# 报告

**学号: 3019213043　　姓名:刘京宗　　班级:软工 5 班**

## 1. 目标

练习如何构建决策树。认识归一化和离散化对构建决策树的影响。

## 2. Data

1) Bank-all.arff 是银行的所有数据。当我们不拆分数据的时候，我们可以用 10-crossvalidation 来测试分类器的准确性。数据的最后一个属性是类标签。

2) Bank-tain.arff is used for constructing the model.

Bank-test.arff is used for testing the model.

The last attribute is the class label.

3）**weather-nominal.arff**

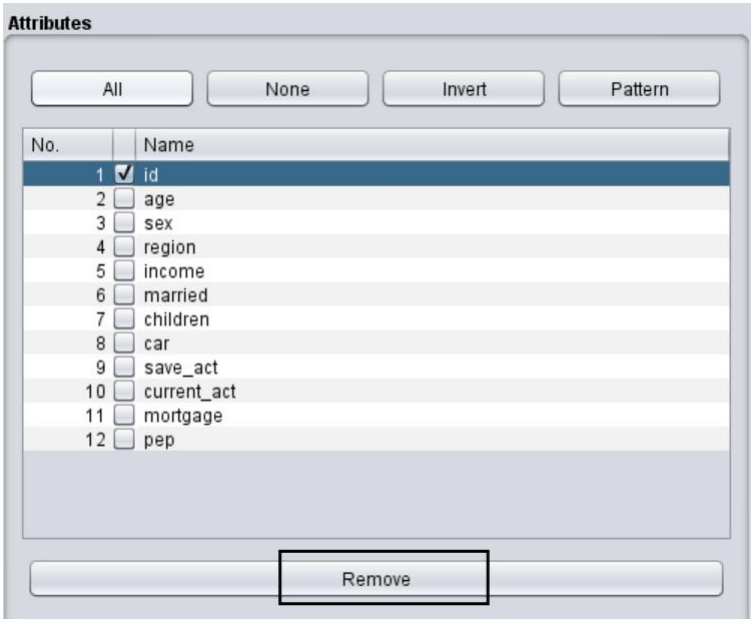
## 3. Contents

两个实验：

### 1. Bank-all.arff

1) 预处理，删除无用属性，保存到新的数据文件。

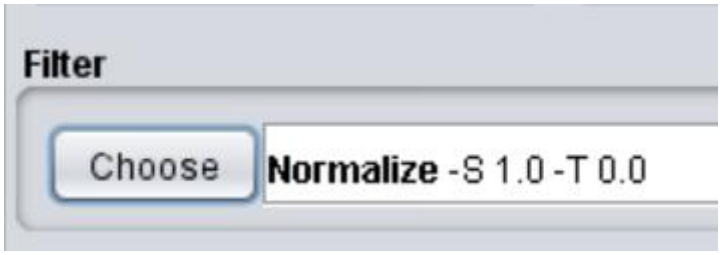
id 这一属性对于我们分类是无用属性，因此删除这一无用数据，保存到新的文件 Bank-all-1.arff。



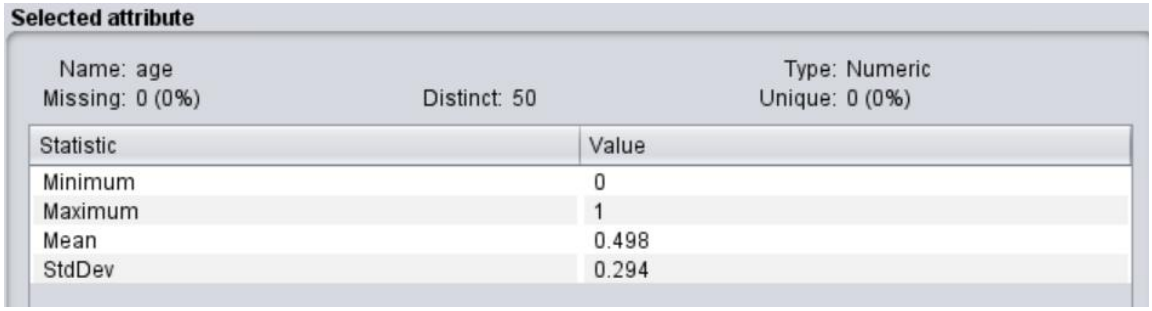

2) 选择两种方法对数据进行归一化，保存到新的数据文件中。并列出归一化的结果。

min-max 标准化：

在 weka 中点击 Filter 下方的 Choose，在指定文件夹下找到 Normalize。




归一化后的结果如下图，并保存到新的文件 Bank-all-2.1.arff。



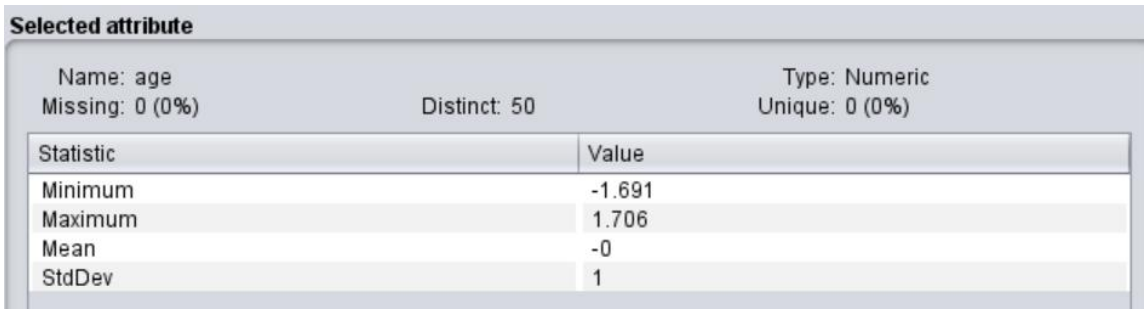
Selected attribute

Name: age　　Type: Numeric
Missing: 0 (0%)　　Distinct: 50　　Unique: 0 (0%)

| Statistic | Value |
|---|---|
| Minimum | 0 |
| Maximum | 1 |
| Mean | 0.498 |
| StdDev | 0.294 |

z-score 标准化：

在 weka 中点击 Filter 下方的 Choose，在指定文件夹下找到 Standardize。点击 Apply，归一化后的结果如下图，并保存到新的文件 Bank-all-2.2.arff。



Selected attribute

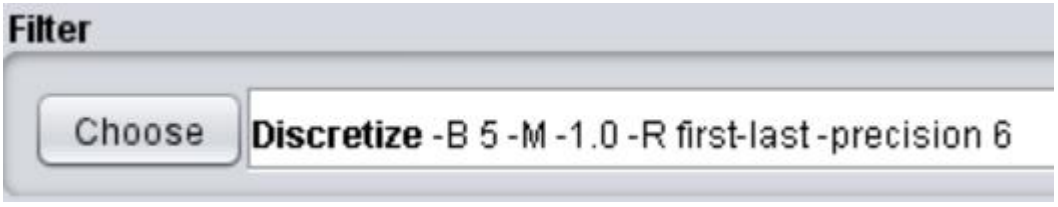
Name: age　　Type: Numeric
Missing: 0 (0%)　　Distinct: 50　　Unique: 0 (0%)

| Statistic | Value |
|---|---|
| Minimum | -1.691 |
| Maximum | 1.706 |
| Mean | -0 |
| StdDev | 1 |

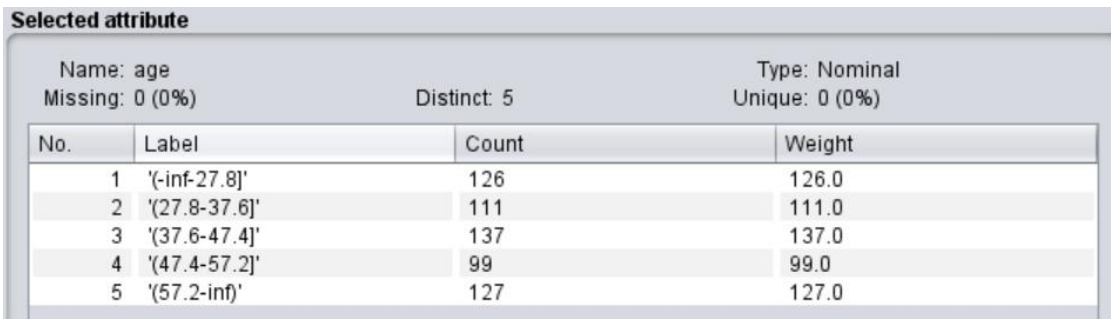
3) 选择两种方法对数据进行离散，保存到新的数据文件中。并列出离散化的结果。

方法一（等宽离散化）：

在 weka 中点击 Filter 下方的 Choose，选择 weka.filters.unsupervised.attribute.Discretize，并选择如下参数。




点击 Apply，离散化后的结果如下图，并保存到新的文件 Bank-all-3.1.arff。



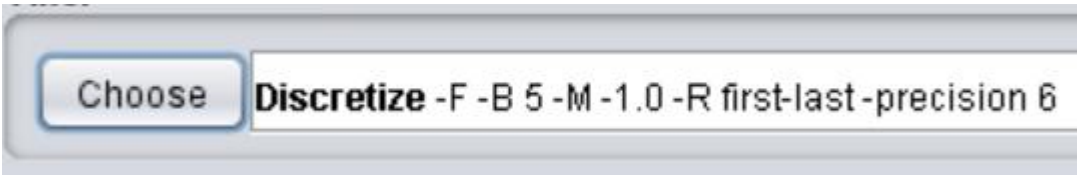
Selected attribute

Name: age　　Type: Nominal
Missing: 0 (0%)　　Distinct: 5　　Unique: 0 (0%)

| No. | Label | Count | Weight |
|---|---|---|---|
| 1 | '(-inf-27.8]' | 126 | 126.0 |
| 2 | '(27.8-37.6]' | 111 | 111.0 |
| 3 | '(37.6-47.4]' | 137 | 137.0 |
| 4 | '(47.4-57.2]' | 99 | 99.0 |
| 5 | '(57.2-inf)' | 127 | 127.0 |

方法二（等频离散化）：

在 weka 中点击 Filter 下方的 Choose，选择 weka.filters.unsupervised.attribute.Discretize，并选择如下参数。




点击 Apply，离散化后的结果如下图，并保存到新的文件 Bank-all-3.2.arff。

**Selected attribute**

Name: age　　Type: Nominal
Missing: 0 (0%)　　Distinct: 5　　Unique: 0 (0%)

| No. | Label | Count | Weight |
|---|---|---|---|
| 1 | '(-inf-27.5]' | 126 | 126.0 |
| 2 | '(27.5-38.5]' | 123 | 123.0 |
| 3 | '(38.5-47.5]' | 125 | 125.0 |
| 4 | '(47.5-58.5]' | 118 | 118.0 |
| 5 | '(58.5-inf)' | 108 | 108.0 |

4) 利用银行原始数据，用 J48 构建决策树。选择 10-crossvalidation。比较 J48 与 binary split 或 multiple split 的结果。分析 "minNumObj "参数的影响（选择 minNumObj=2 或 1）。

a) binary split minNumObj=1

```
=== Summary ===

Correctly Classified Instances         525               87.5    %
Incorrectly Classified Instances        75               12.5    %
Kappa statistic                          0.747
Mean absolute error                      0.1806
Root mean squared error                  0.3431
Relative absolute error                 36.3861 %
Root relative squared error             68.8846 %
Total Number of Instances              600

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.836     0.092     0.884      0.836     0.859      0.748    0.848     0.779     YES
               0.908     0.164     0.868      0.908     0.888      0.748    0.848     0.832     NO
Weighted Avg.  0.875     0.131     0.875      0.875     0.875      0.748    0.848     0.808

=== Confusion Matrix ===

   a   b   <-- classified as
 229  45 |   a = YES
  30 296 |   b = NO
```

b) binary split minNumObj=2

```
=== Summary ===

Correctly Classified Instances         523               87.1667 %
Incorrectly Classified Instances        77               12.8333 %
Kappa statistic                          0.7401
Mean absolute error                      0.1856
Root mean squared error                  0.3451
Relative absolute error                 37.3999 %
Root relative squared error             69.2717 %
Total Number of Instances              600

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.828    0.092    0.883      0.828    0.855      0.741    0.848     0.794     YES
                 0.908    0.172    0.863      0.908    0.885      0.741    0.848     0.818     NO
Weighted Avg.    0.872    0.135    0.872      0.872    0.871      0.741    0.848     0.807

=== Confusion Matrix ===

   a   b   <-- classified as
 227  47 |   a = YES
  30 296 |   b = NO
```

c) multiple split minNumObj=1

```
=== Summary ===

Correctly Classified Instances         546               91      %
Incorrectly Classified Instances        54                9      %
Kappa statistic                          0.8178
Mean absolute error                      0.1559
Root mean squared error                  0.2903
Relative absolute error                 31.4168 %
Root relative squared error             58.2815 %
Total Number of Instances              600

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.872    0.058    0.926      0.872    0.898      0.819    0.893     0.862     YES
                 0.942    0.128    0.898      0.942    0.919      0.819    0.893     0.869     NO
Weighted Avg.    0.910    0.096    0.911      0.910    0.910      0.819    0.893     0.866

=== Confusion Matrix ===

   a   b   <-- classified as
 239  35 |   a = YES
  19 307 |   b = NO
```

d) multiple split minNumObj=2

```
=== Summary ===

Correctly Classified Instances         546               91      %
Incorrectly Classified Instances        54                9      %
Kappa statistic                          0.8178
Mean absolute error                      0.1559
Root mean squared error                  0.2903
Relative absolute error                 31.4168 %
Root relative squared error             58.2815 %
Total Number of Instances              600

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.872     0.058     0.926       0.872   0.898      0.819    0.893     0.862     YES
               0.942     0.128     0.898       0.942   0.919      0.819    0.893     0.869     NO
Weighted Avg.  0.910     0.096     0.911       0.910   0.910      0.819    0.893     0.866

=== Confusion Matrix ===

   a   b   <-- classified as
 239  35 |   a = YES
  19 307 |   b = NO
```

对于利用银行原始数据，用 J48 构建的决策树，分析以上四组的结果。我们不难发现 multiple split 的分类准确率略高于 multiple split，而选择 minNumObj=2 或 1 几乎对实验没有影响。

5) 利用归一化数据用 J48 构建决策树。选择 10-crossvalidation。比较 J48 与 binary split 或 multiple split 的结果。分析 "minNumObj "参数的影响（选择 minNumObj=2 或 1）。

a) binary split minNumObj=1

```
=== Summary ===

Correctly Classified Instances         526               87.6667 %
Incorrectly Classified Instances        74               12.3333 %
Kappa statistic                          0.7504
Mean absolute error                      0.1787
Root mean squared error                  0.3396
Relative absolute error                 36.0083 %
Root relative squared error             68.1725 %
Total Number of Instances              600

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.839     0.092     0.885       0.839   0.861      0.751    0.856     0.792     YES
               0.908     0.161     0.871       0.908   0.889      0.751    0.856     0.839     NO
Weighted Avg.  0.877     0.129     0.877       0.877   0.876      0.751    0.856     0.818

=== Confusion Matrix ===

   a   b   <-- classified as
 230  44 |   a = YES
  30 296 |   b = NO
```

b) binary split minNumObj=2

```
=== Summary ===

Correctly Classified Instances         523               87.1667 %
Incorrectly Classified Instances        77               12.8333 %
Kappa statistic                          0.7401
Mean absolute error                      0.1856
Root mean squared error                  0.3451
Relative absolute error                 37.3999 %
Root relative squared error             69.2717 %
Total Number of Instances              600

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.828    0.092    0.883      0.828    0.855      0.741    0.848     0.794     YES
                 0.908    0.172    0.863      0.908    0.885      0.741    0.848     0.818     NO
Weighted Avg.    0.872    0.135    0.872      0.872    0.871      0.741    0.848     0.807

=== Confusion Matrix ===

   a   b   <-- classified as
 227  47 |   a = YES
  30 296 |   b = NO
```

c) multiple split minNumObj=1

```
=== Summary ===

Correctly Classified Instances         546               91      %
Incorrectly Classified Instances        54                9      %
Kappa statistic                          0.8178
Mean absolute error                      0.1559
Root mean squared error                  0.2903
Relative absolute error                 31.4168 %
Root relative squared error             58.2815 %
Total Number of Instances              600

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.872    0.058    0.926      0.872    0.898      0.819    0.893     0.862     YES
                 0.942    0.128    0.898      0.942    0.919      0.819    0.893     0.869     NO
Weighted Avg.    0.910    0.096    0.911      0.910    0.910      0.819    0.893     0.866

=== Confusion Matrix ===

   a   b   <-- classified as
 239  35 |   a = YES
  19 307 |   b = NO
```

d) multiple split minNumObj=2

```
=== Summary ===

Correctly Classified Instances         546               91      %
Incorrectly Classified Instances        54                9      %
Kappa statistic                          0.8178
Mean absolute error                      0.1559
Root mean squared error                  0.2903
Relative absolute error                 31.4168 %
Root relative squared error             58.2815 %
Total Number of Instances              600

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.872     0.058     0.926      0.872     0.898      0.819     0.893     0.862     YES
               0.942     0.128     0.898      0.942     0.919      0.819     0.893     0.869     NO
Weighted Avg.  0.910     0.096     0.911      0.910     0.910      0.819     0.893     0.866

=== Confusion Matrix ===

   a   b   <-- classified as
 239  35 |   a = YES
  19 307 |   b = NO
```

对于利用归一化后的数据，用 J48 构建的决策树，分析以上四组的结果。我们不难发现 multiple split 的分类准确率略高于 multiple split，而选择 minNumObj=2 或 1 几乎对实验没有影响。结论与 5）中结果类似。

6) 利用离散化数据，用 ID3 构建决策树。比较 ID3 与 binary split 或 multiple split 的结果。分析 "minNumObj "参数的影响（选择 minNumObj=2 或 1）。

由于 weka 中 id3 方法无法更改参数，此处只能展示一个结果。

```
=== Summary ===

Correctly Classified Instances         459               76.5    %
Incorrectly Classified Instances       116               19.3333 %
Kappa statistic                          0.5931
Mean absolute error                      0.1974
Root mean squared error                  0.4418
Relative absolute error                 41.5496 %
Root relative squared error             90.705  %
UnClassified Instances                  25                4.1667 %
Total Number of Instances              600

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.784     0.190     0.772      0.784     0.778      0.593     0.784     0.701     YES
               0.810     0.216     0.821      0.810     0.815      0.593     0.797     0.765     NO
Weighted Avg.  0.798     0.204     0.799      0.798     0.798      0.593     0.791     0.736

=== Confusion Matrix ===

   a   b   <-- classified as
 203  56 |   a = YES
  60 256 |   b = NO
```

7) 对比 J48 和 ID3 的结果。

a) J48 的属性可以是连续值，ID3 的属性必须是离散值。

b) 就本实验的结果来看，J48 的分类效果要好于 ID3。

**2．用归一化数据和离散化数据生成训练（400 个对象）和测试（200 个对象）文件。使用训练数据来训练模型，使用测试数据来测试模型。**

1) 对于归一化数据，比较 J48 中 binary split 或 multiple split 的结果。分析 "minNumObj "参数的影响（选择 minNumObj=2 或 1）。

a) binary split minNumObj=1

```
=== Summary ===

Correctly Classified Instances         173               86.5    %
Incorrectly Classified Instances        27               13.5    %
Kappa statistic                          0.7281
Mean absolute error                      0.1736
Root mean squared error                  0.3527
Relative absolute error                 34.9003 %
Root relative squared error             70.5275 %
Total Number of Instances              200

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.811    0.086    0.895      0.811    0.851      0.731    0.869     0.814     YES
                 0.914    0.189    0.842      0.914    0.877      0.731    0.869     0.857     NO
Weighted Avg.    0.865    0.140    0.867      0.865    0.864      0.731    0.869     0.837

=== Confusion Matrix ===

  a  b   <-- classified as
 77 18 |  a = YES
  9 96 |  b = NO
```

b) binary split minNumObj=2

```
=== Summary ===

Correctly Classified Instances         177               88.5    %
Incorrectly Classified Instances        23               11.5    %
Kappa statistic                          0.7681
Mean absolute error                      0.1693
Root mean squared error                  0.3246
Relative absolute error                 34.0386 %
Root relative squared error             64.9126 %
Total Number of Instances              200

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.821    0.057    0.929      0.821    0.872      0.773    0.887     0.874     YES
                 0.943    0.179    0.853      0.943    0.896      0.773    0.887     0.861     NO
Weighted Avg.    0.885    0.121    0.889      0.885    0.884      0.773    0.887     0.867

=== Confusion Matrix ===

  a  b   <-- classified as
 78 17 |  a = YES
  6 99 |  b = NO
```

c) mutiple split minNumObj=1

```
=== Summary ===

Correctly Classified Instances         175               87.5    %
Incorrectly Classified Instances        25               12.5    %
Kappa statistic                          0.7482
Mean absolute error                      0.1688
Root mean squared error                  0.3368
Total Number of Instances              200

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.821    0.076    0.907      0.821    0.862      0.751    0.883     0.846     YES
                 0.924    0.179    0.851      0.924    0.886      0.751    0.883     0.871     NO
Weighted Avg.    0.875    0.130    0.878      0.875    0.874      0.751    0.883     0.859

=== Confusion Matrix ===

  a  b   <-- classified as
 78 17 |  a = YES
  8 97 |  b = NO
```

d) mutiple split minNumObj=2

```
=== Summary ===

Correctly Classified Instances         177               88.5    %
Incorrectly Classified Instances        23               11.5    %
Kappa statistic                          0.7681
Mean absolute error                      0.1685
Root mean squared error                  0.3248
Total Number of Instances              200

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.821    0.057    0.929      0.821    0.872      0.773    0.886     0.878     YES
                 0.943    0.179    0.853      0.943    0.896      0.773    0.886     0.856     NO
Weighted Avg.    0.885    0.121    0.889      0.885    0.884      0.773    0.886     0.866

=== Confusion Matrix ===

  a  b   <-- classified as
 78 17 |  a = YES
  6 99 |  b = NO
```

对于不同参数模型在测试集上的表现，我们发现 binary split 和 mutiple split 的结果相近，minNumObj=2 时的结果略好于 minNumObj=1 时的结果。

2) 对于离散数据，比较 ID3 与 binary split 或 multiple split 的结果。

分析 "minNumObj "参数的影响（选择 minNumObj=2 或 1）。

由于 weka 中 id3 方法无法更改参数，此处只能展示一个结果。

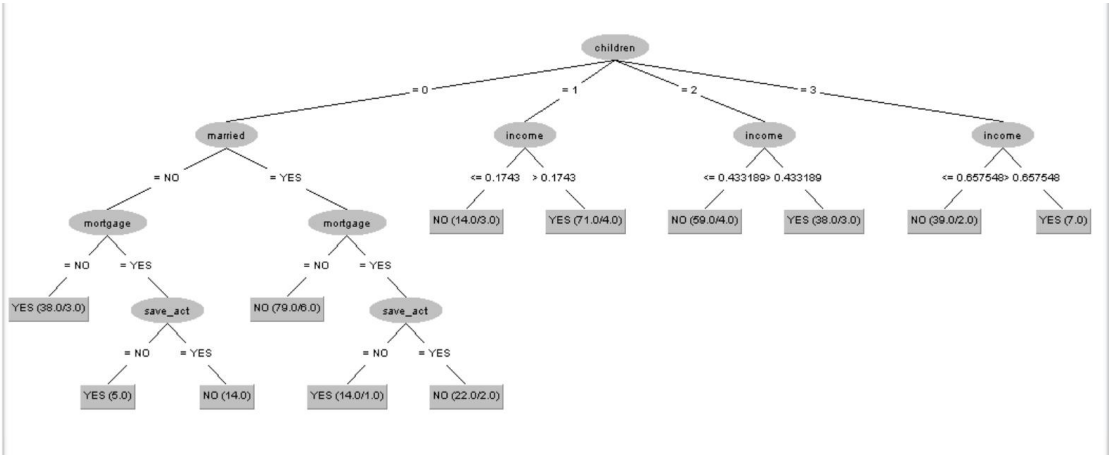
```
=== Summary ===

Correctly Classified Instances         157               78.5    %
Incorrectly Classified Instances        39               19.5    %
Kappa statistic                          0.5995
Mean absolute error                      0.199
Root mean squared error                  0.4432
UnClassified Instances                   4                2      %
Total Number of Instances              200

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.753    0.155    0.814      0.753    0.782      0.601    0.795     0.730     YES
                 0.845    0.247    0.791      0.845    0.817      0.601    0.796     0.748     NO
Weighted Avg.    0.801    0.204    0.802      0.801    0.800      0.601    0.795     0.739

=== Confusion Matrix ===

  a  b   <-- classified as
 70 23 |  a = YES
 16 87 |  b = NO
```

**需要注意的是，测试集和训练集中离散化的标准需要相同，否则无法使用测试集对模型进行训练。**

利用混淆矩阵计算准确率、错误率、精确率和召回率。写下计算的过程。用一些可视化的结果来展示你的结果。

J48：

j48 的决策树可视化结果相似，这里仅展示 mutiple split minNumObj=2 的可视化决策树。




混淆矩阵：

```
=== Confusion Matrix ===

  a  b   <-- classified as
 78 17 |  a = YES
  6 99 |  b = NO
```

准确率=(78+99)/200=88.5%
错误率=1-88.5%=11.5%
精确率=78/(78+6)=92.9%
召回率=78/(78+17)=82.1%

## ID3：

ID3 无法很好地生成可视化决策树。只能使用如下结果进行展示。

```
children = 0
|   married = NO
|   |   mortgage = NO
|   |   |   age = '(-inf-27.8]': YES
|   |   |   age = '(27.8-37.6]'
|   |   |   |   region = INNER_CITY: YES
|   |   |   |   region = TOWN
|   |   |   |   |   sex = FEMALE: NO
|   |   |   |   |   sex = MALE
|   |   |   |   |   |   income = a: NO
|   |   |   |   |   |   income = b: YES
|   |   |   |   |   |   income = c: null
|   |   |   |   |   |   income = d: null
|   |   |   |   |   |   income = e: null
|   |   |   |   region = RURAL: YES
|   |   |   |   region = SUBURBAN: null
|   |   |   age = '(37.6-47.4]': YES
|   |   |   age = '(47.4-57.2]'
|   |   |   |   income = a: NO
|   |   |   |   income = b: YES
|   |   |   |   income = c: YES
|   |   |   |   income = d: null
|   |   |   |   income = e: null
|   |   |   age = '(57.2-inf)': YES
|   |   mortgage = YES
|   |   |   save_act = NO: YES
|   |   |   save_act = YES: NO
|   married = YES
|   |   mortgage = NO
|   |   |   income = a
|   |   |   |   sex = FEMALE: NO
|   |   |   |   sex = MALE
|   |   |   |   |   age = '(-inf-27.8]'
|   |   |   |   |   |   car = NO: NO
|   |   |   |   |   |   car = YES: YES
|   |   |   |   |   age = '(27.8-37.6]': NO
|   |   |   |   |   age = '(37.6-47.4]': YES
|   |   |   |   |   age = '(47.4-57.2]': null
|   |   |   |   |   age = '(57.2-inf)': null
|   |   |   income = b
|   |   |   |   age = '(-inf-27.8]'
|   |   |   |   |   region = INNER_CITY: YES
|   |   |   |   |   region = TOWN: NO
|   |   |   |   |   region = RURAL: NO
|   |   |   |   |   region = SUBURBAN: NO
|   |   |   |   age = '(27.8-37.6]'
|   |   |   |   |   region = INNER_CITY
|   |   |   |   |   |   sex = FEMALE: NO
|   |   |   |   |   |   sex = MALE: YES
|   |   |   |   |   region = TOWN: NO
|   |   |   |   |   region = RURAL: null
|   |   |   |   |   region = SUBURBAN: null
|   |   |   |   age = '(37.6-47.4]': NO
|   |   |   |   age = '(47.4-57.2]': NO
|   |   |   |   age = '(57.2-inf)'
|   |   |   |   |   sex = FEMALE: YES
|   |   |   |   |   sex = MALE: NO
|   |   |   income = c: NO
|   |   |   income = d: NO
|   |   |   income = e: NO
|   |   mortgage = YES
|   |   |   save_act = NO
|   |   |   |   region = INNER_CITY: YES
|   |   |   |   region = TOWN
|   |   |   |   |   age = '(-inf-27.8]': null
|   |   |   |   |   age = '(27.8-37.6]': NO
|   |   |   |   |   age = '(37.6-47.4]': null
|   |   |   |   |   age = '(47.4-57.2]': YES
|   |   |   |   |   age = '(57.2-inf)': null
|   |   |   |   region = RURAL: YES
|   |   |   |   region = SUBURBAN: YES
|   |   |   save_act = YES
|   |   |   |   region = INNER_CITY
|   |   |   |   |   income = a: NO
|   |   |   |   |   income = b: NO
|   |   |   |   |   income = c
|   |   |   |   |   |   age = '(-inf-27.8]': null
|   |   |   |   |   |   age = '(27.8-37.6]': null
|   |   |   |   |   |   age = '(37.6-47.4]': YES
|   |   |   |   |   |   age = '(47.4-57.2]': null
|   |   |   |   |   |   age = '(57.2-inf)': NO
|   |   |   |   |   income = d: NO
|   |   |   |   |   income = e: null
|   |   |   |   region = TOWN: NO
|   |   |   |   region = RURAL: NO
|   |   |   |   region = SUBURBAN
|   |   |   |   |   age = '(-inf-27.8]': null
|   |   |   |   |   age = '(27.8-37.6]': NO
|   |   |   |   |   age = '(37.6-47.4]': NO
|   |   |   |   |   age = '(47.4-57.2]': null
|   |   |   |   |   age = '(57.2-inf)': YES
children = 1
|   income = a
|   |   age = '(-inf-27.8]'
|   |   |   sex = FEMALE: NO
|   |   |   sex = MALE
|   |   |   |   married = NO: NO
|   |   |   |   married = YES
|   |   |   |   |   save_act = NO: NO
|   |   |   |   |   save_act = YES: YES
|   |   age = '(27.8-37.6]'
|   |   |   region = INNER_CITY
|   |   |   |   sex = FEMALE: NO
|   |   |   |   sex = MALE
|   |   |   |   |   save_act = NO: NO
|   |   |   |   |   save_act = YES
|   |   |   |   |   |   current_act = NO: NO
|   |   |   |   |   |   current_act = YES: YES
|   |   |   region = TOWN: YES
|   |   |   region = RURAL: null
|   |   |   region = SUBURBAN: null
|   |   age = '(37.6-47.4]'
|   |   |   region = INNER_CITY: NO
|   |   |   region = TOWN: YES
|   |   |   region = RURAL: null
|   |   |   region = SUBURBAN: null
|   |   age = '(47.4-57.2]': YES
|   |   age = '(57.2-inf)': null
|   income = b
|   |   sex = FEMALE
|   |   |   region = INNER_CITY: YES
|   |   |   region = TOWN
|   |   |   |   car = NO: YES
|   |   |   |   car = YES: NO
|   |   |   region = RURAL: NO
|   |   |   region = SUBURBAN: YES
|   |   sex = MALE: YES
|   income = c
|   |   mortgage = NO: YES
|   |   mortgage = YES
|   |   |   age = '(-inf-27.8]': null
|   |   |   age = '(27.8-37.6]': NO
|   |   |   age = '(37.6-47.4]': YES
|   |   |   age = '(47.4-57.2]': YES
|   |   |   age = '(57.2-inf)'
|   |   |   |   sex = FEMALE: YES
|   |   |   |   sex = MALE: NO
|   income = d: YES
|   income = e: YES
children = 2
|   income = a
|   |   region = INNER_CITY
|   |   |   age = '(-inf-27.8]': NO
|   |   |   age = '(27.8-37.6]'
```

```
|   |   |   |   sex = FEMALE: NO
|   |   |   |   sex = MALE: YES
|   |   |   age = '(37.6-47.4]': null
|   |   |   age = '(47.4-57.2]': null
|   |   |   age = '(57.2-inf)': null
|   |   region = TOWN: NO
|   |   region = RURAL
|   |   |   age = '(-inf-27.8]': NO
|   |   |   age = '(27.8-37.6]': YES
|   |   |   age = '(37.6-47.4]': null
|   |   |   age = '(47.4-57.2]': null
|   |   |   age = '(57.2-inf)': null
|   |   region = SUBURBAN: YES
|   income = b
|   |   age = '(-inf-27.8]'
|   |   |   current_act = NO
|   |   |   |   sex = FEMALE: NO
|   |   |   |   sex = MALE: YES
|   |   |   current_act = YES: NO
|   |   age = '(27.8-37.6]': NO
|   |   age = '(37.6-47.4]': NO
|   |   age = '(47.4-57.2]': NO
|   |   age = '(57.2-inf)': NO
|   income = c
|   |   age = '(-inf-27.8]': null
|   |   age = '(27.8-37.6]': YES
|   |   age = '(37.6-47.4]'
|   |   |   mortgage = NO
|   |   |   |   sex = FEMALE
|   |   |   |   |   married = NO: YES
|   |   |   |   |   married = YES: NO
|   |   |   |   sex = MALE: YES
|   |   |   mortgage = YES: NO
|   |   age = '(47.4-57.2]'
|   |   |   region = INNER_CITY: YES
|   |   |   region = TOWN
|   |   |   |   sex = FEMALE: YES
|   |   |   |   sex = MALE: NO
|   |   |   region = RURAL: null
|   |   |   region = SUBURBAN: YES
|   |   age = '(57.2-inf)': YES
|   income = d
|   |   age = '(-inf-27.8]': null
|   |   age = '(27.8-37.6]': null
|   |   age = '(37.6-47.4]': YES
|   |   age = '(47.4-57.2]': YES
|   |   age = '(57.2-inf)'
|   |   |   sex = FEMALE: YES
|   |   |   sex = MALE: YES
|   income = e: YES
children = 3
|   income = a: NO
|   income = b
|   |   age = '(-inf-27.8]': NO
|   |   age = '(27.8-37.6]': NO
|   |   age = '(37.6-47.4]': NO
|   |   age = '(47.4-57.2]'
|   |   |   sex = FEMALE
|   |   |   |   save_act = NO: NO
|   |   |   |   save_act = YES: YES
|   |   |   sex = MALE: NO
|   |   age = '(57.2-inf)': NO
|   income = c
|   |   age = '(-inf-27.8]': null
|   |   age = '(27.8-37.6]'
|   |   |   sex = FEMALE: NO
|   |   |   sex = MALE: YES
|   |   age = '(37.6-47.4]': NO
|   |   age = '(47.4-57.2]': NO
|   |   age = '(57.2-inf)': NO
|   income = d
|   |   mortgage = NO
|   |   |   age = '(-inf-27.8]': null
|   |   |   age = '(27.8-37.6]': null
|   |   |   age = '(37.6-47.4]': YES
|   |   |   age = '(47.4-57.2]'
|   |   |   |   sex = FEMALE: YES
|   |   |   |   sex = MALE: YES
|   |   |   age = '(57.2-inf)': YES
|   |   mortgage = YES: NO
|   income = e: YES
```

混淆矩阵：

```
=== Confusion Matrix ===

  a  b   <-- classified as
 70 23 |  a = YES
 16 87 |  b = NO
```

准确率=(70+87)/200=78.5%

错误率=1-78.5%=11.5%

精确率=70/(70+6)=92.1%

召回率=70/(70+23)=75.3%

3. **Data: weather-nominal.arff, which is included in the path of weka.**

1) use weka with ID3 to construct a tree.

使用 weka 构造的决策树如下：

```
outlook = sunny
|   humidity = high: no
|   humidity = normal: yes
outlook = overcast: yes
outlook = rainy
|   windy = TRUE: no
|   windy = FALSE: yes
```

混淆矩阵如下：

```
=== Confusion Matrix ===

 a b   <-- classified as
 8 1 | a = yes
 1 4 | b = no
```

2) construct a tree manually

| No. | 1: outlook | 2: temperature | 3: humidity | 4: windy | 5: **play** |
|---|---|---|---|---|---|
| | Nominal | Nominal | Nominal | Nominal | Nominal |
| 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 |

数据集S的信息熵为 $H(S) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.9403$

$H_{outlook}(S) = \frac{5}{14} \times -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.6936$

$Gain_0(Outlook) = 0.9403 - 0.6936 = 0.2467$

同理 $Gain_0(temperature) = 0.0292$

$Gain_0(humidity) = 0.1518$

$Gain_0(windy) = 0.0481$

因此选择 outlook属性分裂

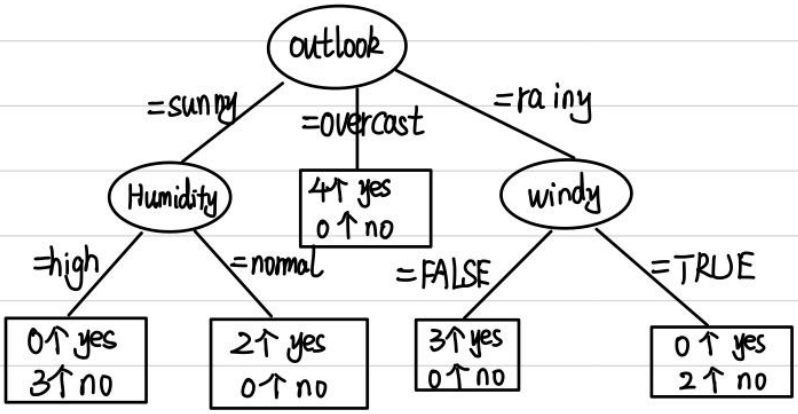
再对 sunny 属性划分

$Gain_1(temperature) = 0.4000$

$Gain_1(humidity) = 0.9710$

$Gain_1(windy) = 0.0200$

剩余划分类似，最终决策树：




3) compare the upper two methods.

对比以上两种方法，结果是相同的。