验证

对模型结果进行验证。

本文剩余部分将以这个流程为主线，以分类为示例，介绍使用 weka 进行数据挖掘的步骤。

5. 数据预处理

打开 Explorer 界面，点“ open file ”，在 weka 安装目录下，选择 data 目录里的“ labor.arff ”文件，将会看到如下界面。我们将整个区域分为 7 部分，下面将分别介绍每部分的功能。

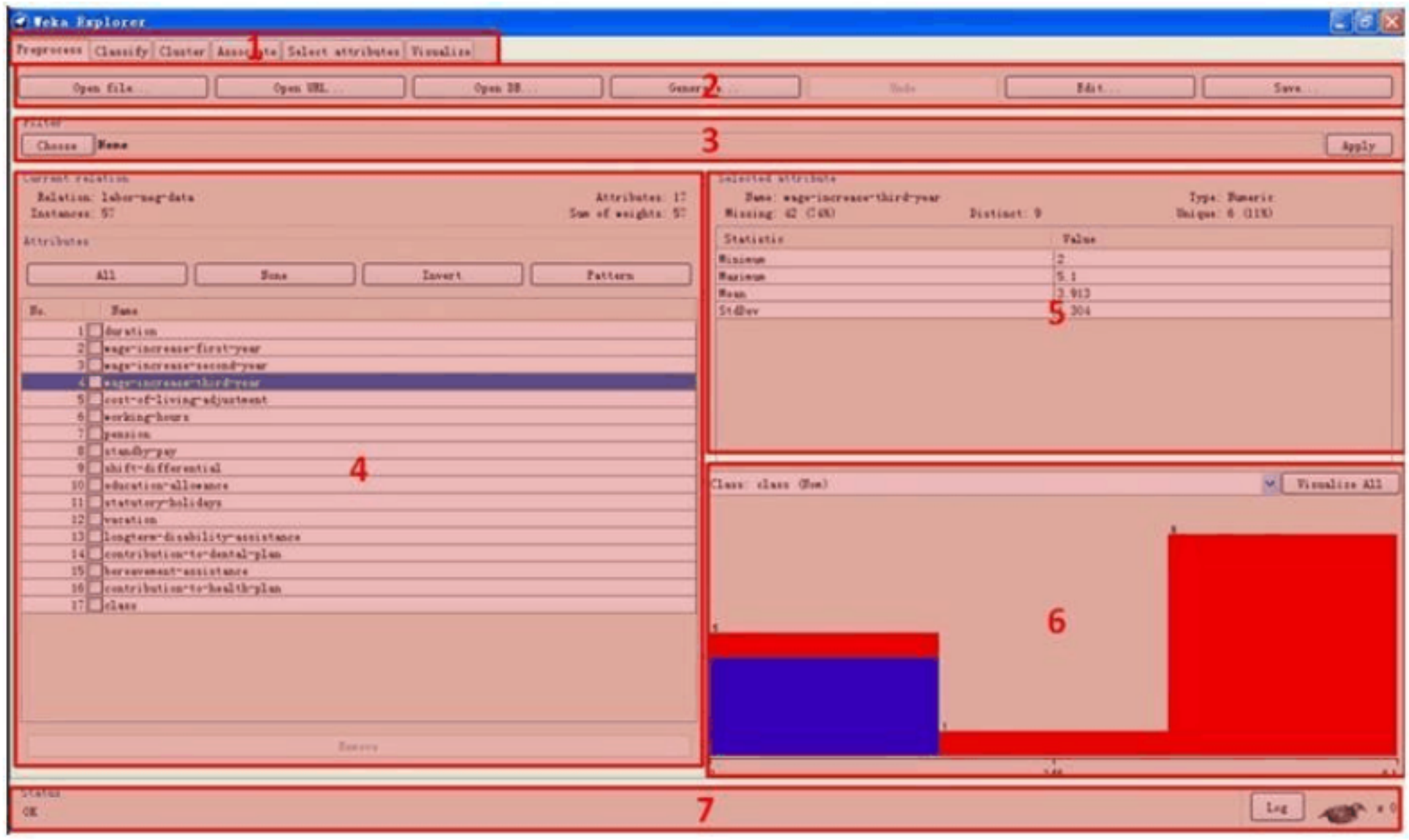

图 5.1 Explorer 界面

1）区域 1 共 6 个选项卡，用来选择不同的数据挖掘功能面板，从左到右依次是 Preprocess（预处理）、Classify（分类）、Cluster（聚类）、Associate（关联规则）、Select attribute（特征选择）和 Visualize（可视化）。

2）区域 2 提供了打开、保存，编辑文件的功能。打开文件不仅仅可以直接从本地选择，还可以使用 url 和 db 来做数据源。 Generate 按钮提供了数据生成的功能， weka 提供了几种生成数据的方法。点开 Edit，将看到如下界面

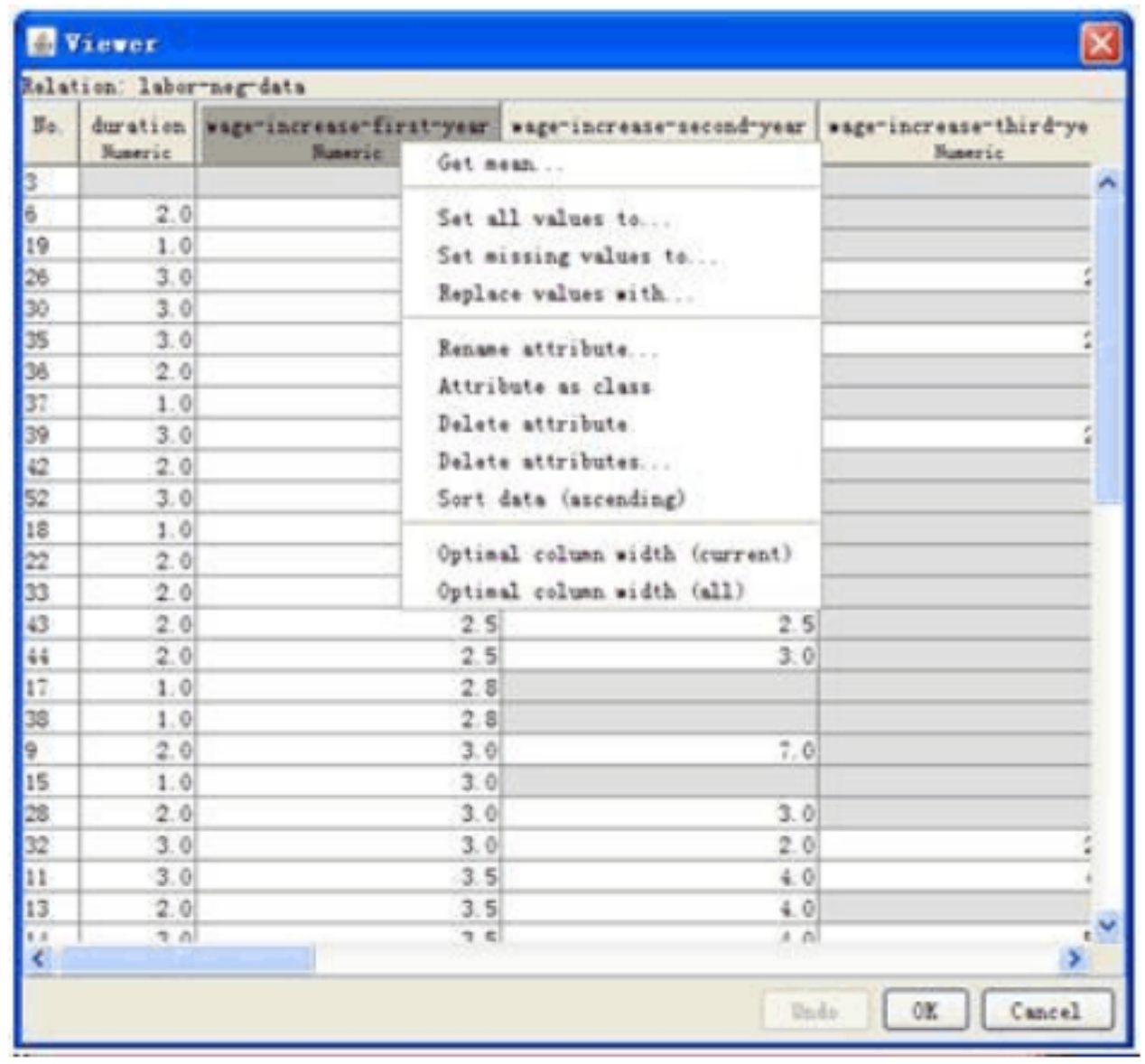

图 5.2 arff viewer

在这个界面，可以看到各行各列对应的值，右键每一列的名字，可以看到一些编辑数据的功能，这些功能还是比较实用的。

3）区域 3 名为 Filter，有些人可能会联想到特征选择里面的 Filter 方法，事实上， Filter 针对特征（ attribute）和样本（ instance）提供了大量的操作方法，功能十分强大。

4）在区域 4，可以看到当前的特征、样本信息，并提供了特征选择和删除的功能。

5）在区域 4 用鼠标选择单个特征后，区域 5 将显示该特征的信息。包括最小值、最大值、期望和标准差。

6）区域 6 提供了可视化功能，选择特征后，该区域将显示特征值在各个区间的分布情况，不同的类别标签以不同的颜色显示。

7）区域 7 是状态栏，没有任务时，小鸟是坐着的，任务运行时，小鸟会站起来左右摇摆。如果小鸟站着但不转动，表示任务出了问题。

下面将通过实例介绍 Filters 的各项功能。

点开 Filter 下面的 choose 按钮，可以看到如下界面

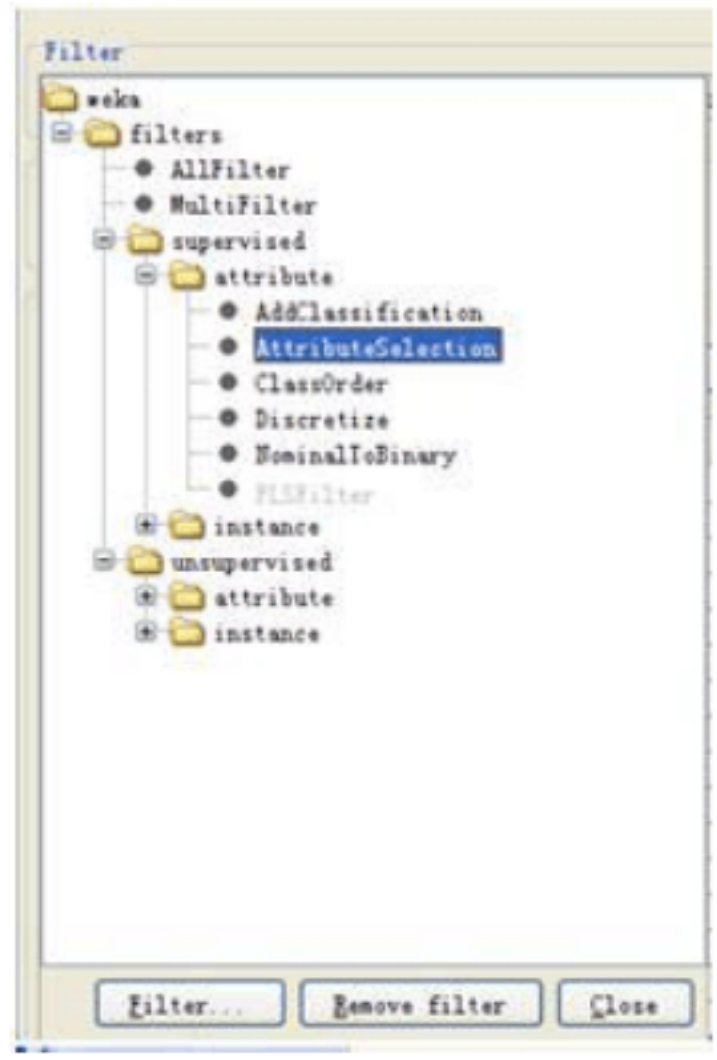

图 5.3 filter 方法选择界面

Filters 可分为两大类， supervised 和 unsupervised。supervised 下的方法需要类别标签， 而 unsupervised 则不需要。attribute 类别表示对特征做筛选， instance 表示对样本做选择。

1） case 1：特征值归一化

该项功能与类别无关， 且是针对 attribute 的, 我们选择 unsupervised -> attribute 下面的 Normalize。点开 Normalize 所在的区域，将看到如下界面。左边的窗口，有几个参数可以选择。点击 more，将出现右边的窗口，该窗口详细介绍了此功能。

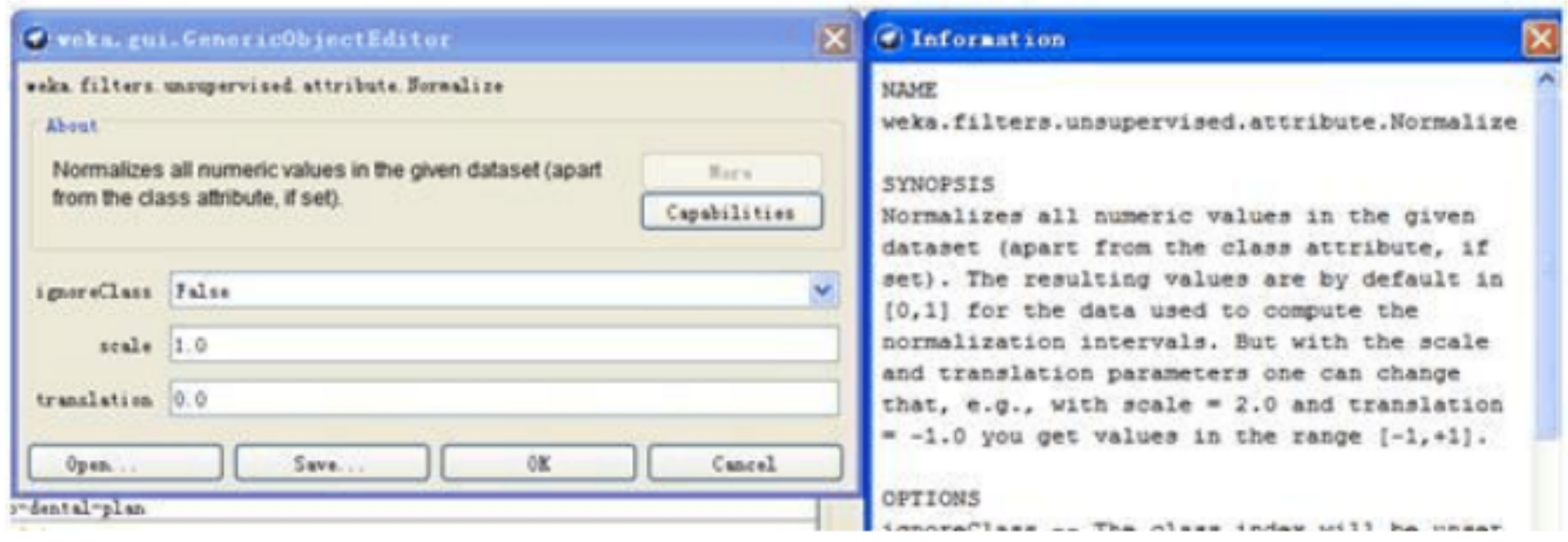

图 5.4 归一化参数设置界面

使用默认参数，点击 ok，回到主窗口。在区域 4 选好将要归一化的特征，可以是一个或多个，然后点击 apply。在可视化区域中，我们可以看到特征值从 1 到 3 被归一到了 0 到 1 之间。

Selected attribute

Name: duration　　Type: Numeric
Missing: 1 (2%)　　Distinct: 3　　Unique: 0 (0%)

Statistic	Value
Minimum	1
Maximum	3
Mean	2.161
StdDev	0.708

图5.5 duration特征归一化之前

Selected attribute

Name: duration　　Type: Numeric
Missing: 1 (2%)　　Distinct: 3　　Unique: 0 (0%)

Statistic	Value
Minimum	0
Maximum	1
Mean	0.58
StdDev	0.354

图5.6 duration特征归一化之后

2） case 2: 分类器特征筛选

该功能与类别相关，　选择 supervised -> attribute 下面的 AttributeSelection 。该界面有两个选项，　evaluator 是评价特征集合有效性的方法，　search 是特征集合搜索的方法。在这里，我们使用　InformationGainAttributeEval　作为 evaluator，使用 Ranker 作为 search，表示我们将根据特征的信息增益值对特征做排序。　Ranker 中可以设置阈值，低于这个阈值的特征将被扔掉。

图 5.7 特征选择参数

点击 apply，可以看到在区域　4 里特征被重新排序，低于阈值的已被删掉。

No.		Name
1	☐	duration
2	☐	wage-increase-first-year
3	☐	wage-increase-second-year
4	☐	wage-increase-third-year
5	☐	cost-of-living-adjustment
6	☐	working-hours
7	☐	pension
8	☐	standby-pay
9	☐	shift-differential
10	☐	education-allowance
11	☐	statutory-holidays
12	☐	vacation
13	☐	longterm-disability-assistance
14	☐	contribution-to-dental-plan
15	☐	bereavement-assistance
16	☐	contribution-to-health-plan
17	☐	class

图5.8 特征选择之前

No.		Name
1	☐	wage-increase-first-year
2	☐	wage-increase-second-year
3	☐	statutory-holidays
4	☐	contribution-to-dental-plan
5	☐	contribution-to-health-plan
6	☐	vacation
7	☐	longterm-disability-assistance
8	☐	shift-differential
9	☐	pension
10	☐	class

图5.9 特征选择之后

3） case 3：选择分类器错分的样本

选择 unsupervised -> instance 下面的 RemoveMisclassified ，可以看到 6 个参数， classIndex 用来设置类别标签，classifier 用来选择分类器，这里我们选择 J48 决策树， invert 我们选择 true，这样保留的是错分样本， numFolds 用来设置交叉验证的参数。设置好参数之后，点击 apply，可以看到样本的数量从 57 减少到了 7。

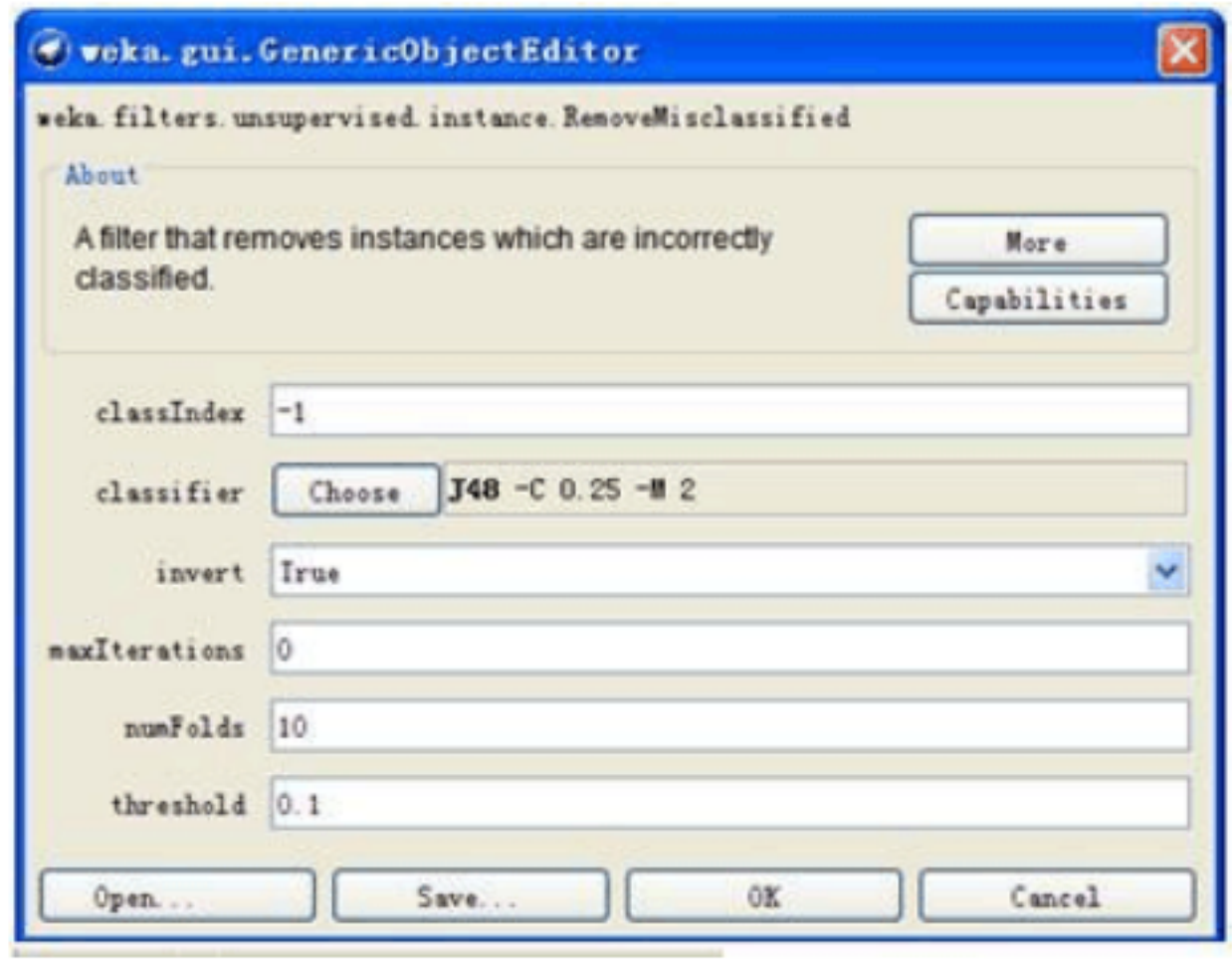

图 5.10 参数设置

6. 分类

在 Explorer 中，打开 classifer 选项卡，整个界面被分成几个区域。分别是

1） Classifier

点击 choose 按钮，可以选择 weka 提供的分类器。常用的分类器有

a） bayes 下的 Naʔve Bayes（朴素贝叶斯）和 BayesNet（贝叶斯信念网络） 。

b） functions 下的 LibLinear 、LibSVM （这两个需要安装扩展包） 、Logistic Regression 、Linear Regression。

c） lazy 下的 IB1 （1-NN ）和 IBK （KNN ）。

d） meta 下的很多 boosting 和 bagging 分类器，比如 AdaBoostM1 。

e） trees 下的 J48（weka 版的 C4.5）、RandomForest。

2） Test options

评价模型效果的方法，有四个选项。

a） Use training set ：使用训练集，即训练集和测试集使用同一份数据，一般不使用这种方法。

b） Supplied test set：设置测试集，可以使用本地文件或者 url，测试文件的格式需要跟训练文件格式一致。

c） Cross-validation ：交叉验证，很常见的验证方法。 N-folds cross-validation 是指，将训练集分为 N 份，使用 N-1 份做训练，使用 1 份做测试，如此循环 N 次，最后整体计算结果。

d） Percentage split：按照一定比例，将训练集分为两份，一份做训练，一份做测试。

在这些验证方法的下面，有一个 More options 选项，可以设置一些模型输出，模型验证的参数。

3） Result list

这个区域保存分类实验的历史，右键点击记录，可以看到很多选项。常用的有保存或加载模型以及可视化的一些选项。

4） Classifier output

分类器的输出结果，默认的输出选项有 Run information ，该项给出了特征、样本及模型验证的一些概要信息；Classifier model ，给出的是模型的一些参数， 不同的分类器给出的信息不同。 最下面是模型验证的结果， 给出了 一些常用的一些验证标准的结果，比如准确率（ Precision），召回率（ Recall），真阳性率（ True positive rate），假阳性率（ False positive rate）， F 值（ F-Measure），Roc 面积（ Roc Area ）等。 Confusion Matrix 给出了测试样本的分类情况，通过它，可以很方便地看出正确分类或错误分类的某一类样本的数量。

Case 1：使用 **J48** 对 **labor** 文件做分类

1）打开 labor.arff 文件，切换到 classify 面板。

2）选择 trees->J48 分类器，使用默认参数。

3） Test options 选择默认的十折交叉验证，点开 More options ，勾选 Output predictions 。

4）点击 start 按钮，启动实验。

5）在右侧的 Classifier output 里面，我们看到了实验的结果。

```
=== Run information ===

Scheme:weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     labor-neg-data
Instances:    57
Attributes:   17
              duration
              wage-increase-first-year
              wage-increase-second-year
              wage-increase-third-year
              cost-of-living-adjustment
              working-hours
              pension
              standby-pay
              shift-differential
              education-allowance
              statutory-holidays
              vacation
              longterm-disability-assistance
              contribution-to-dental-plan
              bereavement-assistance
              contribution-to-health-plan
              class
Test mode:10-fold cross-validation
```

图 6.1 Run information

上图给出了实验用的分类器以及具体参数，实验名称，样本数量，特征数量以及所用特征，测试模式。

```
=== Classifier model (full training set) ===

J48 pruned tree
------------------

wage-increase-first-year <= 2.5: bad (15.27/2.27)
wage-increase-first-year > 2.5
|   statutory-holidays <= 10: bad (10.77/4.77)
|   statutory-holidays > 10: good (30.96/1.0)

Number of Leaves  :     3

Size of the tree :      5


Time taken to build model: 0 seconds
```

图 6.2 模型信息

上图给出了生成的决策树，以及叶子节点数、树的节点数、模型训练时间。如果觉得这样不直观，可以在 Result list 里面右键点击刚刚进行的实验，点击 Visualize Tree，可以看到图形界面的决策树，十分直观。

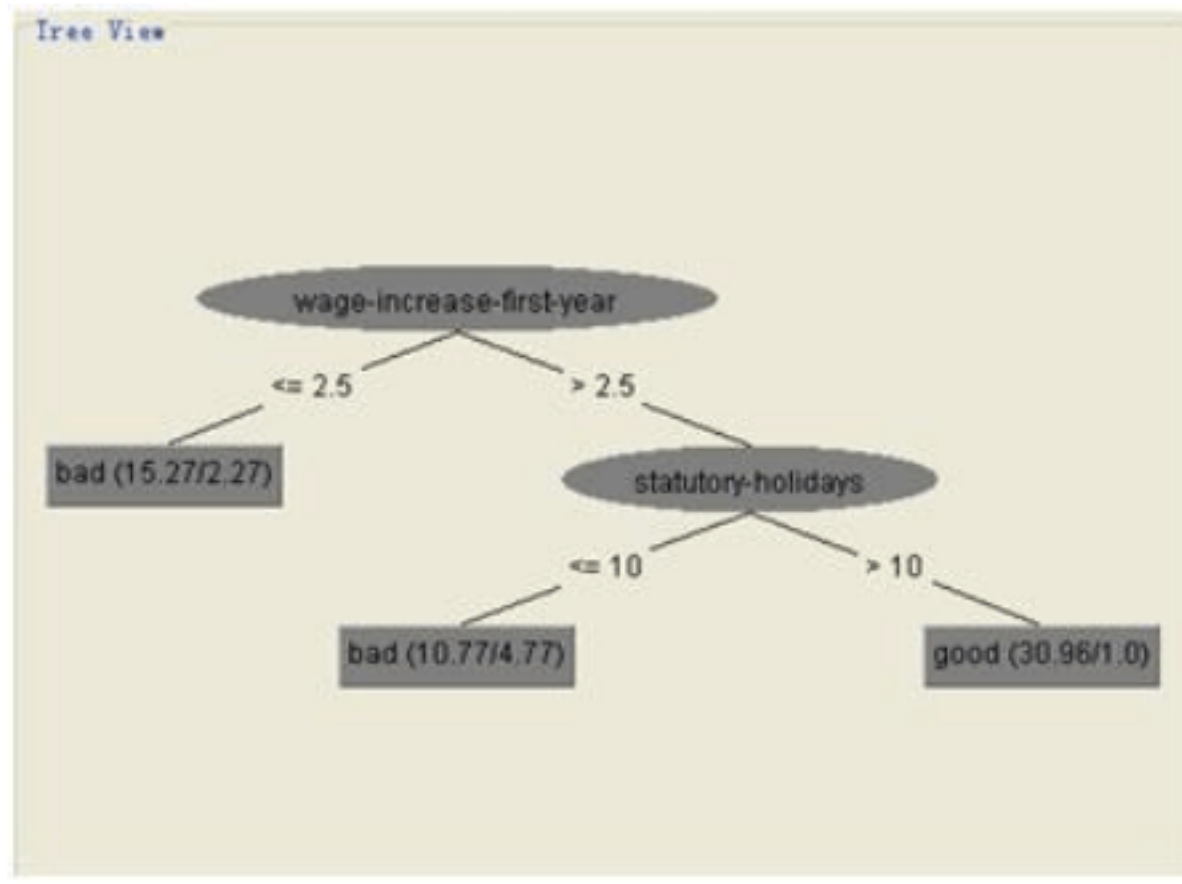

图 6.3 决策树

再往下是预测结果，可以看到每个样本的实际分类，预测分类，是否错分，预测概率这些信息。

```
=== Predictions on test data ===

inst#,     actual, predicted, error, probability distribution
     1       1:bad      2:good      +   0      *1
     2       1:bad       1:bad          *0.762  0.238
     3      2:good      2:good           0.082 *0.918
     4      2:good       1:bad      +  *0.762  0.238
     5      2:good       1:bad      +  *0.762  0.238
     6      2:good      2:good           0      *1
     1       1:bad       1:bad          *0.85   0.15
     2       1:bad      2:good      +   0      *1
     3      2:good      2:good           0      *1
     4      2:good      2:good           0.14  *0.86
     5      2:good       1:bad      +  *0.85   0.15
     6      2:good      2:good           0.14  *0.86
```

图 6.4 预测结果

最下面是验证结果，整体的 accuracy 是 73.68%，bad 类准确率是 60.9%，召回率 70.0%，good 类准确率是 82.4%，召回率 75.7%。

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          42               73.6842 %
Incorrectly Classified Instances        15               26.3158 %
Kappa statistic                          0.4415
Mean absolute error                      0.3192
Root mean squared error                  0.4669
Relative absolute error                 69.7715 %
Root relative squared error             97.7888 %
Total Number of Instances               57

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.7       0.243       0.609     0.7       0.651      0.695    bad
                 0.757     0.3         0.824     0.757     0.789      0.695    good
Weighted Avg.    0.737     0.28        0.748     0.737     0.74       0.695

=== Confusion Matrix ===

  a  b   <-- classified as
 14  6 |  a = bad
  9 28 |  b = good
```

图 6.5 模型效果评估结果

7. 可视化

打开 Explorer 的 Visualize 面板，可以看到最上面是一个二维的图形矩阵，该矩阵的行和列均为所有的特征（包括类别标签），第 i 行第 j 列表示特征 i 和特征 j 在二维平面上的分布情况。图形上的每个点表示一个样本，不同的类别使用不同的颜色标识。

下面有几个选项，PlotSize 可以调整图形的大小，PointSize 可以调整样本点的大小，Jitter 可以调整点之间的距离，有些时候点过于集中，可以通过调整 Jitter 将它们分散开。

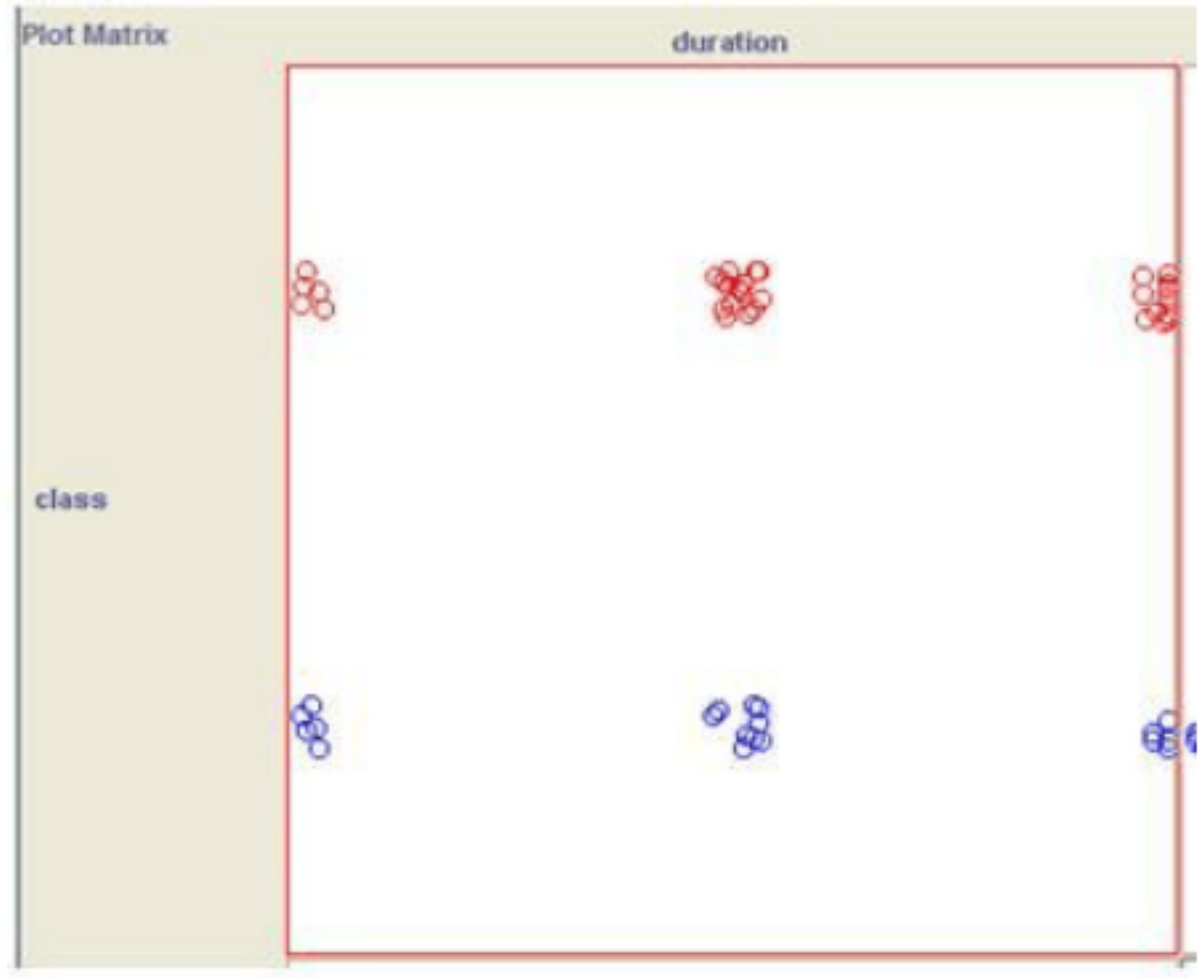

图 7.1 plot matrix 二维图

上图是 duration 和 class 两个特征的图形，可以看出，duration 并不是一个好特征，在各个特征值区间，good 和 bad 的分布差不多。

单击某个区域的图形，会弹出另外一个窗口，这个窗口给出的也是某两个特征之间分布的图形，不同的是，在这里，通过点击样本点，可以弹出样本的详细信息。

可视化还可以用来查看误分的样本，这是非常实用的一个功能。分类结束后，在 Result list 里右键点击分类的记录，选择 Visualize classify errors，会弹出如下窗口。

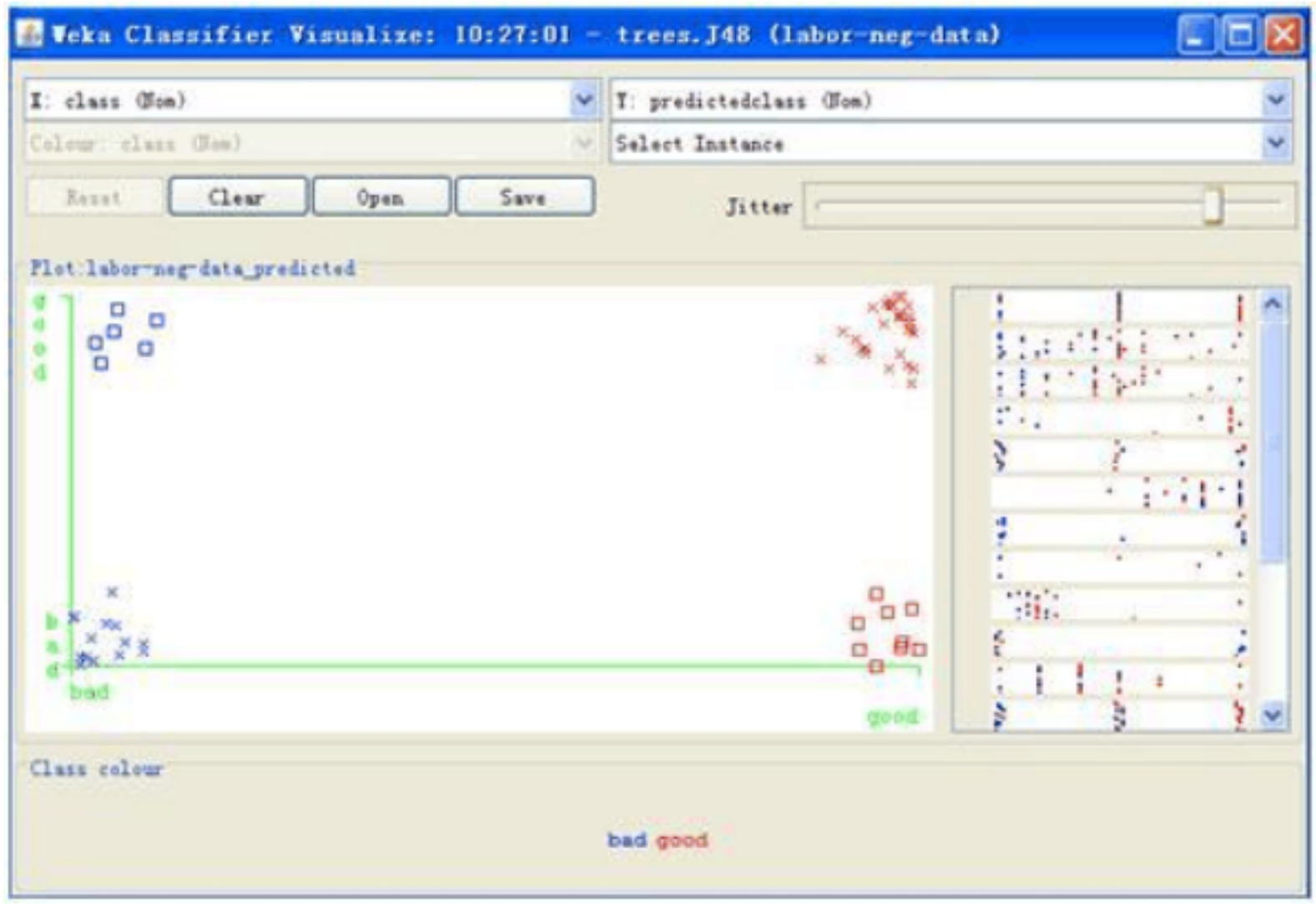

图 7.2 误分样本可视化

这个窗口里面 ,十字表示分类正确的样本, 方块表示分类错误的样本, X 轴为实际类别, Y 轴为预测类别, 蓝色为实际的 bad, 红色为实际的 good。这样, 蓝色方块就表示实际为 bad, 但为误分为 good 的样本, 红色方块表示实际为 good, 被误分为 bad 的样本。单击这些点, 便可以看到该样本的各个特征值, 分析为什么这个样本被误分了。

再介绍一个比较实用的功能, 右键点击 Result list 里的记录, 选择 Visualize threshold curve , 然后选好类别, 可以看到如下图形

图 7.3 阈值曲线

该图给出的是分类置信度在不同阈值下, 分类效果评价标准的对比情况。 上图给出的是假阳性比率和真阳性比率在不同阈值下的对比, 其实给出的就是 ROC 曲线。我们可以通过选择颜色, 方便地观察不同评价标准的分布情况。如果 X 轴和 Y 轴选择的是准确率和召回率, 那我们可以通过这个图, 在这两个值之间做 trade-off , 选择一个合适的阈值。

其它的一些可视化功能, 不再一一介绍。

8. 小结

本文仅仅针对 weka 的 Explorer 界面的某些功能做了介绍, Explorer 其它的功能, 比如聚类、关联规则、特征选择, 以及 Experimentor 和 KnowledgeFlow 界面使用, 可以参考 weka 的官方文档。

另外, weka 支持扩展包, 可以很方便地把 liblinear 、libsvm 这样的开源工具放进来。

在 Linux 下面，可以使用 weka 的命令行进行实验，具体的使用方法，也请参考 weka 官方文档。

有这样一款开源、免费、强大的数据挖掘工具，你还在等什么呢？没有用过 weka 的数据挖掘工程师们，赶紧行动吧。