

Using classifiers for mail promotions. Part I. Building response predictor

Lab 2.1

Lab consists of two parts: classification and business analysis

- ▶ Part I. Data mining: build the classifier and use it for the prediction of potential responders
- Part II. Business analytics: how to design the most profitable campaign

Plan

Part I. Data Mining. Classification with WEKA.

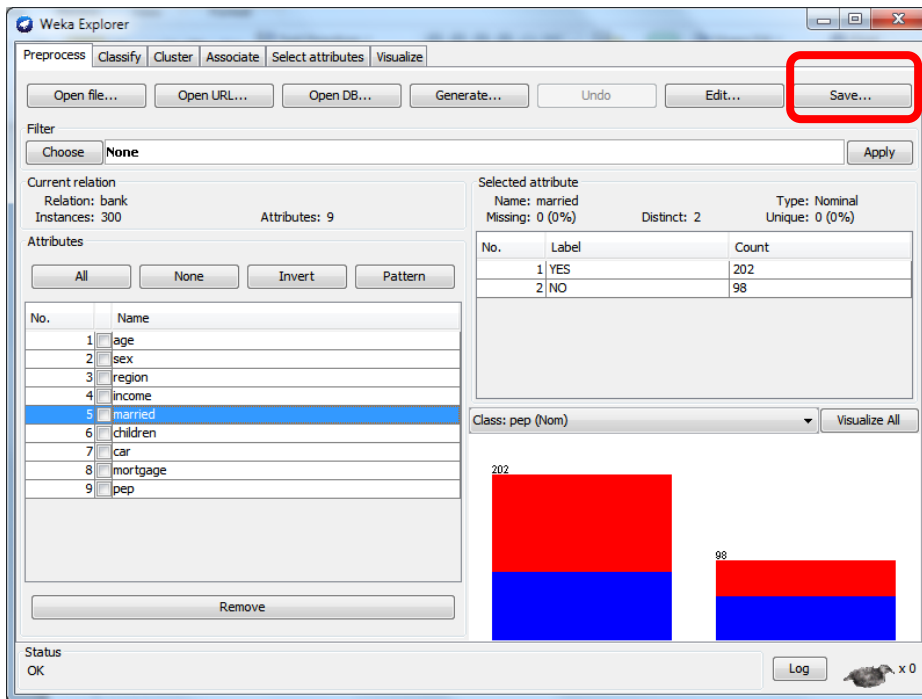
1. Prepare data
2. Build several classifiers. Choose the most accurate one.
3. Divide dataset into training and validation datasets
4. Predict class in the validation dataset
5. Prepare output for business analysis

Part II. Business analysis

1. Generate Lift chart(s)
2. Cost-benefit analysis
3. Recommendations

Dataset

- Load bank_data.csv into WEKA explorer
- Save file as bank1.arff



Part I. Data Mining.

1. Prepare data
2. Build several classifiers. Choose the most accurate one.
3. Divide dataset into training and validation datasets
4. Predict class in the validation dataset
5. Prepare output for business analysis

Dataset: explore available attributes in text editor

- @relation bank-data
- @attribute id
{ID12101,ID12102,ID12103,ID12104,ID12105,ID12106,ID12107,ID12108,ID12109,ID12110,ID12111,ID12112,ID12113,ID12114,ID12115,ID12116,ID12117,ID12118,ID12119,ID12120,ID12121,ID12122,ID12123,ID12124,ID12125,ID12126,ID12127,ID12128,ID12129,ID12130,ID12131,ID12132,ID12133,ID12134,ID12135,ID12136,ID12137,ID12138,ID12139,ID12140,ID12141,ID12142,ID12143,ID12144,ID12145,ID12146,ID12147,ID12148,ID12149,ID12150,ID12151,ID12152,ID12153,ID12154,ID12155,ID12156,ID12157,ID12158,ID12159,ID12160,ID12161,ID12162}
- ...
- @attribute age numeric
- @attribute sex {FEMALE,MALE}
- @attribute region {INNER_CITY,TOWN,RURAL,SUBURBAN}
- @attribute income numeric
- @attribute married {NO,YES}
- @attribute children numeric
- @attribute car {NO,YES}
- @attribute save_act {NO,YES}
- @attribute current_act {NO,YES}
- @attribute mortgage {NO,YES}
- @attribute pep {YES,NO}

Class attribute:
bought Personal
Equity Plan after the
last mailing

Part I. Data Mining.

1. Prepare data
2. Build several classifiers.
Choose the most accurate one.
3. Divide dataset into training and validation datasets
4. Predict class in the validation dataset
5. Prepare output for business analysis

Dataset: working with attributes

- @relation bank-data
- @attribute id {ID12101,ID12102,ID12103,ID12104,ID12105,ID12106,ID12107,ID12108,ID12109,ID12110,ID12111,ID12112,ID12113,ID12114,ID12115,ID12116,ID12117,ID12118,ID12119,ID12120,ID12121,ID12122,ID12123,ID12124,ID12125,ID12126,ID12127,ID12128,ID12129,ID12130,ID12131,ID12132,ID12133,ID12134,ID12135,ID12136,ID12137,ID12138,ID12139,ID12140,ID12141,ID12142,ID12143,ID12144,ID12145,ID12146,ID12147,ID12148,ID12149,ID12150,ID12151,ID12152,ID12153,ID12154,ID12155,ID12156,ID12157,ID12158,ID12159,ID12160,ID12161,ID12162,ID12163,ID12164,ID12165,ID12166,ID12167,ID12168,ID12169,ID12170,ID12171,ID12172,ID12173,ID12174,ID12175...}
- @attribute age numeric
- @attribute sex {FEMALE,MALE}
- @attribute region {INNER_CITY,TOWN,RURAL,SUBURBAN}
- @attribute income numeric
- @attribute married {NO,YES}
- @attribute children numeric
- @attribute car {NO,YES}
- @attribute save_act {NO,YES}
- @attribute current_act {NO,YES}
- @attribute mortgage {NO,YES}
- @attribute pep {YES,NO}

Non-predictive
attribute: remove it
and save file

Part I. Data Mining.

1. Prepare data
2. Build several classifiers. Choose the most accurate one.
3. Divide dataset into training and validation datasets
4. Predict class in the validation dataset
5. Prepare output for business analysis

Dataset: working with attributes

- @relation bank-data
- @attribute age numeric
- @attribute sex {FEMALE,MALE}
- @attribute region {INNER_CITY,TOWN,RURAL,SUBURBAN}
- @attribute income numeric
- @attribute married {NO,YES}
- @attribute children numeric
- @attribute car {NO,YES}
- @attribute save_act {NO,YES}
- @attribute current_act {NO,YES}
- @attribute mortgage {NO,YES}
- @attribute pep {YES,NO}

Numeric attributes
age and income:
discretize into groups

Part I. Data Mining.

1. Prepare data
2. Build several classifiers. Choose the most accurate one.
3. Divide dataset into training and validation datasets
4. Predict class in the validation dataset
5. Prepare output for business analysis

Discretize numeric attributes

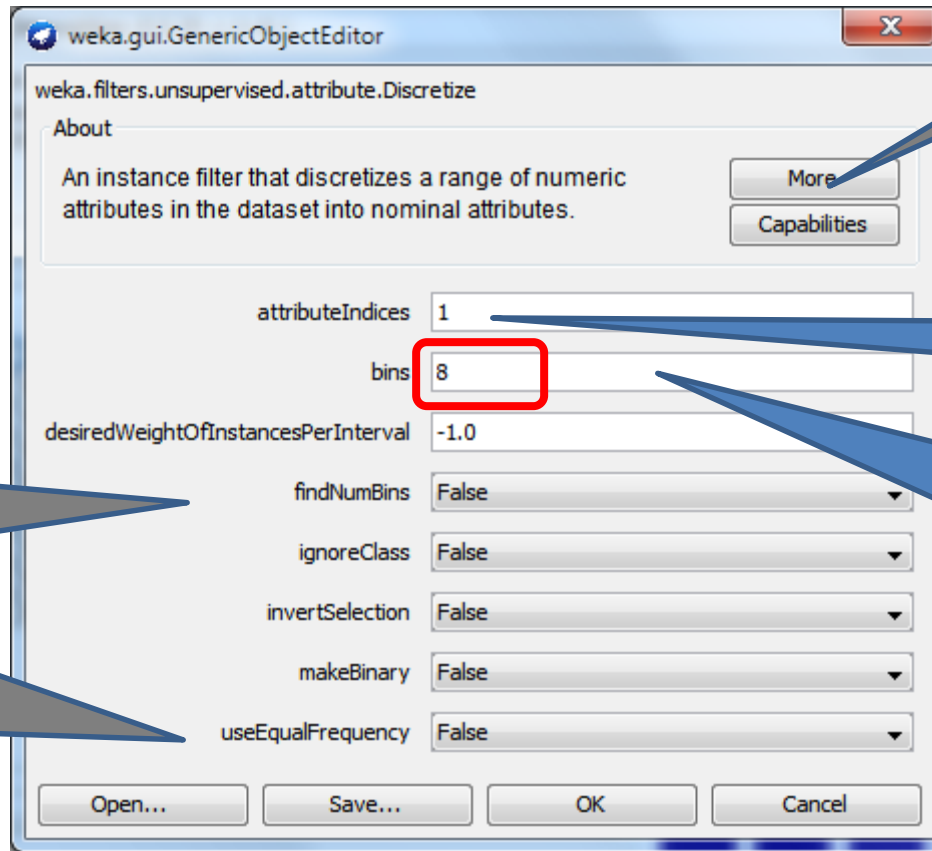
- Simple discretization techniques: distribute numeric values into a predefined number of bins
 - Equal intervals: the bins are defined as equal-size numeric intervals
 - Equal frequency: the bins are defined such as to contain equal number of instances in each interval
- In WEKA: Filter: Choose -> Filters-> Unsupervised -> Attribute-> Discretize.
- Left-click to open parameters window

Part I. Data Mining.

1. Prepare data
2. Build several classifiers. Choose the most accurate one.
3. Divide dataset into training and validation datasets
4. Predict class in the validation dataset
5. Prepare output for business analysis

Discretize numeric attributes

- Age



Explains parameters

Index of the attribute to apply filter on: 1

Number of bins: based on min-max values and common sense

Finds optimal number of bins by data mining techniques

If true - equal frequency binning, if false – equal interval binning

The number of bins is found experimentally, by observing the distribution of the class label in different bins. To play with different settings, use the Undo button

Discretize numeric attributes

- Age after discretization

Current relation
Relation: bank-weka.filters.unsupervised.attribute.Discretize-B6-M-1...
Instances: 300 Attributes: 9

Attributes

All None Invert Pattern

| No. | Name |
|-------------------------------------|------------|
| <input checked="" type="checkbox"/> | 1 age |
| <input type="checkbox"/> | 2 sex |
| <input type="checkbox"/> | 3 region |
| <input type="checkbox"/> | 4 income |
| <input type="checkbox"/> | 5 married |
| <input type="checkbox"/> | 6 children |
| <input type="checkbox"/> | 7 car |
| <input type="checkbox"/> | 8 mortgage |
| <input type="checkbox"/> | 9 pep |

Remove

Status

Selected attribute
Name: age
Missing: 0 (0%) Distinct: 8
Type: Nominal
Unique: 0 (0%)

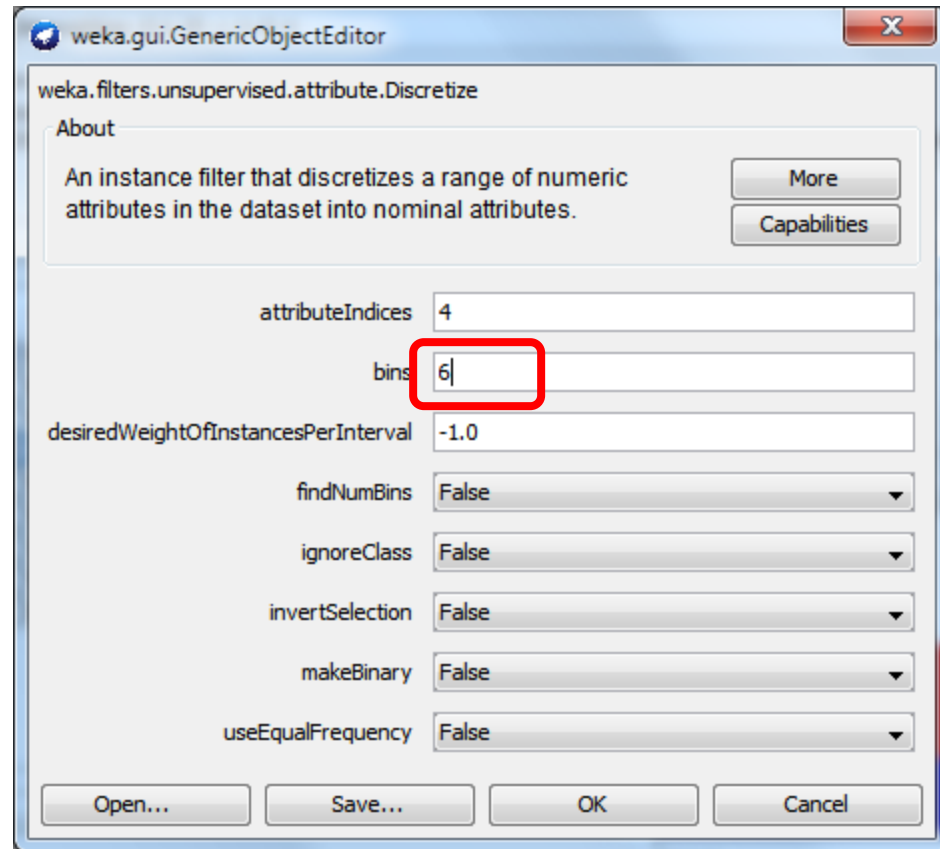
| No. | Label | Count |
|-----|------------------|-------|
| 1 | '(-inf-24.125]' | 41 |
| 2 | '(24.125-30.25]' | 36 |
| 3 | '(30.25-36.375]' | 36 |
| 4 | '(36.375-42.5]' | 36 |
| 5 | '(42.5-48.625]' | 45 |
| 6 | '(48.625-54.75]' | 33 |
| 7 | '(54.75-60.875]' | 26 |
| 8 | '(60.875-inf]' | 47 |

Class: pep (Nom) Visualize All

| Bin No. | Label | Count |
|---------|------------------|-------|
| 1 | '(-inf-24.125]' | 41 |
| 2 | '(24.125-30.25]' | 36 |
| 3 | '(30.25-36.375]' | 36 |
| 4 | '(36.375-42.5]' | 36 |
| 5 | '(42.5-48.625]' | 45 |
| 6 | '(48.625-54.75]' | 33 |
| 7 | '(54.75-60.875]' | 26 |
| 8 | '(60.875-inf]' | 47 |

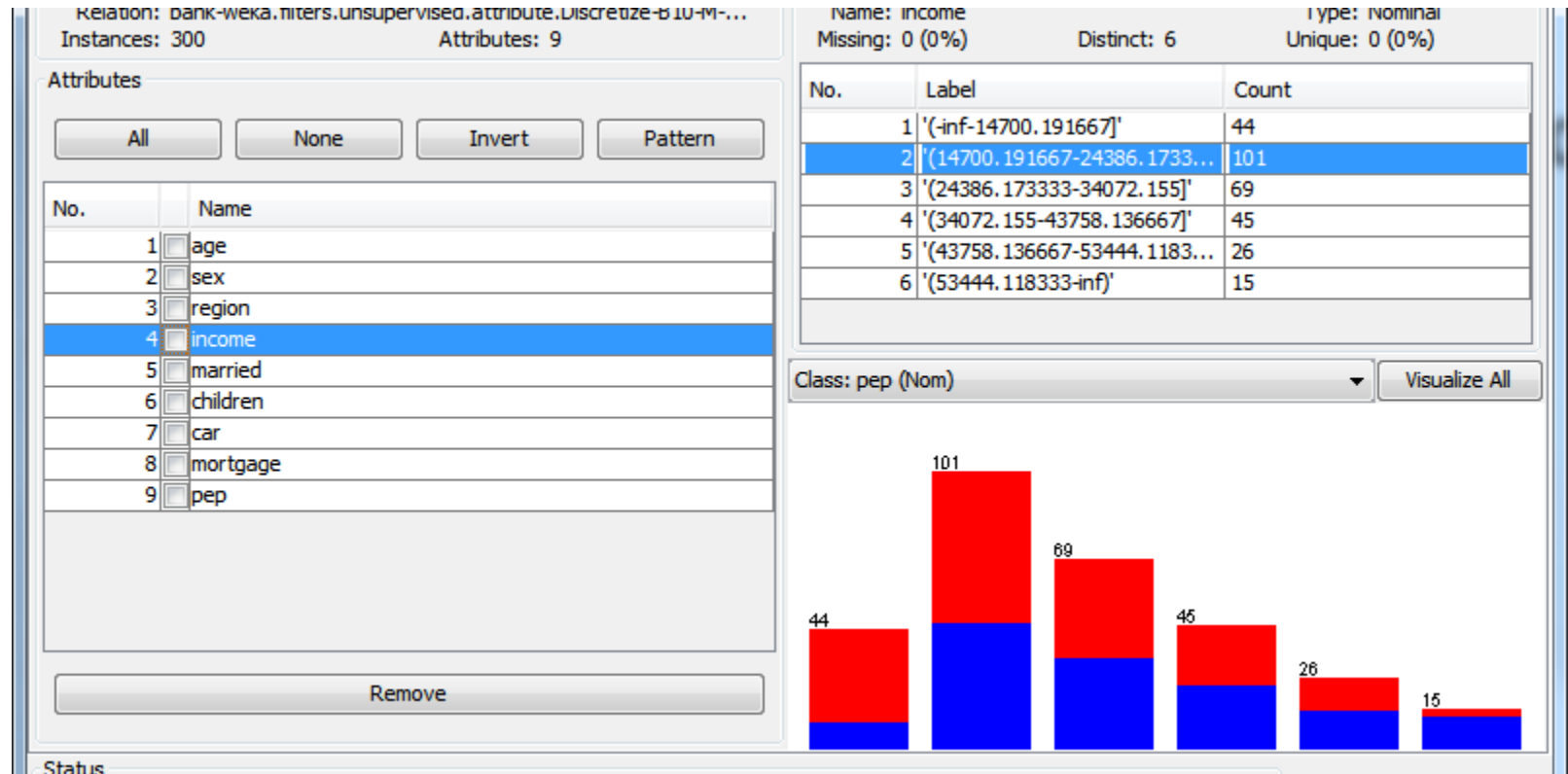
Discretize numeric attributes

- Income



Discretize numeric attributes

- Income after discretization



Optional. [Binaryze multi-valued attributes]

- Undo

The screenshot shows the Weka Explorer interface with the 'NominalToBinary' filter applied to the 'region=RURAL' attribute. The 'Undo' button is highlighted in the top toolbar. The 'Current relation' is 'bank-weka.filters.unsupervised.attribute.Discretize-B6-M-1....' with 300 instances and 12 attributes. The 'Selected attribute' is 'region=RURAL', which is a numeric attribute with 2 distinct values and 0 missing values. The 'Class' is 'pep (Nom)'. A bar chart shows the distribution of the attribute values, with 249 instances for value 0 and 51 instances for value 1.

Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Open file... | Open URL... | Open DB... | Generate... | Undo | Edit... | Save...

Filter: Choose **NominalToBinary -R 3** Apply

Current relation
Relation: bank-weka.filters.unsupervised.attribute.Discretize-B6-M-1....
Instances: 300 Attributes: 12

Attributes
All | None | Invert | Pattern

| No. | Name |
|-----|--|
| 1 | <input type="checkbox"/> age |
| 2 | <input type="checkbox"/> sex |
| 3 | <input type="checkbox"/> region=INNER_CITY |
| 4 | <input checked="" type="checkbox"/> region=RURAL |
| 5 | <input type="checkbox"/> region=TOWN |
| 6 | <input type="checkbox"/> region=SUBURBAN |
| 7 | <input type="checkbox"/> income |
| 8 | <input type="checkbox"/> married |
| 9 | <input type="checkbox"/> children |
| 10 | <input type="checkbox"/> car |
| 11 | <input type="checkbox"/> mortgage |
| 12 | <input type="checkbox"/> pep |

Remove

Selected attribute
Name: region=RURAL Type: Numeric
Missing: 0 (0%) Distinct: 2 Unique: 0 (0%)

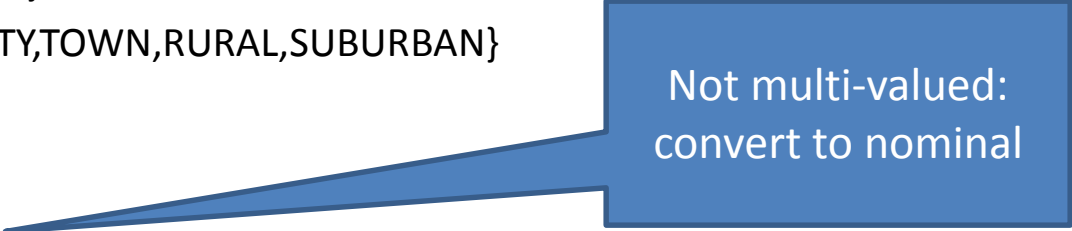
| Statistic | Value |
|-----------|-------|
| Minimum | 0 |
| Maximum | 1 |
| Mean | 0.17 |
| StdDev | 0.376 |

Class: pep (Nom) Visualize All

Status: OK Log x 0

Dataset: working with attributes - children

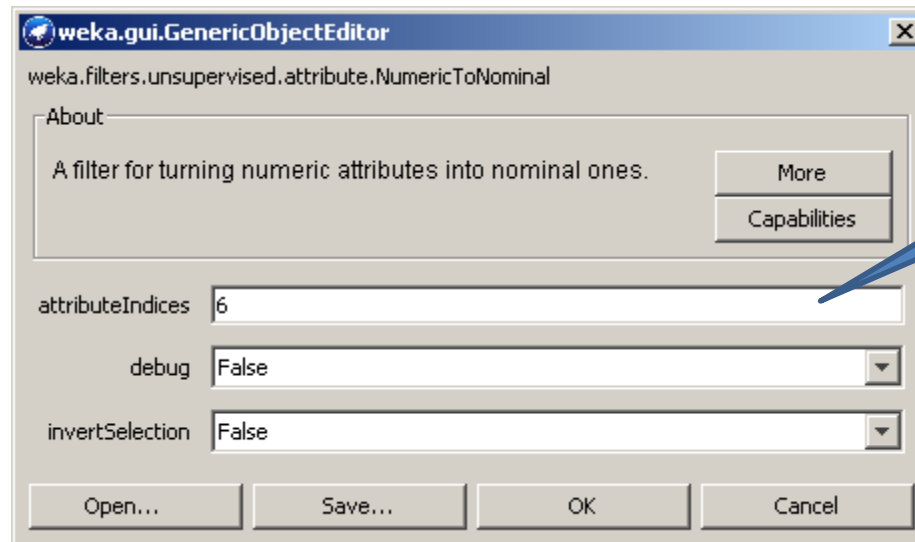
- @relation bank-data
- @attribute age numeric
- @attribute sex {FEMALE,MALE}
- @attribute region {INNER_CITY,TOWN,RURAL,SUBURBAN}
- @attribute income numeric
- @attribute married {NO,YES}
- @attribute children numeric
- @attribute car {NO,YES}
- @attribute save_act {NO,YES}
- @attribute current_act {NO,YES}
- @attribute mortgage {NO,YES}
- @attribute pep {YES,NO}



Not multi-valued:
convert to nominal

Convert numeric to nominal

- Filters->Unsupervised->attribute -> NumericToNominal



Index of the attribute
to apply filter on: 6

Convert numeric to nominal

- Children after nominalizations: 4 groups

The screenshot shows the Weka Explorer interface with the NumericToNominal filter applied to the 'children' attribute. The filter settings are visible in the Filter section, and the resulting nominal distribution is shown in the Selected attribute section.

Current relation:
Relation: bank-data-weka.filters.unsupervised.attribute.Remove-R1-...
Instances: 600
Attributes: 11

Selected attribute:
Name: children
Missing: 0 (0%)
Distinct: 4
Type: Nominal
Unique: 0 (0%)

| No. | Label | Count |
|-----|-------|-------|
| 1 | 0 | 263 |
| 2 | 1 | 135 |
| 3 | 2 | 134 |
| 4 | 3 | 68 |

Class: pep (Nom)

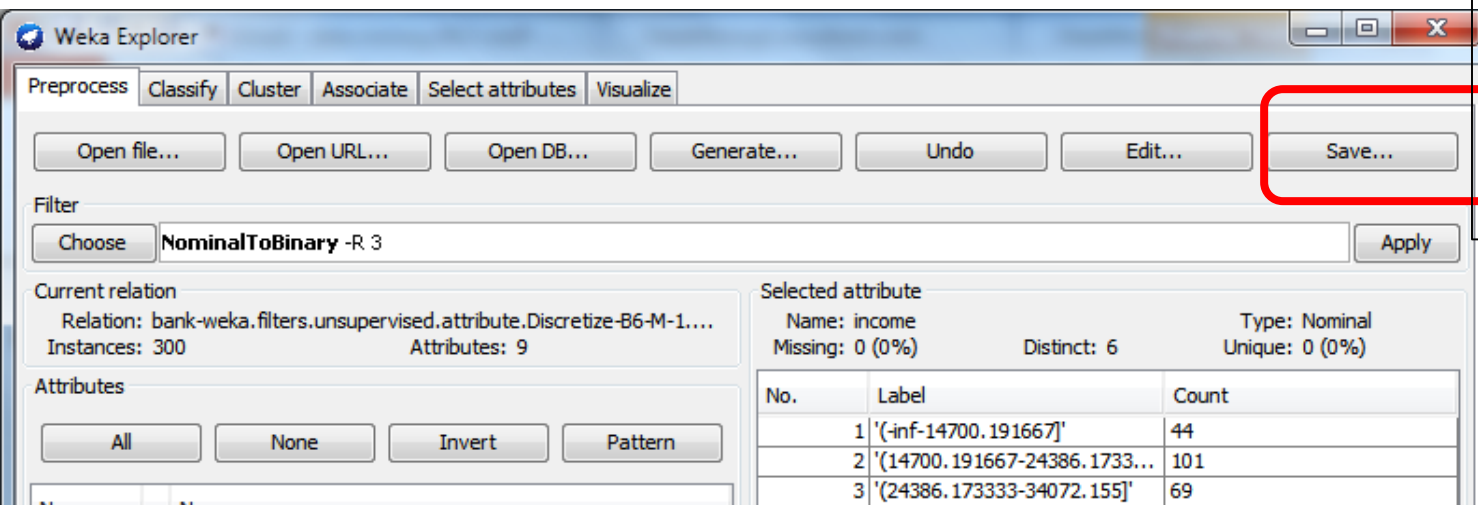
Status: OK

Save the resulting dataset as 'bank2.arff'

- This is the input to our classifiers

Part I. Data Mining.

1. Prepare data
2. Build several classifiers.
Choose the most accurate one.
3. Divide dataset into training and validation datasets
4. Predict class in the validation dataset
5. Prepare output for business analysis



Classification

- Our goal: the most accurate classifier

| Algorithm | Dataset | Accuracy |
|-----------|---------|----------|
| | | |
| | | |
| | | |
| | | |

Part I. Data Mining.

1. Prepare data
2. Build several classifiers. Choose the most accurate one.
3. Divide dataset into training and validation datasets
4. Predict class in the validation dataset
5. Prepare output for business analysis

Classification. Trees: J48

Weka Explorer

Preprocess **Classify** Cluster Associate Select attributes Visualize

Classifier: Choose **J48 -C 0.25 -M 2**

Test options:

- Use training set
- Supplied test set (Set...)
- Cross-validation Folds: 10
- Percentage split %: 66

More options...

(Nom) pep

Start Stop

Result list (right-click for options)

09:59:30 - trees_J48

Classifier output

Time taken to build model: 0.08 seconds

=== Stratified cross-validation ===

=== Summary ===

| | | |
|----------------------------------|-----------|--------|
| Correctly Classified Instances | 537 | 89.5 % |
| Incorrectly Classified Instances | 63 | 10.5 % |
| Kappa statistic | 0.7866 | |
| Mean absolute error | 0.1767 | |
| Root mean squared error | 0.3063 | |
| Relative absolute error | 35.6033 % | |
| Root relative squared error | 61.497 % | |
| Total Number of Instances | 600 | |

=== Detailed Accuracy By Class ===

| | TP Rate | FP Rate | Precision | Recall | F-Measure | ROC Area |
|---------------|---------|---------|-----------|--------|-----------|----------|
| | 0.832 | 0.052 | 0.931 | 0.832 | 0.879 | 0.902 |
| | 0.948 | 0.168 | 0.87 | 0.948 | 0.907 | 0.902 |
| Weighted Avg. | 0.895 | 0.115 | 0.898 | 0.895 | 0.894 | 0.902 |

=== Confusion Matrix ===

Status: OK Log x 0

Accuracy:
89.5%

Part I. Data Mining.

1. Prepare data
2. Build several classifiers. Choose the most accurate one.
3. Divide dataset into training and validation datasets
4. Predict class in the validation dataset
5. Prepare output for business analysis

Report

| Algorithm | Dataset | Accuracy |
|-----------|------------|----------|
| J48 | bank2.arff | 89.5 |
| | | |
| | | |
| | | |

Attribute selection. Decision tree: J48

- The most important attributes (used in the tree for splitting nodes): children, married, mortgage, save_act, income
- Let's remove the rest of the attributes (but leave the class attribute!), save file as 'bank3.arff' and try J48 again

```
children = 0
| married = NO
| | mortgage = NO: YES (48.0/3.0)
| | mortgage = YES
| | | save_act = NO: YES (12.0)
| | | save_act = YES: NO (23.0)
| | married = YES
| | | save_act = NO
| | | | mortgage = NO: NO (36.0/5.0)
| | | | mortgage = YES: YES
(25.0/3.0)
| | save_act = YES: NO (119.0/12.0)
children = 1
| income = '(-inf-14700.191667]': NO
(21.0/3.0)
| income = '(14700.191667-
24386.173333]': YES (45.0/3.0)
| income = '(24386.173333-
34072.155]': YES (33.0/2.0)
| income = '(34072.155-
```

Type I classifiers.

Decision tree: J48 on reduced dataset

- Even better accuracy. Record

The screenshot shows the Weka Explorer interface with the J48 classifier selected. The classifier output window displays the following information:

```
income = '(43736.138867-53444.118333]': YES (77.0)
| income = '(53444.118333-inf)': YES (1.0)

Number of Leaves :    24
Size of the tree :   33

Time taken to build model: 0.02 seconds

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

Correctly Classified Instances      538      89.6667 %
Incorrectly Classified Instances    62
Kappa statistic                     0.7901
Mean absolute error                  0.1766
Root mean squared error              0.3056
Relative absolute error              35.5796 %
Root relative squared error         61.3513 %
Total Number of Instances          600

=== Detailed Accuracy By Class ===
```

The accuracy value 89.6667 % is highlighted with a red box. The result list shows two entries for 'trees.J48' with timestamps 09:59:30 and 10:04:11.

Report

| Algorithm | Dataset | Accuracy |
|-----------|------------|----------|
| J48 | bank2.arff | 89.5 |
| J48 | bank3.arff | 89.7 |
| | | |
| | | |

Part I. Data Mining.

1. Prepare data
2. Build several classifiers. Choose the most accurate one.
3. Divide dataset into training and validation datasets
4. Predict class in the validation dataset
5. Prepare output for business analysis

Decision trees: Id3 and Simple cart

| Algorithm | Dataset | Accuracy, % |
|------------|------------|-------------|
| J48 | bank2.arff | 89.5 |
| J48 | bank3.arff | 89.7 |
| Id3 | bank2.arff | 77.0 |
| Id3 | bank3.arff | 86.0 |
| SimpleCart | bank2.arff | 86.8 |
| SimpleCart | bank3.arff | 89.5 |

The best accuracy for decision trees: J48 and on bank3.arff

Report so far

| Algorithm | Dataset | Accuracy, % |
|-----------|------------|-------------|
| J48 | bank3.arff | 89.7 |
| | | |
| | | |
| | | |

Part I. Data Mining.

1. Prepare data
2. Build several classifiers.
Choose the most accurate one.
3. Divide dataset into training and validation datasets
4. Predict class in the validation dataset
5. Prepare output for business analysis

Type 2 classifiers - Rules: DecisionTable on the full dataset bank2.arff

The screenshot shows the Weka Explorer interface. The classifier is set to DecisionTable with 10-fold cross-validation. The output shows a feature set of 4, 5, 6, 8, 10, and 11, and an overall accuracy of 81.5%.

Classifier output

Total number of subsets evaluated: 99
Merit of best subset found: 82.5
Evaluation (for feature selection): CV (leave one out)
Feature set: 4,5,6,8,10,11
Time taken to build model: 0.14 seconds

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

| | | |
|----------------------------------|-----------|--------|
| Correctly Classified Instances | 489 | 81.5 % |
| Incorrectly Classified Instances | 111 | 18.5 % |
| Kappa statistic | 0.6238 | |
| Mean absolute error | 0.3148 | |
| Root mean squared error | 0.3862 | |
| Relative absolute error | 63.4235 % | |
| Root relative squared error | 77.5316 % | |
| Total Number of Instances | 600 | |

=== Detailed Accuracy By Class ===

| | TP Rate | FP Rate | Precision | Recall | F-Measure | ROC Area |
|--|---------|---------|-----------|--------|-----------|----------|
| | 0.741 | 0.123 | 0.835 | 0.741 | 0.785 | 0.85 |
| | 0.877 | 0.259 | 0.801 | 0.877 | 0.837 | 0.85 |

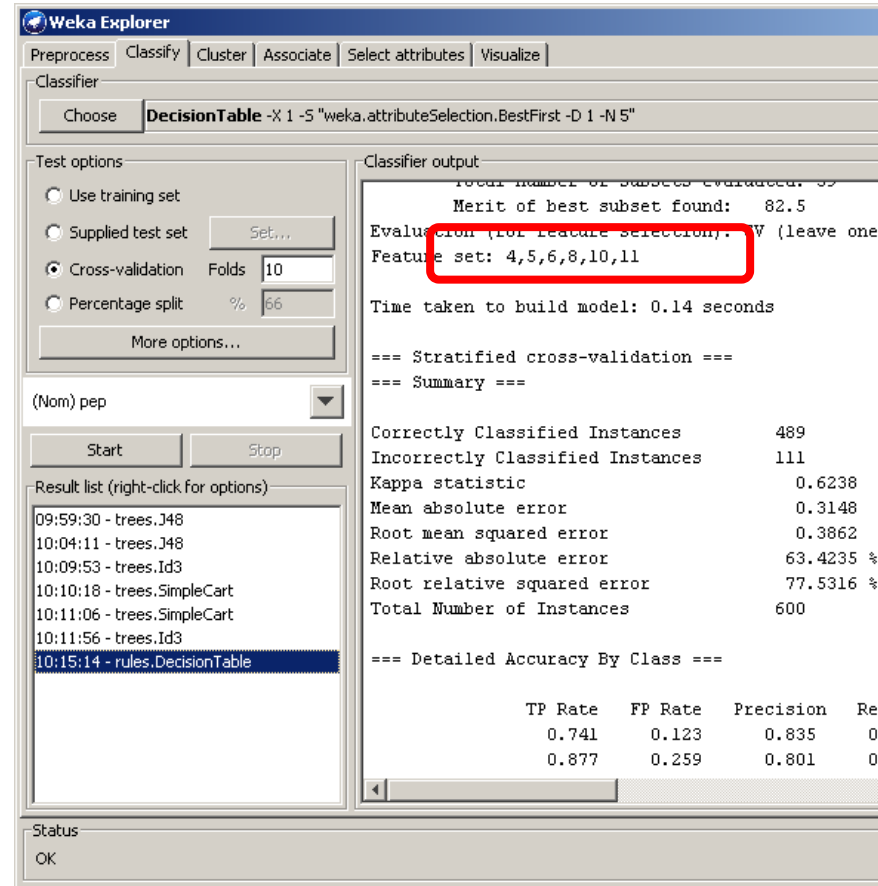
Result list (right-click for options)

- 09:59:30 - trees.J48
- 10:04:11 - trees.J48
- 10:09:53 - trees.Id3
- 10:10:18 - trees.SimpleCart
- 10:11:06 - trees.SimpleCart
- 10:11:56 - trees.Id3
- 10:15:14 - rules.DecisionTable

The most important
attributes for
classification

Attribute selection: DecisionTable on the full dataset

- The most important attributes:
 - 4- income
 - 5- married
 - 6- children
 - 8- save_act
 - 10 - mortgage
- Let's remove the rest
- Save file as bank4.arff
- Re-build decision tree J48:
accuracy **89.7 – very high!**
- We will use bank4.arff as our
input for the rest of the lab



Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Classifier: DecisionTable -X 1 -S "weka.attributeSelection.BestFirst -D 1 -N 5"

Test options:

- Use training set
- Supplied test set (Set...)
- Cross-validation (Folds: 10)
- Percentage split (%: 66)

More options...

(Nom) pep

Start Stop

Result list (right-click for options):

- 09:59:30 - trees.J48
- 10:04:11 - trees.J48
- 10:09:53 - trees.Id3
- 10:10:18 - trees.SimpleCart
- 10:11:06 - trees.SimpleCart
- 10:11:56 - trees.Id3
- 10:15:14 - rules.DecisionTable**

Classifier output:

Total number of subsets evaluated: 99
Merit of best subset found: 82.5
Evaluation (for feature selection): W (leave one out)
Feature set: 4,5,6,8,10,11
Time taken to build model: 0.14 seconds

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

| | |
|----------------------------------|-----------|
| Correctly Classified Instances | 489 |
| Incorrectly Classified Instances | 111 |
| Kappa statistic | 0.6238 |
| Mean absolute error | 0.3148 |
| Root mean squared error | 0.3862 |
| Relative absolute error | 63.4235 % |
| Root relative squared error | 77.5316 % |
| Total Number of Instances | 600 |

=== Detailed Accuracy By Class ===

| | TP Rate | FP Rate | Precision | Re |
|--|---------|---------|-----------|----|
| | 0.741 | 0.123 | 0.835 | 0 |
| | 0.877 | 0.259 | 0.801 | 0 |

Status: OK

The rest of the Rule learners on bank4.arff

| Algorithm | Dataset | Accuracy, % |
|-----------|------------|-------------|
| J48 | bank3.arff | 89.7 |
| J48 | bank4.arff | 89.7 |
| JRip | bank4.arff | 87.8 |
| Part | bank4.arff | 88.3 |
| Prism | bank4.arff | 67.3 |
| Ridor | bank4.arff | 84.7 |

Rules

The best result
for rule learners

Part I. Data Mining.

1. Prepare data
2. Build several classifiers. Choose the most accurate one.
3. Divide dataset into training and validation datasets
4. Predict class in the validation dataset
5. Prepare output for business analysis

Report so far

| Algorithm | Dataset | Accuracy, % |
|-----------|------------|-------------|
| J48 | bank4.arff | 89.7 |
| Part | bank4.arff | 88.3 |
| | | |
| | | |

Type III classifiers: Naïve Bayes

- For 'bank2.arff' (full dataset):
70.5% accurate
- For 'bank3.arff' (J48 reduction):
72.5% accurate
- For 'bank4.arff' (DecisionTable reduction): 72.5% accurate

Part I. Data Mining.

1. Prepare data
2. Build several classifiers.
Choose the most accurate one.
3. Divide dataset into training and validation datasets
4. Predict class in the validation dataset
5. Prepare output for business analysis

Report

| Algorithm | Dataset | Accuracy, % |
|------------|------------|-------------|
| J48 | bank4.arff | 89.7 |
| Part | bank4.arff | 88.3 |
| NaiveBayes | bank4.arff | 72.5 |
| | | |


Part I. Data Mining.

1. Prepare data
2. Build several classifiers.
Choose the most accurate one.
3. Divide dataset into training and validation datasets
4. Predict class in the validation dataset
5. Prepare output for business analysis

Generating validation dataset

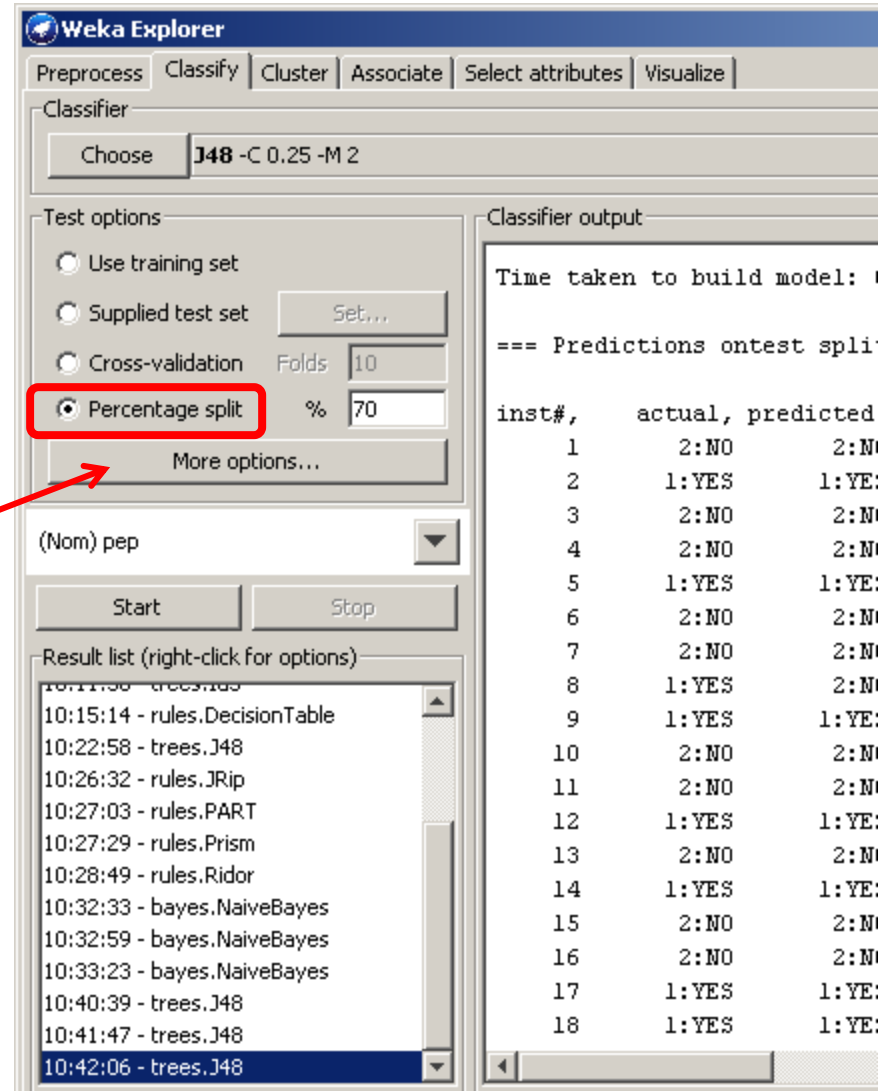
- We will use 70% of the data for training the classifier, and 30% for the validation
- The validation dataset contains actual responses, but we will try to predict them with our best classifier, to see how good is the prediction

Part I. Data Mining.

1. Prepare data
2. Build several classifiers.
Choose the most accurate one.
-  3. Divide dataset into training and validation datasets
4. Predict class in the validation dataset
5. Prepare output for business analysis

Generating output for business analysis

- Re-open bank4.arff
- Choose one of our best classifiers: J48
- Test options: Percentage split
- Press More Options button



The screenshot shows the Weka Explorer interface. The 'Classifier' tab is active, and the 'J48 -C 0.25 -M 2' classifier is selected. In the 'Test options' section, the 'Percentage split' radio button is selected and highlighted with a red box. A red arrow points from the 'More options...' button in the 'Test options' section to the 'Percentage split' radio button. The 'Classifier output' section shows the time taken to build the model and a table of predictions on the test split.

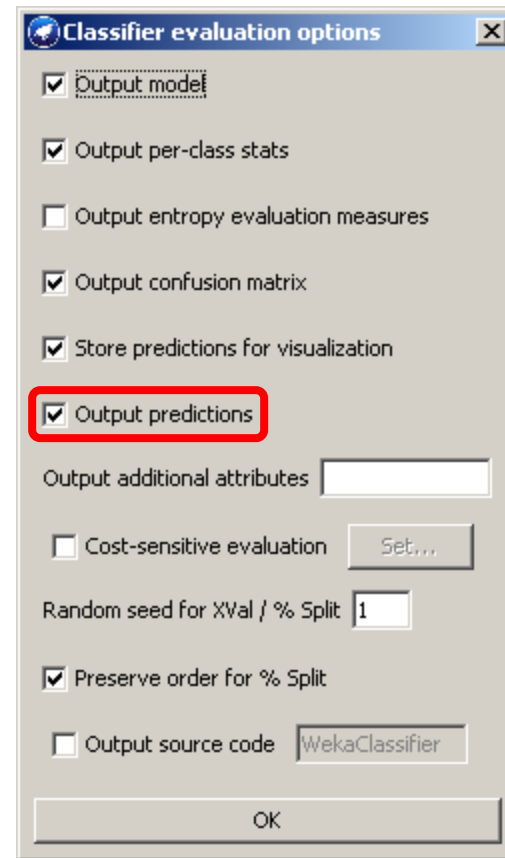
Time taken to build model: ...

=== Predictions on test split

| inst# | actual | predicted |
|-------|--------|-----------|
| 1 | 2:NO | 2:NO |
| 2 | 1:YES | 1:YES |
| 3 | 2:NO | 2:NO |
| 4 | 2:NO | 2:NO |
| 5 | 1:YES | 1:YES |
| 6 | 2:NO | 2:NO |
| 7 | 2:NO | 2:NO |
| 8 | 1:YES | 2:NO |
| 9 | 1:YES | 1:YES |
| 10 | 2:NO | 2:NO |
| 11 | 2:NO | 2:NO |
| 12 | 1:YES | 1:YES |
| 13 | 2:NO | 2:NO |
| 14 | 1:YES | 1:YES |
| 15 | 2:NO | 2:NO |
| 16 | 2:NO | 2:NO |
| 17 | 1:YES | 1:YES |
| 18 | 1:YES | 1:YES |

Generating output for business analysis

- Check: Output predictions
- Run J48 Decision tree classifier



Predict class in the validation dataset

- Run J48 using training and validation datasets. Note that the accuracy has decreased.

Part I. Data Mining.

1. Prepare data
2. Build several classifiers.
Choose the most accurate one.
3. Divide dataset into training and validation datasets
4. Predict class in the validation dataset
5. Prepare output for business analysis



Transfer prediction into a text file

- Copy predictions and paste into a text file
- Save file as bank_predicted.txt

Do find *,+ and replace them with a space

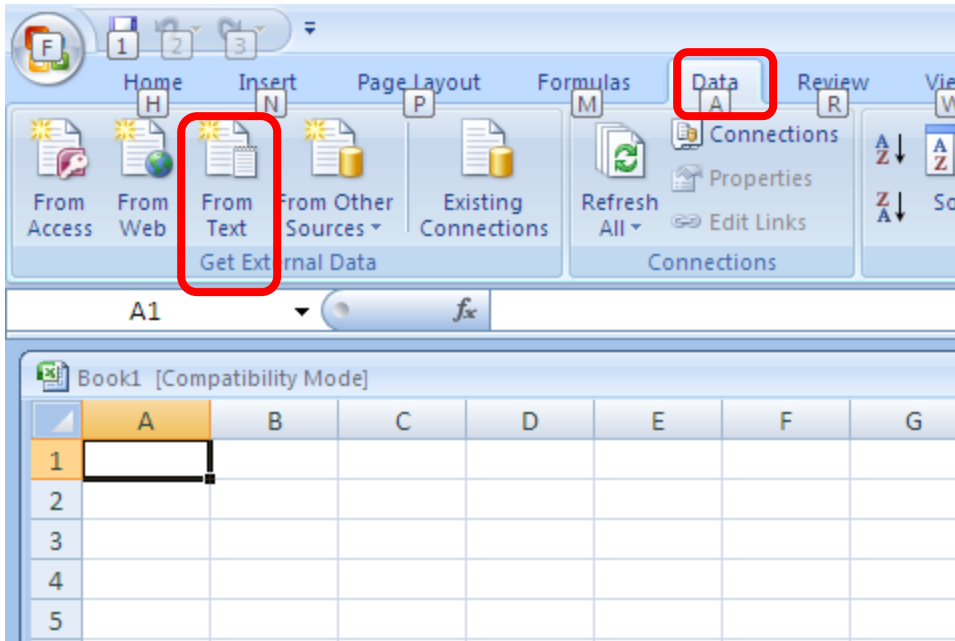
The screenshot shows the Weka Explorer interface. The 'Classifier' tab is active, displaying 'J48 -C 0.25 -M 2'. The 'Test options' section shows 'Percentage split' set to 70%. The 'Classifier output' window displays a table of results for instances 163 to 180. The 'Result list' shows various classifiers, with 'trees.J48' selected. A 'Replace' dialog box is open in the foreground, with 'Find what' set to '*' and 'Replace with' set to a space character. The 'Match case' checkbox is unchecked.

| Instance | Actual | Predicted | Confidence | Probability |
|----------|--------|-----------|------------|-------------|
| 163 | 2:NO | 2:NO | 0.077 | *0.923 |
| 164 | 2:NO | 2:NO | 0.263 | *0.737 |
| 165 | 1:YES | 1:YES | *0.923 | 0.077 |
| 166 | 2:NO | 2:NO | 0.111 | *0.889 |
| 167 | 1:YES | 1:YES | *0.923 | 0.077 |
| 168 | 1:YES | 1:YES | *0.917 | 0.083 |
| 169 | 2:NO | 2:NO | 0.076 | *0.924 |
| 170 | 2:NO | 2:NO | 0.214 | *0.786 |
| 171 | 2:NO | 2:NO | 0.111 | *0.889 |
| 172 | 1:YES | 2:NO | 0.263 | *0.737 |
| 173 | 2:NO | 2:NO | 0.076 | *0.924 |
| 174 | 1:YES | 1:YES | *1 | 0 |
| 175 | 2:NO | 2:NO | 0.1 | *0.9 |
| 176 | 2:NO | 1:YES | + *1 | 0 |
| 177 | 2:NO | 2:NO | 0.076 | *0.924 |
| 178 | 1:YES | 2:NO | + 0.076 | *0.924 |
| 179 | 2:NO | 2:NO | 0.076 | *0.924 |
| 180 | 1:YES | 1:YES | *1 | 0 |

=== Evaluation on test split ===
=== Summary ===
Correctly Classified Instances 158
Incorrectly Classified Instances 22

Import predictions into Electronic tables program: example - Excel

- Import data from bank_predicted.txt



Part I. Data Mining.

1. Prepare data
2. Build several classifiers.
Choose the most accurate one.
3. Divide dataset into training and validation datasets
4. Predict class in the validation dataset
5. Prepare output for business analysis

Import predictions into Electronic tables program: example - Excel

- Import data from bank_predicted.txt

Text Import Wizard - Step 1 of 3

The Text Wizard has determined that your data is Fixed Width.
If this is correct, choose Next, or choose the data type that best describes your data.

Original data type

Choose the file type that best describes your data:

Delimited - Characters such as commas or tabs separate each field.

Fixed width - Fields are aligned in columns with spaces between each field.

Start import at row: 1 File origin: 437 : OEM United States

Preview of file C:\Documents and Settings\barskym\My Documents\DMCourse\...\bank_predicted.txt.

| | inst# | actual | predicted | error | probability | distribution |
|---|-------|--------|-----------|-------|-------------|--------------|
| 2 | 1 | 2:NO | 2:NO | | 0.076 | 0.924 |
| 3 | 2 | 1:YES | 1:YES | | 0.938 | 0.063 |
| 4 | 3 | 2:NO | 2:NO | | 0.214 | 0.786 |
| 5 | 4 | 2:NO | 2:NO | | 0.076 | 0.924 |

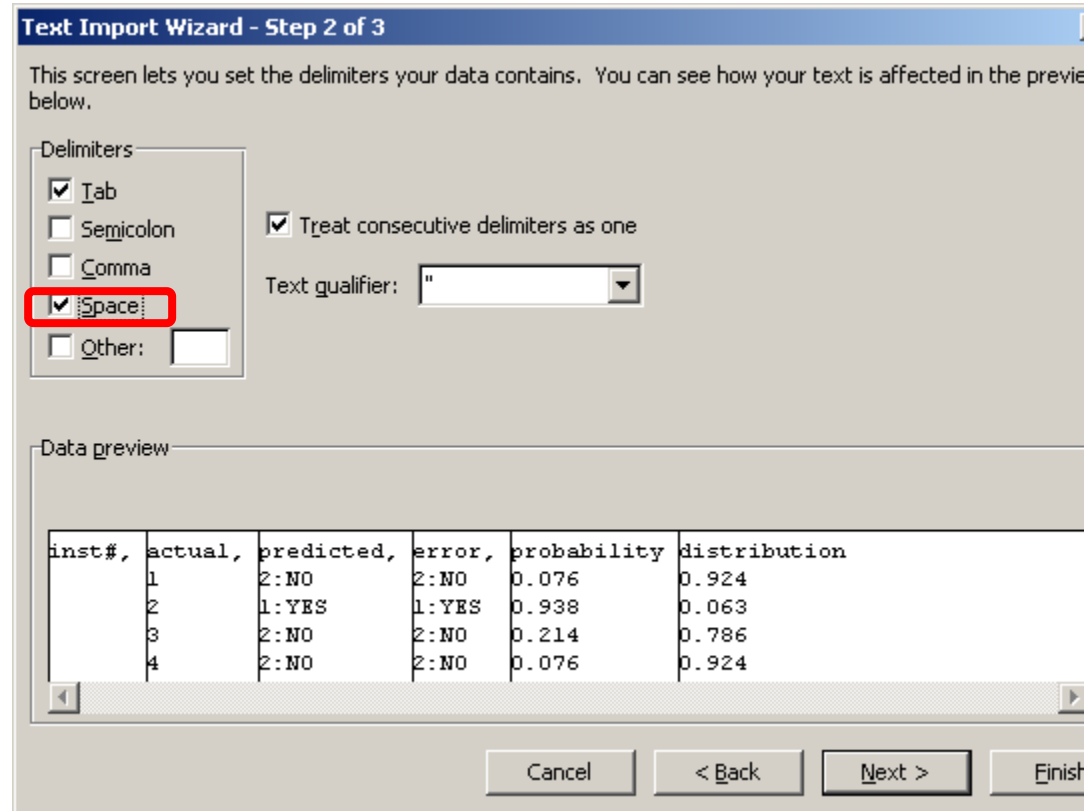
Buttons: Cancel, < Back, Next >, Finish

Part I. Data Mining.

1. Prepare data
2. Build several classifiers.
Choose the most accurate one.
3. Divide dataset into training and validation datasets
4. Predict class in the validation dataset
5. Prepare output for business analysis

Import predictions into Electronic tables program: example - Excel


- Import data from bank_predicted.txt
- Save file as bank_results.xls (sample file is attached)



Close WEKA

- The data mining part is complete

Part I. Data Mining.

 Part II. Business analysis

1. Generate Lift chart(s)
2. Cost-benefit analysis
3. Recommendations