

课程设计报告

**姓 名：** 周永峰

**学 号：** 19851259

**专 业：**计算机科学与技术

**系 别：**计算机科学与技术

**指导教师：** 王志海

**2022年6月**

# 你自己的标题

## 数据分析网站 (Weka)

### .1.1 199IT大数据导航

http://hao.199it.com/

这个网站是数据的搬运工，里面的数据报告是别的网站产出的，可以直接通过这个网站看报告，也可以通过这个网站去相应的行业网站看报告[8]。

### 1.2 CEIC

https://www.ceicdata.com/zh-hans

涵盖超过195个国家400多万个时间序列的数据源，最完整的一套超过128个国家的经济数据，能够精确查找GDP、CPI、进口、出口、外资等深度数据[8]。

### 1.3 github

https://github.com/

一个非常全面的数据获取渠道，包含各个细分领域的数据库资源，自然科学和社会科学的覆盖都很全面，适合做研究和数据分析的人员[8]。

### 1.4 Weka

Weka的全名是怀卡托智能分析环境（Waikato Environment for Knowledge Analysis），是一款[免费](https://baike.baidu.com/item/%E5%85%8D%E8%B4%B9/131326" \t "_blank)的，非商业化（与之对应的是SPSS公司商业数据挖掘产品--Clementine ）的，基于JAVA环境下开源的[机器学习](https://baike.baidu.com/item/%E6%9C%BA%E5%99%A8%E5%AD%A6%E4%B9%A0/217599" \t "_blank)（machine learning）以及[数据挖掘](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E6%8C%96%E6%8E%98/216477" \t "_blank)（data mining）软件。它和它的[源代码](https://baike.baidu.com/item/%E6%BA%90%E4%BB%A3%E7%A0%81/3969" \t "_blank)可在其官方网站下载。有趣的是，该软件的缩写WEKA也是新西兰独有的一种鸟名（[新西兰秧鸡](https://baike.baidu.com/item/%E6%96%B0%E8%A5%BF%E5%85%B0%E7%A7%A7%E9%B8%A1/5991395" \t "_blank)），而Weka的主要开发者同时恰好来自新西兰的[怀卡托大学](https://baike.baidu.com/item/%E6%80%80%E5%8D%A1%E6%89%98%E5%A4%A7%E5%AD%A6/7082322" \t "_blank)（The University of Waikato）。

Weka的全名是怀卡托智能分析环境（Waikato Environment for Knowledge Analysis），同时Weka也是新西兰的一种鸟名，而Weka的主要开发者来自新西兰。WEKA作为一个公开的数据挖掘工作平台，集合了大量能承担数据挖掘任务的[机器学习算法](https://baike.baidu.com/item/%E6%9C%BA%E5%99%A8%E5%AD%A6%E4%B9%A0%E7%AE%97%E6%B3%95/18635836" \t "_blank)，包括对数据进行预处理，分类，回归、聚类、[关联规则](https://baike.baidu.com/item/%E5%85%B3%E8%81%94%E8%A7%84%E5%88%99)以及在新的交互式界面上的可视化。

如果想自己实现数据挖掘算法的话，可以参考Weka的接口文档。在Weka中集成自己的算法甚至借鉴它的方法自己实现可视化工具并不是件很困难的事情。

2005年8月，在第11届ACM SIGKDD国际会议上，怀卡托大学的Weka小组荣获了数据挖掘和知识探索领域的最高服务奖，Weka系统得到了广泛的认可，被誉为数据挖掘和机器学习 历史上的里程碑，是现今最完备的数据挖掘工具之一（已有11年的发展历史）。Weka的每月下载次数已超过万次。

2014年3月起，新西兰怀卡托大学将推出Weka免费网课，课程分为初级和高级两个部分，每个部分时长5周。初级课程将于2014年3月3日开课，高级课程于2014年4月下旬开课。课程具体内容参见怀卡托大学网站Weka MOOC。课程在优酷网站也有专辑[1]。

Weka系统汇集了前沿的机器学习算法和数据预处理工具，以便用户能够快速灵活地将已有的成熟处理方法应用于新的数据集。

1、处理方法包括处理标准数据挖掘问题的所有方法：回归、分类、聚类、关联规则和属性选择。

2、输入数据通过以ARFF格式为代表的文件进行输入。

（1）探索者（Explorer）  
图形用户界面，通过选择菜单和填写表单，可以调用Weka的所有功能。

（2）知识流（Knowledge Flow）  
使用增量（分批）方式的算法来处理大型数据集，用户可以定制处理数据流的方式和顺序。按照一定顺序将代表数据源、预处理工具、学习算法、评估手段和可视化模块的各组件组合在一起，形成数据流。

（3）实验者（Experimenter）  
帮助用户解答实际应用分类和回归技术中遇到的一个基本问题。

（4）简单命令行（Simple CLI）  
该界面用于和用户进行交互，可以直接执行Weka命令[2]。

训练classifier：

首先在Preprocess标签下，点击Open file选择一个你要打开的dataset，这个dataset需要符合一定的文件格式才能被Weka解析，最常用的是arff格式的文件，所以最好先将你的dataset转换成这个格式的文件。打开文件后，Weka会自动解析数据，并且显示关于这个dataset的统计信息在图形界面上。比如这个dataset含有多少instances，每个instances有多少attributes，attributes的分布情况，而且提供可视化图像使得数据更加直观。

如果要做一个简单的classifier的训练，首先我们打开一个Weka自带的dataset ：breast-cancer.arff，这是一个关于患者乳腺癌是否复发的dataset，数据比较简单，每一个instance代表一个病人，有9个attributes，包括age，tumor-size等等，所有attributes都是nominal属性的，每一个instance带有一个class属性，no-recur或者recur。

现在我们选择一个classifier来进行训练。切换到Classify标签下，点击Choose选择一个classifier，在Weka-classifier-trees下选择Decision Stump，此时点击Choose旁边的文本栏可以设置Decision Stump的具体参数，调整完毕之后选择如何进行测试，这里我们选择使用cv进行测试，folds选择10，点击Start，好了，Weka就会开始使用我们的dataset并且按照我们调整的参数来和训练方法来训练classifier了。训练完毕后classifier的结构以及测试结果都会输出在output栏，你可以很直观地看到当前模型的训练成果。拿我们这个例子来说，因为Decision Stump模型是在太过简单，仅仅根据deg-malig属性是否等于3来进行分类，正确率仅有68%左右。让我们换一个classifier再试试，这次我们选择J48，这是一个C4.5的实现。如果只是选择默认参数并且不prune的话，我们会发现训练出来的tree结构非常复杂，并且分类效果依然很差，仅有69%左右，很明显，训练出的模型出现了overfitting。此时我们设置一下J48的属性，将unpruned的参数调增为False，即允许prune，再次进行训练，可以看到，这次训练出来的tree结构得到了极大的简化，并且正确率也提升到了75%。当然，要想得到好的结果还需要进行参数调整，我们这里只是使用了默认设置。再换一个knnclassifier试试，选择Weka-classifiers-lazy下面的IBK，并且设置它的KNN参数为5，即使用离输入点最近的5个样本点作为分类参考，训练完成后分类准确率在73%左右。

Unbalance数据预处理：

有时候我们的dataset会有严重的unbalance问题，会导致classifier建模过程受到极大影响。比如我曾经使用过一个dataset，数据总量为9688，但是其中8704个instances为class 0，而剩下的984个instances为class 1。使用J48训练过后，class0的分类效果较为理想，准确率可达96%，但是class1因为训练样本量过小，准确率仅有32%。在选择dataset时，我们应尽量保证其中的数据在各个class之中是平衡的，并且如实反映真实世界中的数据分布。如果手头数据偏差过大，那么我们就需要在将数据送入classifier进行训练之前对数据进行一些预处理。这里我们讨论两个比较典型的做法。

第一个是under sampling，即从class占比例较大instances中sample出一部分数据，从而和class占比例较小的instances组成更为均衡的dataset。在Preprocess标签下，在Filter下点击Choose，在Weka-supervised-instance下选择Spread Subsample，然后设置参数，其中distribution Spread选择1.0，代表uniform distribution，使得从major class 中sample出的instances数量和minor class保持一致，然后点击Apply，这时候我们得到的dataset 其中class 0和class 1的instances都是984个，这样就可以进行后续的数据处理了。

第二个是oversampling，我们讨论一个比较常用的算法：SMOTE，这个算法的特点是可以根据原有数据的特点生成原本不存在dataset里的instances，从而扩充minor的dataset。同样在Weka-supervised-instance下选择SMOTE，然后设置参数，class Value采用默认值，这样SMOTE会自动识别哪一个class的数量是少数，并进行扩充，在percentege中选择你想要将minor class的instances扩充到多少比例，比如我的数据中class0的数据有8704，而class1的数据有984，那么可以设置percentage为800，那么扩充后的class1就可以和class0保持平衡。结果显示class0的数据数量不变，而class1的数据量变成了8856.

数据降维：

使用Weka还可以很方便的进行数据降维。当一个dataset中attributes太多的时候，会出现curse of dimensionality，这是因为数据维度空间过于复杂而造成数据难以处理。此时进行适当的降维就在所难免。我们这里也讨论两个Weka中比较常用的降维方法。

第一个是使用Weka的Info Gain Attribute Eval进行attributes选择。基本原理是通过计算每一个attribute对于降低entropy的贡献来进行排序，然后选择排序较高的留下来，去掉排序较低的attributes，从而达到降维的目的。切换到Select attributes标签下，选择Weka-attribute Selection下的Info Gain Attribute Eval，然后再选择一个Search Method，比如我们选择Ranker，然后设置Ranker的threshold为0.2，意思是去掉所有Info Gain低于0.2的attributes，那么剩下的attributes就是对分类作用比较大的。以我手头的一个dataset为例，降维后attributes数量从524降到了128。我们可以保存这个dataset到新的arff文件中。

第二个方法是PCA。基本思想是不删除任何一个attribute，但是将原有的attributes进行线性组合，从而生成一组新的attributes，但是数量远小于原有attributes，从而达到降维的目的。同样在Weka-attribute Selection下选择Principal components，即可进行PCA降维，降维后的dataset同样可以保存并进行后续处理[3]。

数据挖掘终于可以和烦人的代码们说再见了!Weka，一个不足两岁的新生儿，让数据挖掘轻松易行，无需编程也能轻松搞定。Weka是基于java，用于数据挖掘和知识分析一个平台。来自世界各地的java爱好者们都可以把自己的算法放在这个平台上，然后从海量数据中发掘其背后隐藏的种种关系；也许你只是出于对数据的狂热爱好，但也许你的发现会蕴含着无限的商机。打开Weka，首先出现一个命令行窗口。原以为要在这个命令行下写java语句呢，不过稍等一秒，Weka GUIChooser的出现了。这是一个很简单的窗体，提供四个按钮：SimpleCLI、Explorer、Experimenter、 KnowledgeFlow。SimpleCLI应该是一个使用命令行的界面，有点像SAS的编辑器；Explorer是则是视窗模式下的数据挖掘工 具；Experimenter和KnowledgeFlow的使用有待进一步摸索....    先打开WekaExlporer感受一下它的强大吧。它有六个标签页，分别是Preprocess、Classify、Cluster、 Associate、Selectattributes、Visualize。在Preprocess中Open一个数据文件(Weka使用的数据文件是arff，其实是一个文本数据集，格式并不复杂，用notepad打开一看就明白了)。当然也可以Open URL或Open DB，不过我没有check一下支持哪些DB。   打开数据文件后，可以使用Filter进行一下过滤，相当于“预处理的预处理”。Filter提供了许多算法来过滤数据，比如filters/unsupervised/instance/normalize应该是一个标准化的算法。当然，也可以编写你自己的算法!    这时窗体上已经给出这个数据集的一些基本特征了，比如有多少属性，各属性的一些简单统计量，右下方还给出一些可视化效果比如柱状图。通过这些可以初步了解这个数据集了。但这些都是很直观的可以看出来，好戏在后头，隐藏的关系即将登场。    接 下来的两个标签页是classify(分类)和cluster(聚类)，接触数据挖掘的人对它们一定不会陌生。同样Weka有许多分类和聚类算法可供选 择，在这里面称为clasifier和clusterer。不过Weka提供的classify功能似乎还不够灵活，只能定长度和定频率地分类。但这个关 系不大，现在很多数据处理软件都可以做到这个，比如excel。Cluster功能强大，提供了许多巧妙的聚类算法，选定一个算法，给出你所需要生成的聚 类数目，就可以自动完成。当然如果能不给出聚类数目也能自动聚类的话就更佳了，不过我还没发现怎么做。Next，终于到伟大的Associate了! 这是一个用于发掘AssociateRules(关联规则)的模块。对商学略有涉猎的人一定熟知沃尔玛发现了啤酒和尿布销售的关系这一佳话。有了WekaAssociate，任何一家超市都可以做到这一点了。将前面导入的数据使用Associator进行发掘，就可以发现其中无数隐藏的关系。Weka-3-4提供了Apriori、PredictiveApriori、Tertius三种关联规则发掘算法，不过我感觉这已经够用了。选定一个算 法，进行一些必要的设置，包括支持度上界、下界，每次运算的支持度递减值，等等。另外一个重要的参数：所需要生成的关联规则个数。太不可思议了，以前我们 能从海量数据中发现一个关联规则就已经沾沾自喜，现在Weka居然问你想生成多少关联规则!     参数设置完成，点Start，就可以去喝茶了。不一会，10条关联规则已经生成，可以提交给老板了。当然，你还可以分析一下哪些规则比较有用，哪一条有潜在收益，这就需要business sense了。另外两个标签页还没怎么看。Selectattributes大概是针对单属性的分析？Visualize则提供了许多可视化效果，需要拿出去演示时很方便。不过今天使用感觉这个模块的功能有点问题，没太搞懂。也有可能是我用错了。    Weka实在是一个伟大的工具。基于java，却没有运行其它java程序那种慢吞吞的感觉。前天我还在说Data Mining isexcruciating but interesting，有了Weka，Data Mining也可以轻轻松松了[4]!

## 2 Weka软件系统功能

Weka系统汇集了前沿的机器学习算法和数据预处理工具，以便用户能够快速灵活地将已有的成熟处理方法应用于新的数据集。

### 2．1、处理方法

包括处理标准数据挖掘问题的所有方法：回归、分类、聚类、关联规则和属性选择。

### 2.2、输入数据

通过以ARFF格式为代表的文件进行输入

直接读取数据库表

### 2.3、Weka主界面 Weka GUI Chooser

#### 2.3.1探索者（Explorer）

图形用户界面，通过选择菜单和填写表单，可以调用Weka的所有功能。

使用 WEKA 探索数据的环境。在这个环境中，Weka提供了数据的预处理，数据格式的转化（从CSV格式到ARFF格式的转化，详见第4部分），各种数据挖掘算法（包括分类与回归算法，聚类算法，关联规则等），并提供了结果的可视化工具。  
对于一个数据集，通过简单的数据的预处理，并对数据挖掘算法进行选择（在Weka3.5版本之后，加入了算法的过滤功能，可以过滤掉那些不适合当前数据集类型的算法），接着通过窗口界面对算法的参数进行配置，最后点击“Start”按钮就可以运行了。  
可视化工具分为对数据集的可视化和对部分结果的可视化（详情可以参考 4.2 Weka的输出格式），并且我们可以通过属性选择工具(Select Attribute)，通过搜索数据集中所有属性的可能组合，找出预测效果最好的那一组属性。

#### 2.3.2知识流（Knowledge Flow）

使用增量（分批）方式的算法来处理大型数据集，用户可以定制处理数据流的方式和顺序。按照一定顺序将代表数据源、预处理工具、学习算法、评估手段和可视化模块的各组件组合在一起，形成数据流。

这个环境本质上和Explorer所支持的功能是一样的，但是它有一个可以拖放的界面。它有一个优势，就是支持增量学习（incremental learning）。  
Knowledge Flow 为Weka 提供了一个＂数据流＂形式的界面．用户可以从一个工具栏中选择组件，把它们放置在面板上并按一定的顺序连接起来，这样组成一个＂知识流＂（knowledge flow）来处理和分析数据．目前，所有的Weka 分类器（classifier）、筛选器（filter）、聚类器（clusterer）、载入器（loader）、保存器（saver），以及一些其他的功能可以在Knowledge Flow中使用。  
Knowledge Flow 可以使用增量模式（incrementally）或者批量模式（inbatches）来处理数据（Explorer只能使用批量模式）．当然对数据进行增量学习要求分类器能够根据各实例逐个逐个的更新．现在Weka 中有五个分类器能够增量地处理数据：Naïve Bayes Updateable，IB1，IBk，LWR（局部加权回归）．还有一个meta 分类器Raced Incremental Logit Boost 可以使用任意基于回归的学习器来增量地学习离散的分类任务。

#### 2.3.3实验者（Experimenter）

帮助用户解答实际应用分类和回归技术中遇到的一个基本问题。

运行算法试验、管理算法方案之间的统计检验的环境。Experiment环境可以让用户创建，运行，修改和分析算法试验，这也许比单独的分析各个算法更加方便。例如，用户可创建一次试验，在一系列数据集上运行多个算法（schemes），然后分析结果以判断是否某个算法比其他算法（在统计意义下）更好。  
Explorermenter主要包括简单模式，复杂模式和远程模式。复杂模式是对简单模式的基本功能的扩充，而远程模式允许我们通过分布式的方法进行实验。  
就功能模块而言，分为设置模块，运行模块和分析模块。在设置模块中我们可以自定义实验，加入多个算法和多方的源数据（支持ARFF文件，CSV文件和数据库），在运行模块中我们可以运行我们的实验，而在分析模块中，我们可以分析各种算法的的准确性，并提供了各种统计方法对结果进行检验比较。  
值得一提的是，我们可以把实验的各种参数，包括算法，数据集等，保存以方便下一次相同实验的进行；也可以把各种算法保存，方便应用在不同的数据集上；如果数据集来源于数据库的话，实验在过程中可以中止并继续（原因可以是被中止或者是扩展了实验），这样就不用重新运行那些已实验过的算法/数据集祝贺，而仅计算还没有被实验的那些。

#### 2.3.4简单命令行（Simple CLI）

该界面用于和用户进行交互，可以直接执行Weka命令[5]。

Weka包的主要内容和特点3.1 Weka的核心功能Weka主要包括4个核心部分，也是4大功能环境，分别是：提供了一个简单的[命令行](https://so.csdn.net/so/search?q=%E5%91%BD%E4%BB%A4%E8%A1%8C&spm=1001.2101.3001.7020)界面，从而可以在没有自带命令行的操作系统中直接执行 WEKA 命令。对于Weka的命令详细，我们可以参考Wiki[1]。  
使用命令行有两个好处：一个是可以把模型保存下来，这样有新的待预测数据出现时，不用每次重新建模，直接应用保存好的模型即可。另一个是对预测结果给出了置信度，我们可以有选择的采纳预测结果，例如，只考虑那些置信度在85%以上的结果。  
Knowledge Flow相对于Explorer的几个特性：（a）直观的数据流式布局  
（b）批量或增量地处理数据  
（c）并行处理多批数据或流程（每个流在各自的线程中执行）  
（d）筛选器可链接在一起  
（e）可以查看分类器在交叉验证的各折所产生的模型  
（f）可以对处理中的增量分类器的效果进行可视化（为分类准确度、RMS误差、预测等作图）[6]。

## Weka数据集合格式

### 3．1、处理方法

Weka存储数据的格式是ARFF(Attribute-Relation File Format)文件,是一种ASCII文本文件。如下例，Weka自带的weather.arff文件。

% ARFF file for the weather data with some numric features   
%   
@relation weather   
@attribute outlook {sunny, overcast, rainy}   
@attribute temperature real  
@attribute humidity real  
@attributewindy{TRUE, FALSE}  
@attribute play{yes, no}  
@data  
%  
% 4 instances  
%  
sunny,85,85,FALSE,no  
sunny,80,90,TRUE,no  
overcast,83,86,FALSE,yes  
rainy,70,96,FALSE,yes

Weka数据以“%”开始的行是注释，WEKA将忽略这些行。  
除去注释后，整个ARFF文件可以分为两个部分。

第一部分给出了头信息（Head information），包括了对关系的声明和对属性的声明。

第二部分给出了数据信息（Data information），即数据集中给出的数据。从“@data”标记开始，后面的就是数据信息了。

关系声明  
关系名称在ARFF文件的第一个有效行来定义，格式为  
@relation <relation-name>  
<relation-name>是一个字符串。如果这个字符串包含空格，它必须加上引号（指英文标点的单引号或双引号）。  
属性声明  
属性声明用一列以“@attribute”开头的语句表示。数据集中的每一个属性都有它对应的“@attribute”语句，来定义它的属性名称和数据类型。  
声明语句的顺序很重要:它表明了该项属性在数据部分的位置;最后一个声明的属性被称作class属性，在分类或回归任务中，它是默认的目标变量。  
属性声明的格式为  
@attribute <attribute-name> <datatype>  
其中<attribute-name>是必须以字母开头的字符串。和关系名称一样，如果这个字符串包含空格，它必须加上引号。  
Weka支持的<datatype>有四种，分别是  
numeric数值  
<nominal-specification>分类（nominal）型  
string字符串型  
date [<date-format>]日期和时间型  
注意“integer”，“real”，“numeric”，“date”，“string”这些关键字是区分大小写的，而“relation”“attribute”和“date”则不区分。

数值属性   
数值型属性可以是整数或者实数，但Weka把它们都当作实数看待。   
分类属性   
分类属性由<nominal-specification>列出一系列可能的类别名称并放在花括号中：{<nominal-name1>, <nominal-name2>, <nominal-name3>, ...} 。数据集中该属性的值只能是其中一种类别。   
例如如下的属性声明说明“outlook”属性有三种类别：“sunny”，“ overcast”和“rainy”。而数据集中每个实例对应的“outlook”值必是这三者之一。  
@attribute outlook {sunny, overcast, rainy}  
如果类别名称带有空格，仍需要将之放入引号中。  
字符串属性  
字符串属性中可以包含任意的文本。这种类型的属性在文本挖掘中非常有用。   
示例：  
@ATTRIBUTE LCC string  
日期和时间属性  
日期和时间属性统一用“date”类型表示，它的格式是   
@attribute <name> date [<date-format>]   
其中<name>是这个属性的名称，<date-format>是一个字符串，来规定该怎样解析和显示日期或时间的格式，默认的字符串是ISO-8601所给的日期时间组合格式“yyyy-MM-ddTHH:mm:ss”。   
数据信息部分表达日期的字符串必须符合声明中规定的格式要求。

数据信息   
数据信息中“@data”标记独占一行，剩下的是各个实例的数据。   
每个实例占一行。实例的各属性值用逗号“,”隔开。如果某个属性的值是缺失值（missing value），用问号“?”表示，且这个问号不能省略。例如：   
@data   
sunny,85,85,FALSE,no   
?,78,90,?,yes   
字符串属性和分类属性的值是区分大小写的。若值中含有空格，必须被引号括起来。例如：   
@relation LCCvsLCSH   
    @attribute LCC string   
    @attribute LCSH string   
    @data   
    AG5, 'Encyclopedias and dictionaries.;Twentieth century.'   
    AS262, 'Science -- Soviet Union -- History.'   
日期属性的值必须与属性声明中给定的相一致。例如：   
@RELATION Timestamps   
    @ATTRIBUTE timestamp DATE "yyyy-MM-dd HH:mm:ss"   
    @DATA   
    "2001-04-03 12:12:12"   
    "2001-05-03 12:59:55" 稀疏数据

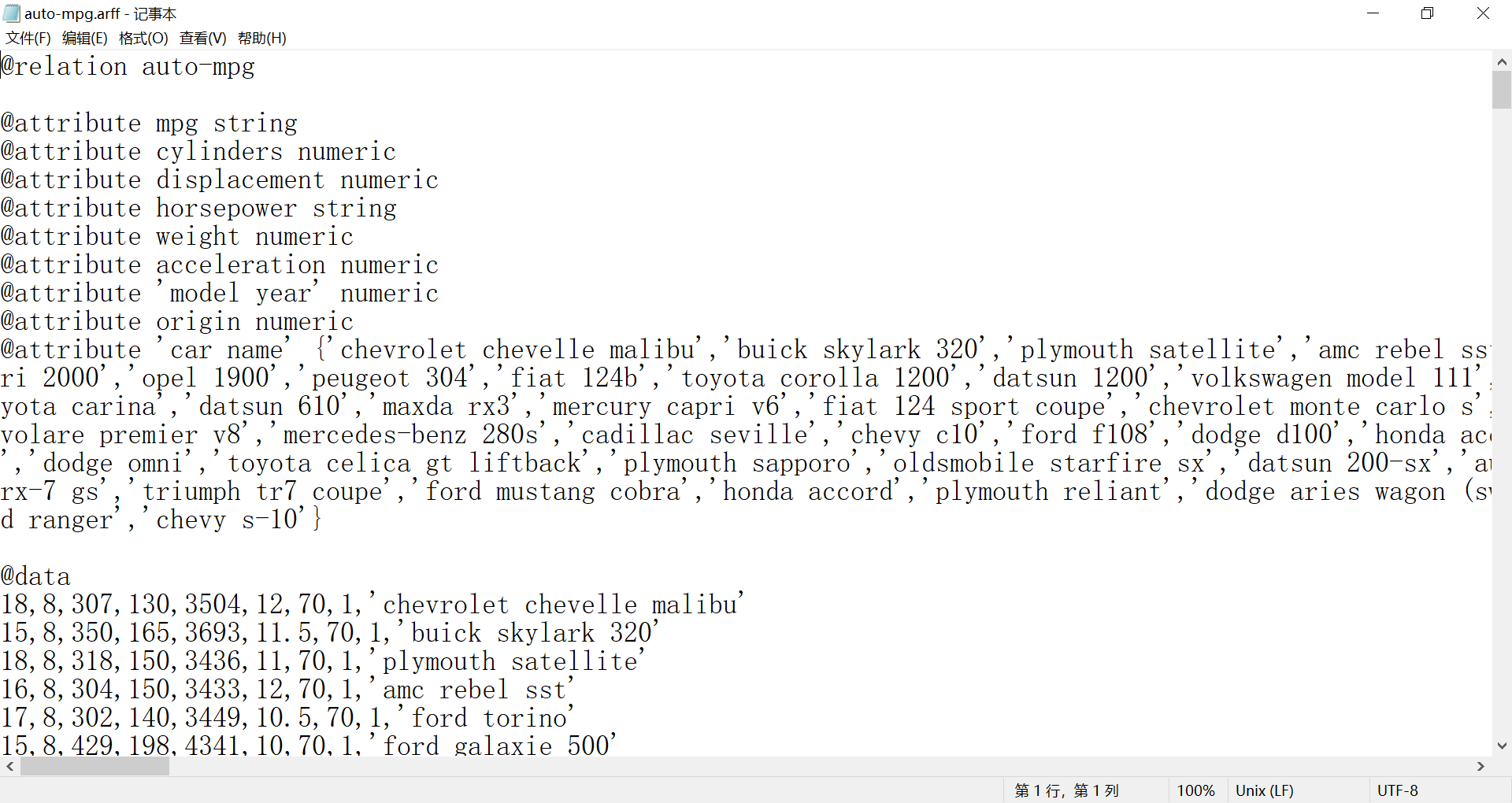
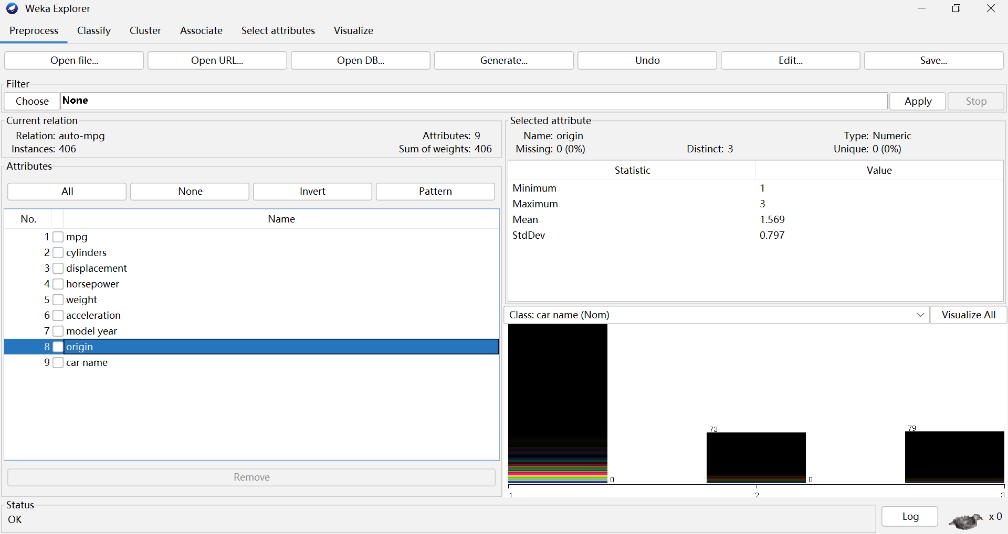
有的时候数据集中含有大量的0值（比如购物篮分析），这个时候用稀疏格式的数据存贮更加省空间。

表1-1 Weather数据集合

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | outlook | temperature | humidity | windy | play |
| 1 | sunny | hot | high | FALSE | no |
| 2 | sunny | hot | high | TRUE | no |
| 3 | overcast | hot | high | FALSE | yes |
| 4 | rainy | mild | high | FALSE | yes |
| 5 | rainy | cool | normal | FALSE | yes |
| 6 | rainy | cool | normal | TRUE | no |
| 7 | overcast | cool | normal | TRUE | yes |
| 8 | sunny | mild | high | FALSE | no |
| 9 | sunny | cool | normal | FALSE | yes |
| 10 | rainy | mild | normal | FALSE | yes |
| 11 | sunny | mild | normal | TRUE | yes |
| 12 | overcast | mild | high | TRUE | yes |
| 13 | overcast | hot | normal | FALSE | yes |
| 14 | rainy | mild | high | TRUE | no |

稀疏格式是针对数据信息中某个实例的表示而言，不需要修改ARFF文件的其它部分。看如下的数据：  
@data  
    0, X, 0, Y, "class A"  
    0, 0, W, 0, "class B"  
用稀疏格式表达的话就是  
@dat  
    {1 X, 3 Y, 4 "class A"}  
    {2 W, 4 "class B"}  
每个实例用花括号括起来。实例中每一个非0的属性值用<index> <空格> <value>表示。<index>是属性的序号，从0开始计；<value>是属性值。属性值之间仍用逗号隔开。注意在稀疏格式中没有注明的属性值不是缺失值，而是0值。若要表示缺失值必须显式的用问号表示出来[7]。

### 3.2 Auto MPG



**考文献**

[1] 百度百科, URL: https://baike.baidu.com/item/weka/10701215?fr=aladdin. [2022-6-16].

[2] CSDN, URL: https://www.csdn.net/tags/OtTacgysMzUyODQtYmxvZwO0O0OO0O0O.html. [2022-6-16].

[3] CSDN, URL: https://blog.csdn.net/ciscoipcbu/article/details/44668235. [2022-6-16].

[4] CSDN, URL: https://www.6miu.com/read-4460287.html. [2022-6-16].

[5] CSDN, URL: https://blog.csdn.net/sinat\_25873421/article/details/82724903. [2022-6-17].

[6] CSDN, URL: https://blog.csdn.net/shaoz/article/details/6841498?spm=1001.2101.3001.6650.1&utm\_medium=distribute.pc\_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-1-6841498-blog-82724903.pc\_relevant\_antiscanv3&depth\_1-utm\_source=distribute.pc\_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-1-6841498-blog-82724903.pc\_relevant\_antiscanv3&utm\_relevant\_index=2. [2022-6-17].

[7] CSDN, URL: https://blog.csdn.net/buaalei/article/details/7103055. [2022-6-17].

[8] 知乎, URL: https://zhuanlan.zhihu.com/p/453897774 [2022-06-17].

。

参考文献以文献在整个论文中出现的次序用［1］、［2］、［3］……形式统一排序、依次列出。

参考文献的表示格式为：

期刊文献：［序号］主要责任者．文献题名［J］．刊名，年，卷（期）：引用部分起止页码．

专著、论文集、学位论文、报告： ［序号］ 主要责任者．文献题名［文献类型标识］．出版地：出版者，出版年. 引用部分起止页码．

论文集中的析出文献：［序号］主要责任者. 析出文献题名［A］. 原文献主要责任者(任选)．原文献题名［C］. 出版地：出版者，出版年. 析出文献起止页码.

专利：［序号］ 专利所有者. 专利题名［P］. 专利国别：专利号, 出版日期．

报纸文章：［序号］ 主要责任者．文献题名［N］． 报纸名，出版日期（版次）．

电子文献：［序号］ 主要责任者. 电子文献题名［电子文献及载体类型标识］.电子文献的出处或可获得地址，发表或更新日期/引用日期（任选）．

国际、国家标准：［序号］ 标准编号，标准名称［S］．