

คู่มือการใช้งาน WEKA

คณะวิศวกรรมศาสตร์
มหาวิทยาลัยรามคำแหง

การทำเหมืองข้อมูลด้วย Weka (Data Mining with Weka)

การติดตั้ง Weka

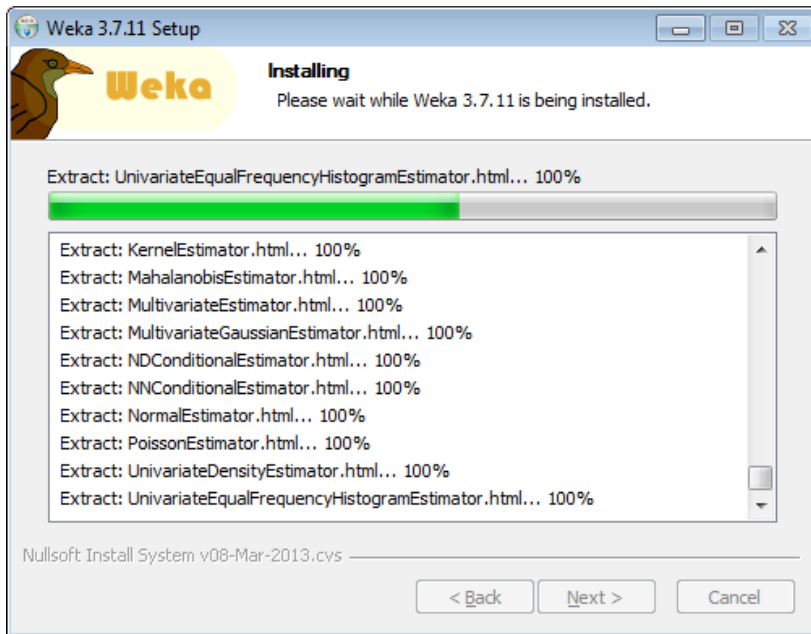
1. Download โปรแกรมที่ <http://www.cs.waikato.ac.nz/ml/weka/downloading.html>
2. ดับเบิลคลิกเพื่อติดตั้ง

ในที่นี้จะใช้ version 3.7.11 (Book version 3.6)

ติดตั้งตาม default และ Next



จนถึง

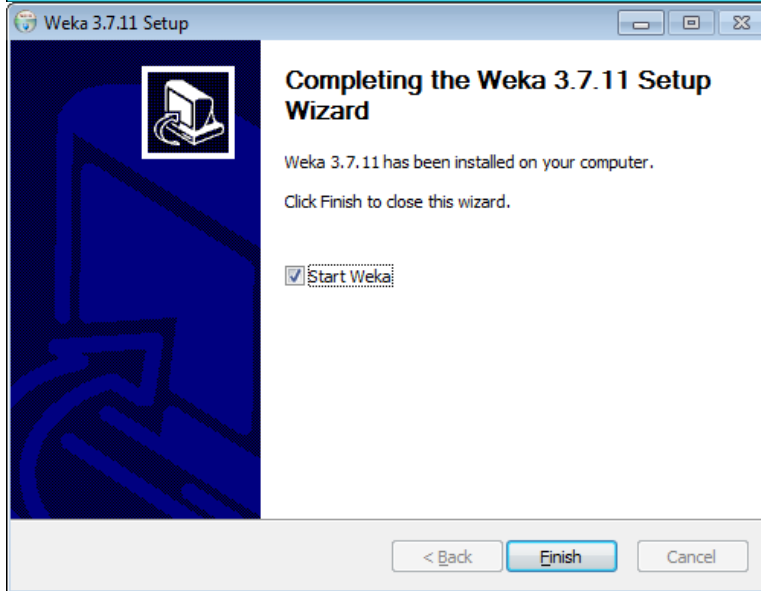
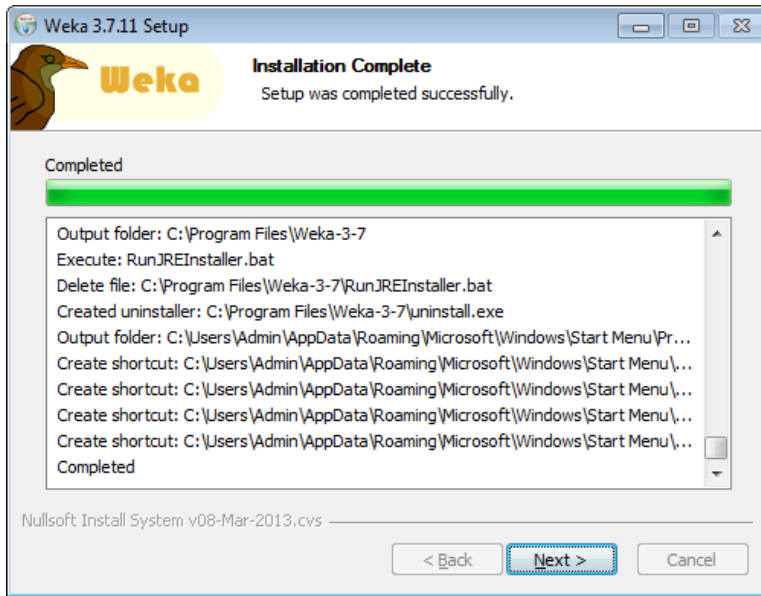


3. Install JRE



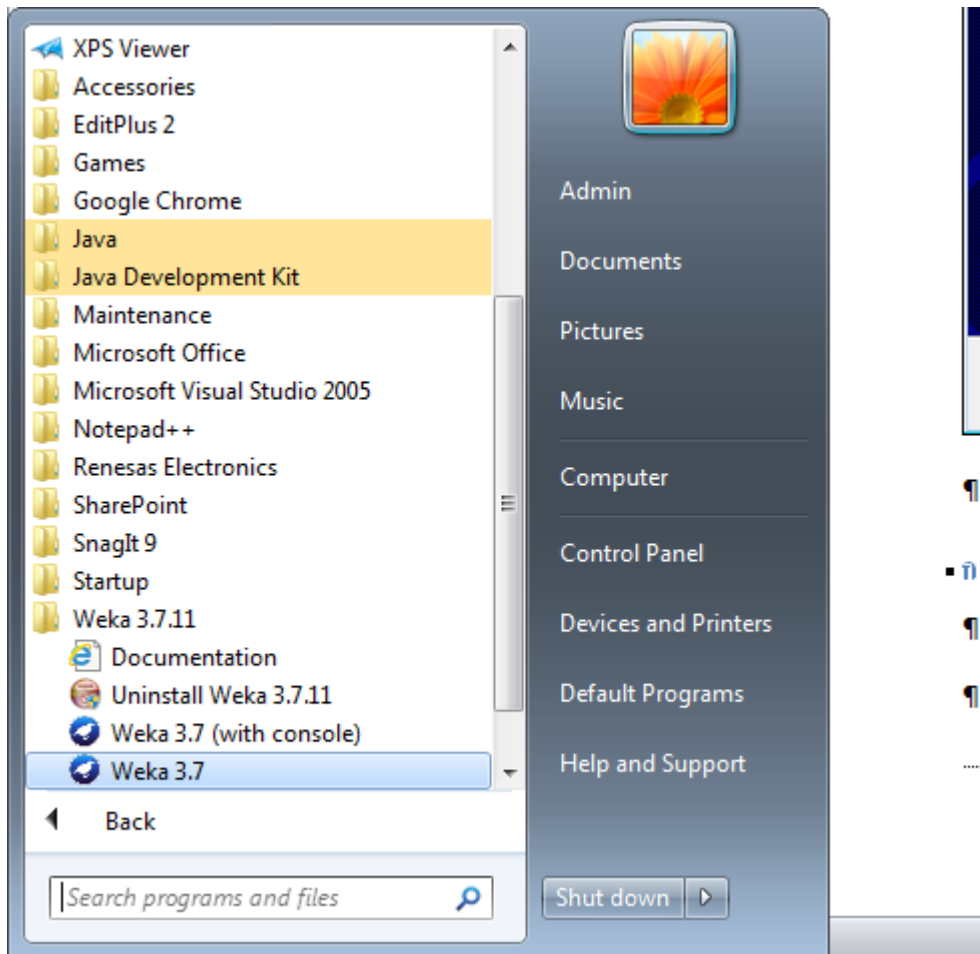
กดปุ่ม Install

4. ติดตั้งเรียบร้อยแล้ว

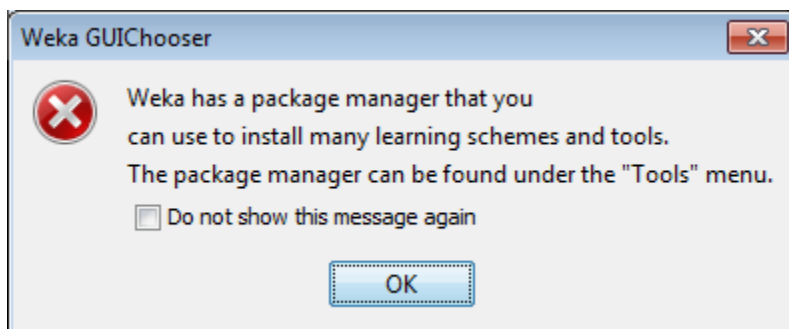


การเรียกใช้ Weka

1. All Programs > Weka 3.7



หรือ สำเนา short cut Weka 3.7 มาไว้บน desktop เพื่อสะดวกในการเรียกใช้
ดับเบิลคลิก Weka ไอคอน มาทำงาน จะเกิด



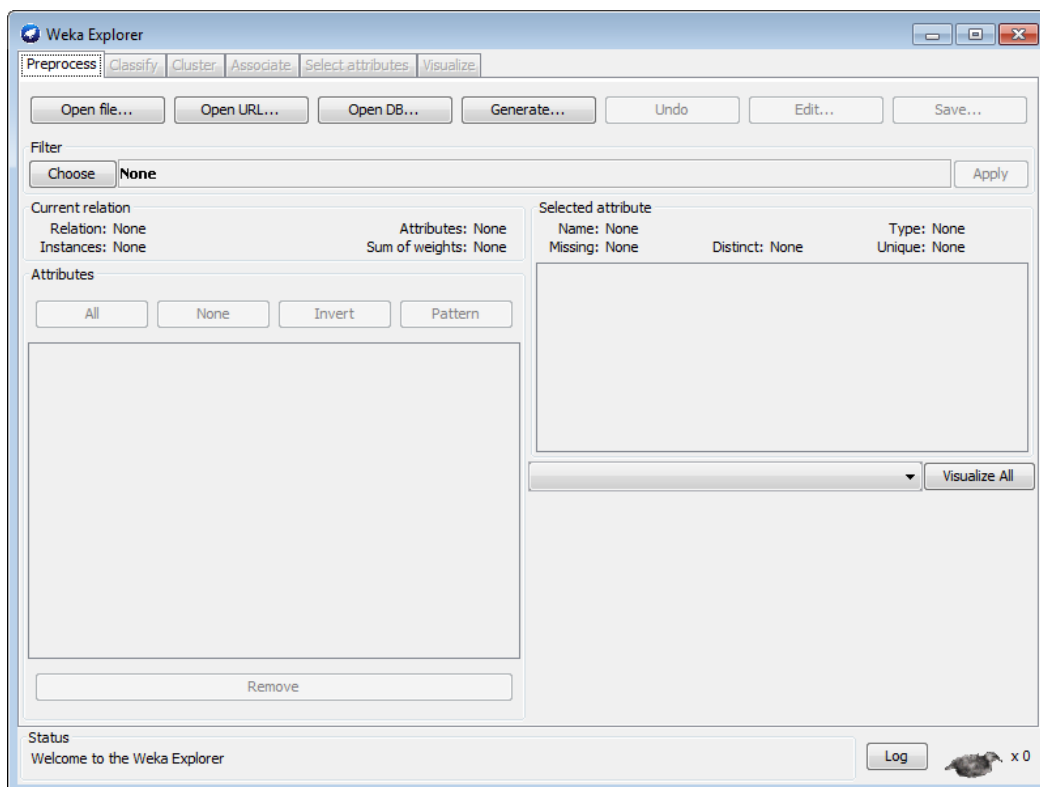
คลิก Do not show this message again แล้วคลิก OK

Weka GUI Chooser

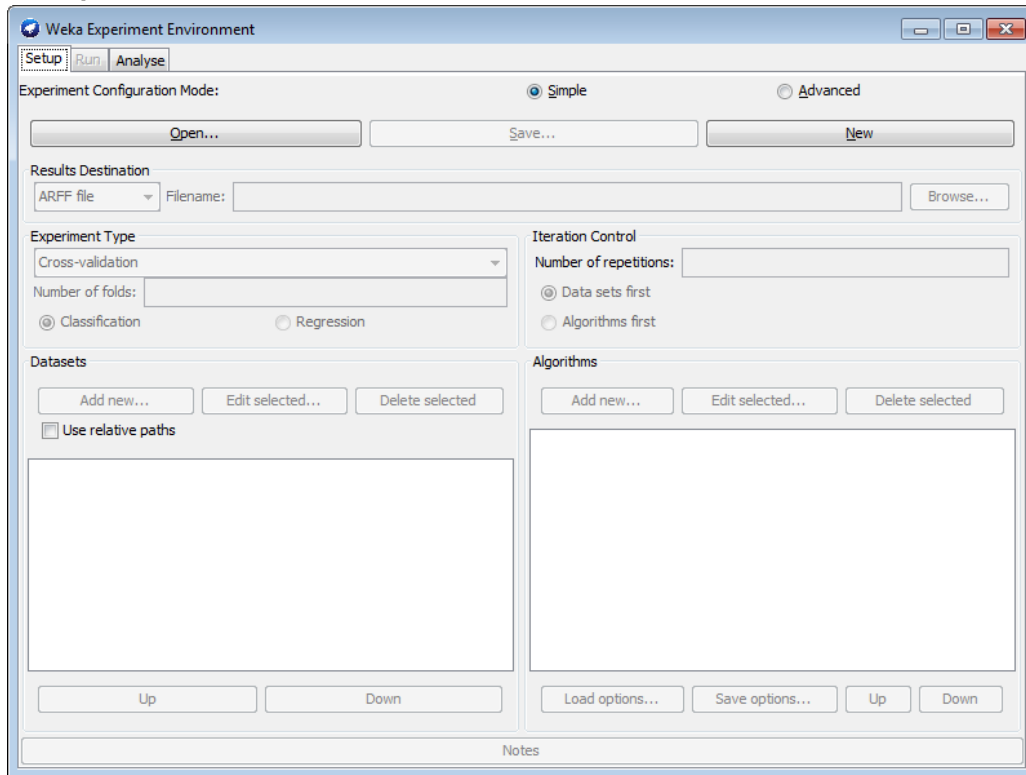


มี Interface การทำงาน 4 แบบ

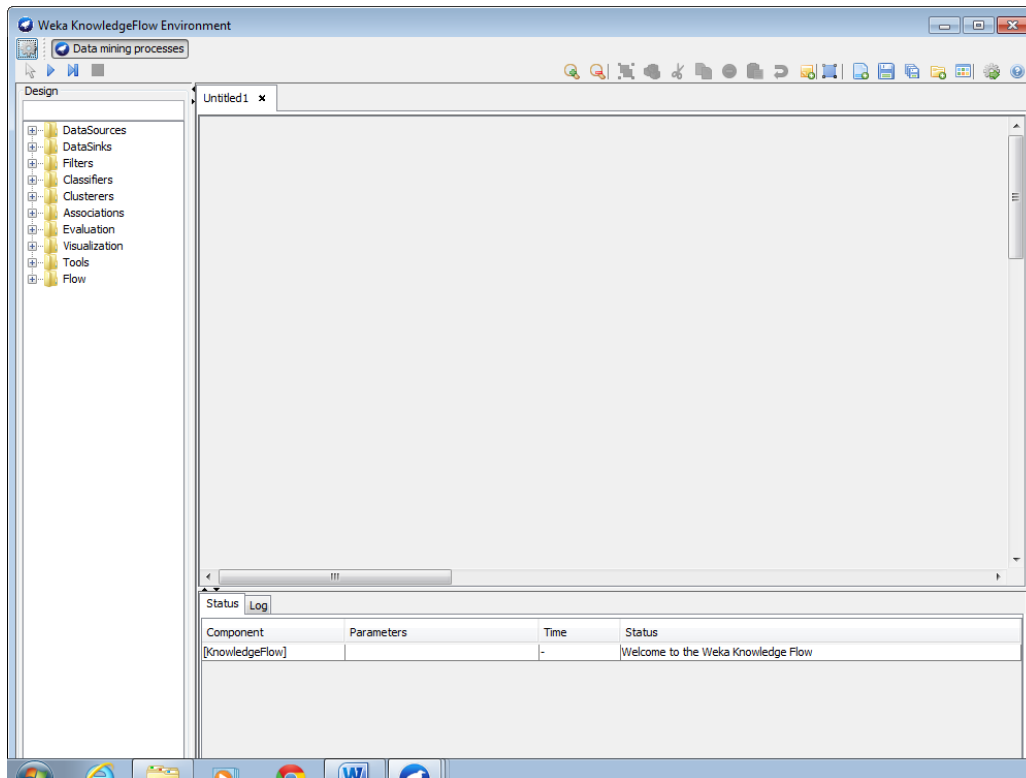
1. Explorer



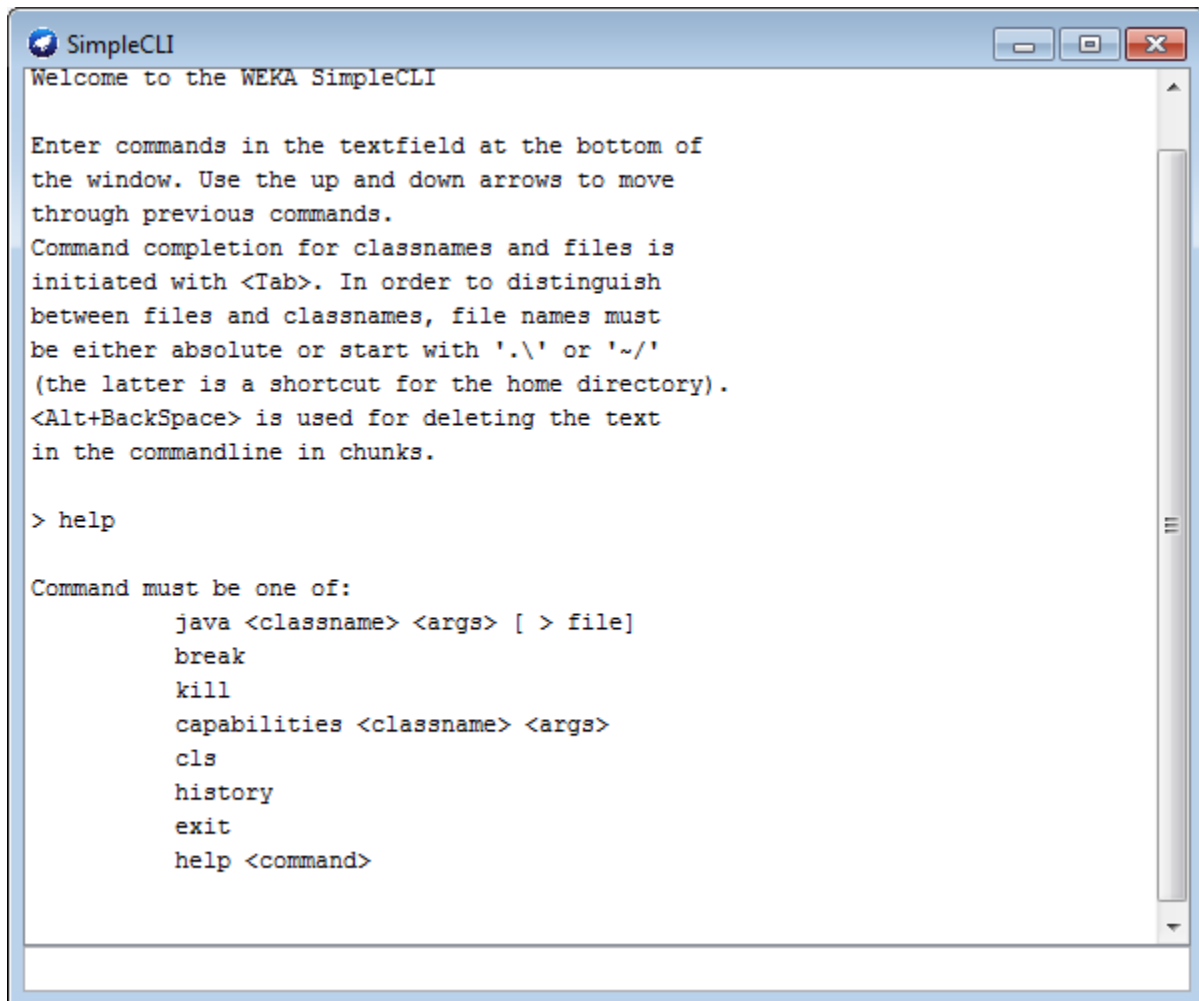
2. Experimenter



3. KnowledgeFlow



4. Simple CLI



The image shows a window titled "SimpleCLI" with standard Windows window controls (minimize, maximize, close). The window contains the following text:

```
Welcome to the WEKA SimpleCLI

Enter commands in the textfield at the bottom of
the window. Use the up and down arrows to move
through previous commands.
Command completion for classnames and files is
initiated with <Tab>. In order to distinguish
between files and classnames, file names must
be either absolute or start with './' or '~/ '
(the latter is a shortcut for the home directory).
<Alt+BackSpace> is used for deleting the text
in the commandline in chunks.

> help

Command must be one of:
    java <classname> <args> [ > file]
    break
    kill
    capabilities <classname> <args>
    cls
    history
    exit
    help <command>
```

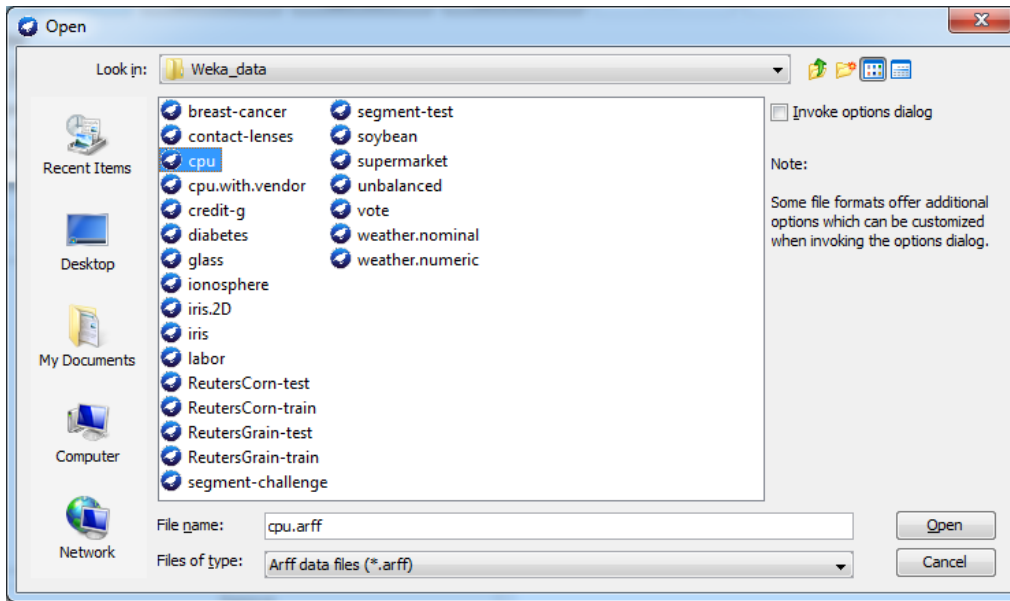

Weka Workshops

WS#1: Numeric Prediction

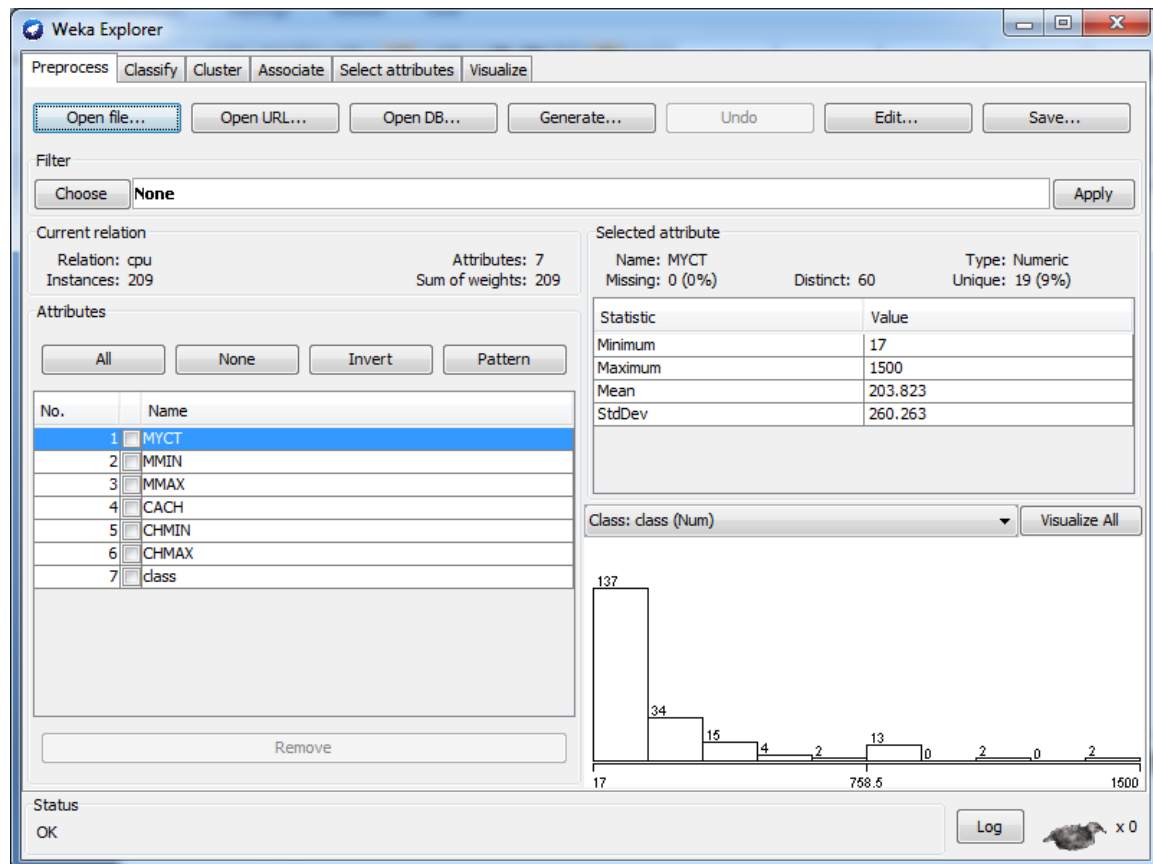
Data set: cpu (Weka_data จะมีไฟล์ format เป็น .arff)

```
1 %  
2 % As used by Kilpatrick, D. & Cameron-Jones, M. (1998). Numeric prediction  
3 % using instance-based learning with encoding length selection. In Progress  
4 % in Connectionist-Based Information Systems. Singapore: Springer-Verlag.  
5 %  
6 % Deleted "vendor" attribute to make data consistent with with what we  
7 % used in the data mining book.  
8 %  
9 @relation 'cpu'  
10 @attribute MYCT numeric  
11 @attribute MMIN numeric  
12 @attribute MMAX numeric  
13 @attribute CACH numeric  
14 @attribute CHMIN numeric  
15 @attribute CHMAX numeric  
16 @attribute class numeric  
17 @data  
18 125,256,6000,256,16,128,198  
19 29,8000,32000,32,8,32,269  
20 29,8000,32000,32,8,32,220  
21 29,8000,32000,32,8,32,172  
22 29,8000,16000,32,8,16,132  
23 26,8000,32000,64,8,32,318
```

1. Preprocess > Open file เลือก cpu



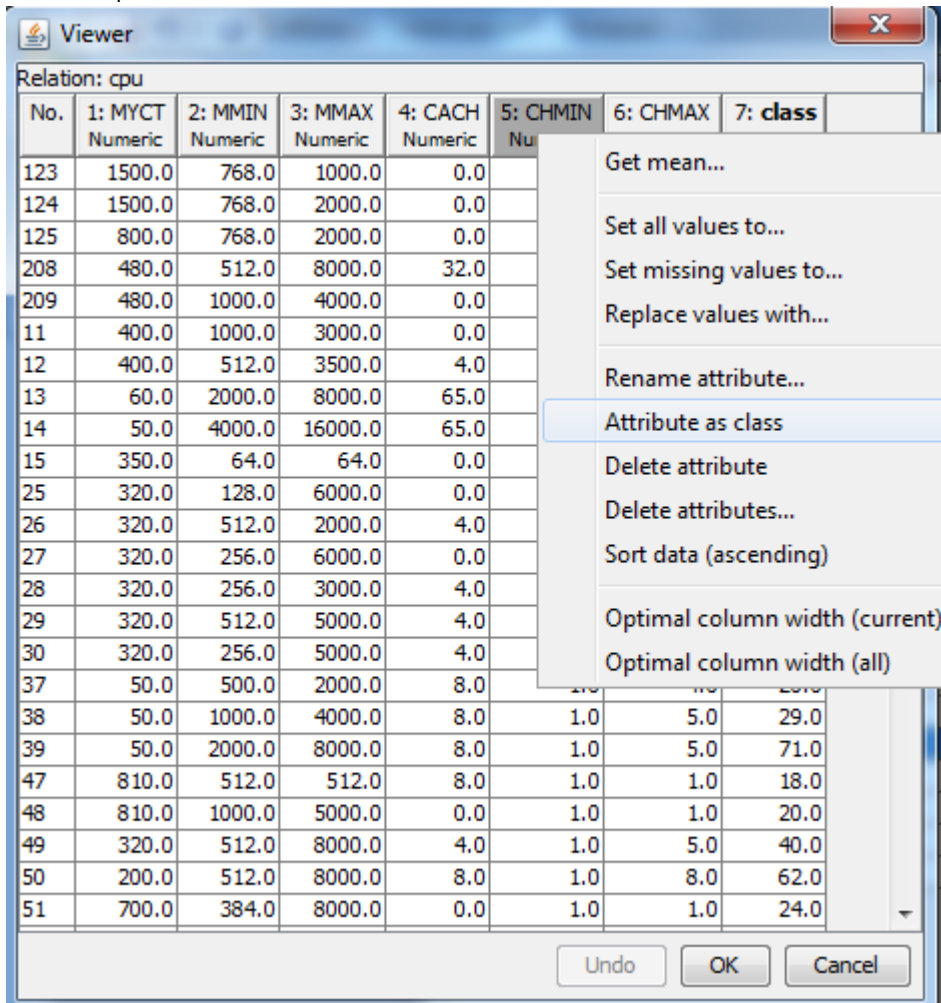
ได้



2. แอททริบิวต์ตัวสุดท้ายจะเป็นตัวแปร class หากต้องการให้ตัวแปรอื่นเป็นตัวแปร class ให้ Edit แล้วคลิกขวาตัวแปรที่ต้องการ เลือก Attribute as class ตัวแปรนั้นก็จะมาอยู่เป็นตัวสุดท้าย และแสดงเป็น

ตัวหนา

หมายเหตุ จะใช้ว่า แอททริบิวต์ หรือ ตัวแปร ก็ได้



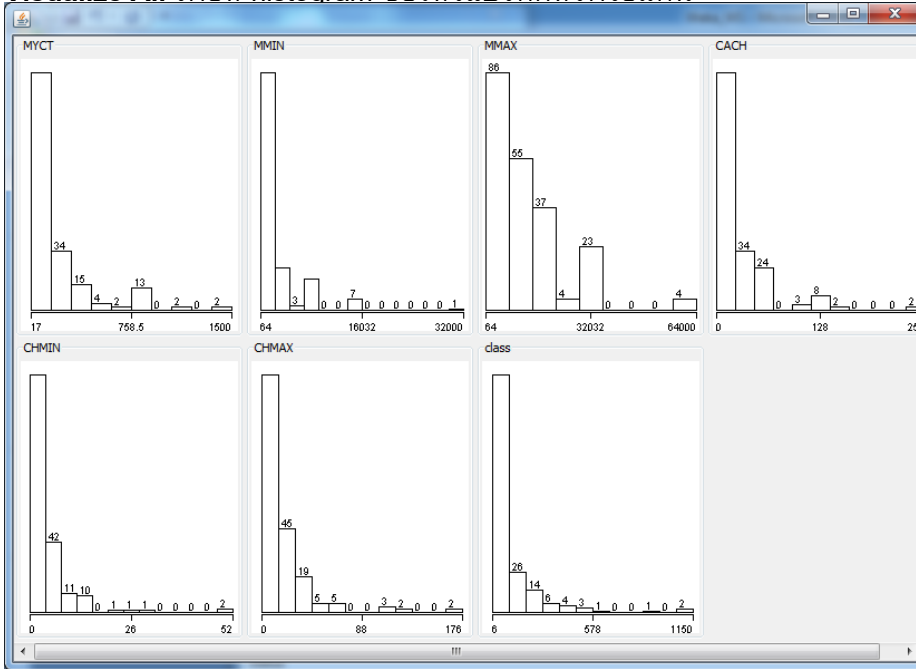
The screenshot shows a 'Viewer' window with a table of data. The table has 7 columns: 'No.', '1: MYCT', '2: MMIN', '3: MMAX', '4: CACH', '5: CHMIN', '6: CHMAX', and '7: class'. The 'class' column is highlighted in bold. A context menu is open over the 'class' column, with the option 'Attribute as class' selected. The menu also includes options like 'Get mean...', 'Set all values to...', 'Set missing values to...', 'Replace values with...', 'Rename attribute...', 'Delete attribute', 'Delete attributes...', 'Sort data (ascending)', 'Optimal column width (current)', and 'Optimal column width (all)'. The table data is as follows:

No.	1: MYCT Numeric	2: MMIN Numeric	3: MMAX Numeric	4: CACH Numeric	5: CHMIN Nu	6: CHMAX	7: class
123	1500.0	768.0	1000.0	0.0			
124	1500.0	768.0	2000.0	0.0			
125	800.0	768.0	2000.0	0.0			
208	480.0	512.0	8000.0	32.0			
209	480.0	1000.0	4000.0	0.0			
11	400.0	1000.0	3000.0	0.0			
12	400.0	512.0	3500.0	4.0			
13	60.0	2000.0	8000.0	65.0			
14	50.0	4000.0	16000.0	65.0			
15	350.0	64.0	64.0	0.0			
25	320.0	128.0	6000.0	0.0			
26	320.0	512.0	2000.0	4.0			
27	320.0	256.0	6000.0	0.0			
28	320.0	256.0	3000.0	4.0			
29	320.0	512.0	5000.0	4.0			
30	320.0	256.0	5000.0	4.0			
37	50.0	500.0	2000.0	8.0			
38	50.0	1000.0	4000.0	8.0	1.0	5.0	29.0
39	50.0	2000.0	8000.0	8.0	1.0	5.0	71.0
47	810.0	512.0	512.0	8.0	1.0	1.0	18.0
48	810.0	1000.0	5000.0	0.0	1.0	1.0	20.0
49	320.0	512.0	8000.0	4.0	1.0	5.0	40.0
50	200.0	512.0	8000.0	8.0	1.0	8.0	62.0
51	700.0	384.0	8000.0	0.0	1.0	1.0	24.0

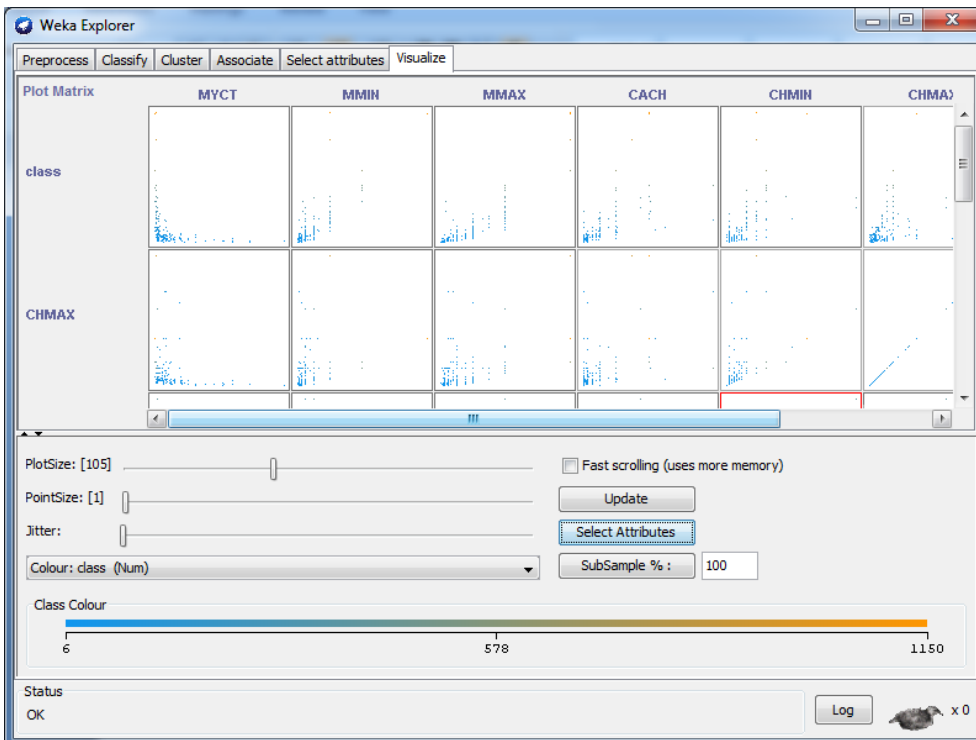
ถ้าต้องการได้ไฟล์ลง Excel ก็ ctrl+A (Select All) แล้วไป paste ลงใน Excel แล้วเพิ่มบรรทัดแรกเป็นชื่อตัวแปร

3. Weka จะสรุปค่า descriptive statistics ของแต่ละตัวแปรให้ พร้อมแสดงกราฟ (histogram) หรือคลิก

Visualize All เพื่อดู histogram ของตัวแปรทุกตัวพร้อมกัน



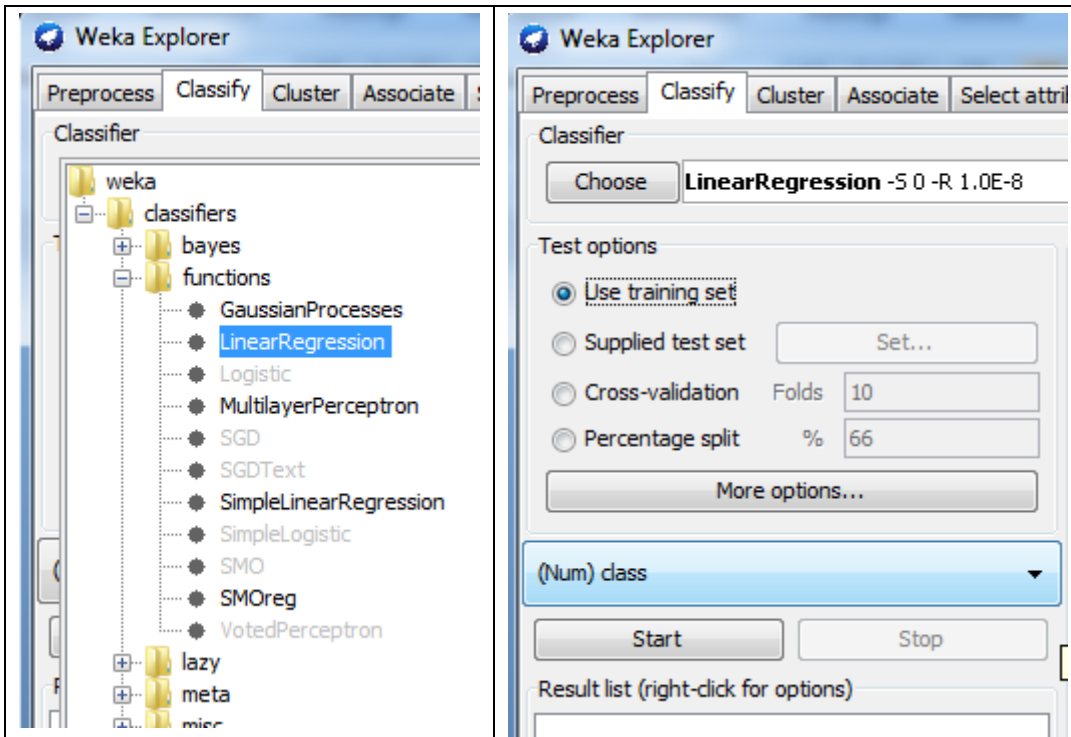
4. Visualize scatter plot (ตัวแปร 2 ตัว) โดยคลิกที่แท็บ **Visualize**



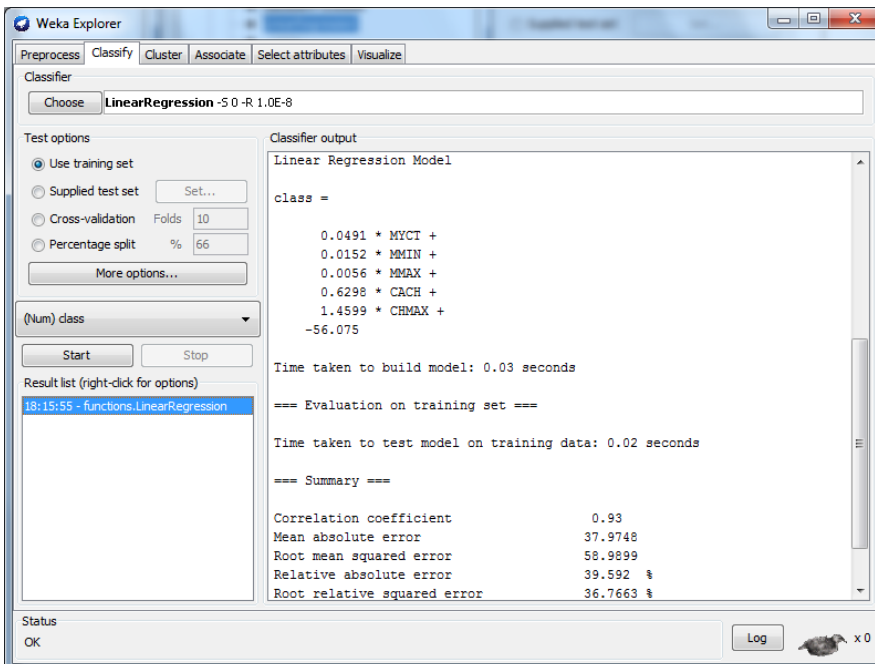
5. Fit linear regression model

Classify > Choose Classifier = functions > Linear Regression

Test options = Use training set (default)



กด Start จะได้ผลลัพธ์



6. การแปลผล

6.1 Model: $\text{Class/PRP} = -56.075 + 0.0491 \cdot \text{MYCT} + 0.0152 \cdot \text{MMIN} + 0.0056 \cdot \text{MMAX} + 0.6298 \cdot \text{CACH} + 1.4599 \cdot \text{CHMAX}$

ตัวแปรใดที่ไม่เข้าในโมเดล?

```
Linear Regression Model

class =

    0.0491 * MYCT +
    0.0152 * MMIN +
    0.0056 * MMAX +
    0.6298 * CACH +
    1.4599 * CHMAX +
    -56.075
```

6.2 Evaluation: Correlation coefficient $R=0.93$ or $R^2=0.8649$

ตัวแปร predictor 5 ตัว คือ MYCT, MMIN, MMAX, CACH, CHMAX สามารถอธิบายความแปรปรวนของตัวแปร Class/PRP ได้ประมาณ 81%

Correlation coefficient	0.93
-------------------------	------

6.3 Deployment: เครื่องคอมพิวเตอร์เครื่องใหม่ 1 เครื่อง มีคุณสมบัติดังนี้ MYCT=200, MMIN=1000, MMAX=2000, CASH=0, CHMAX=64 จะมีประสิทธิภาพ (PRP) = 73.58

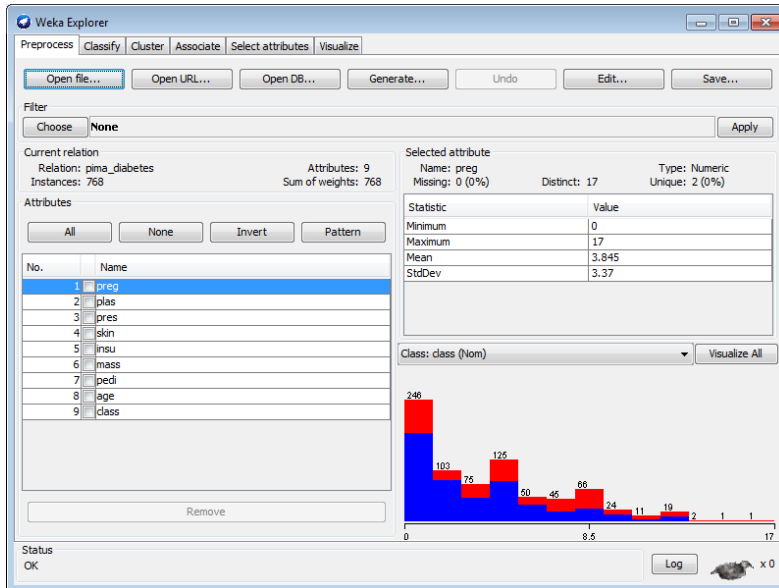
คลิกขวาที่ Result แล้ว Save model

7. ให้ทดสอบทั้ง 3 test options จะได้ว่า Weka ใช้ full training set ในการสร้างโมเดล ซึ่งผลลัพธ์ของโมเดลจะเหมือนกัน ซึ่งสอดคล้องกับ Stepwise Regression ใน SPSS และใน R แต่ผลลัพธ์ของ Evaluation จาก Correlation coefficient ของทั้ง 3 test mode คือ 1) Use training set, 2) Cross-validation (10 folds), 3) Percentage split (66%) จะไม่เท่ากัน

WS#2: Classification by Logistic Regression

Data set: diabetes

1. Preprocess > Open file เลือก diabetes



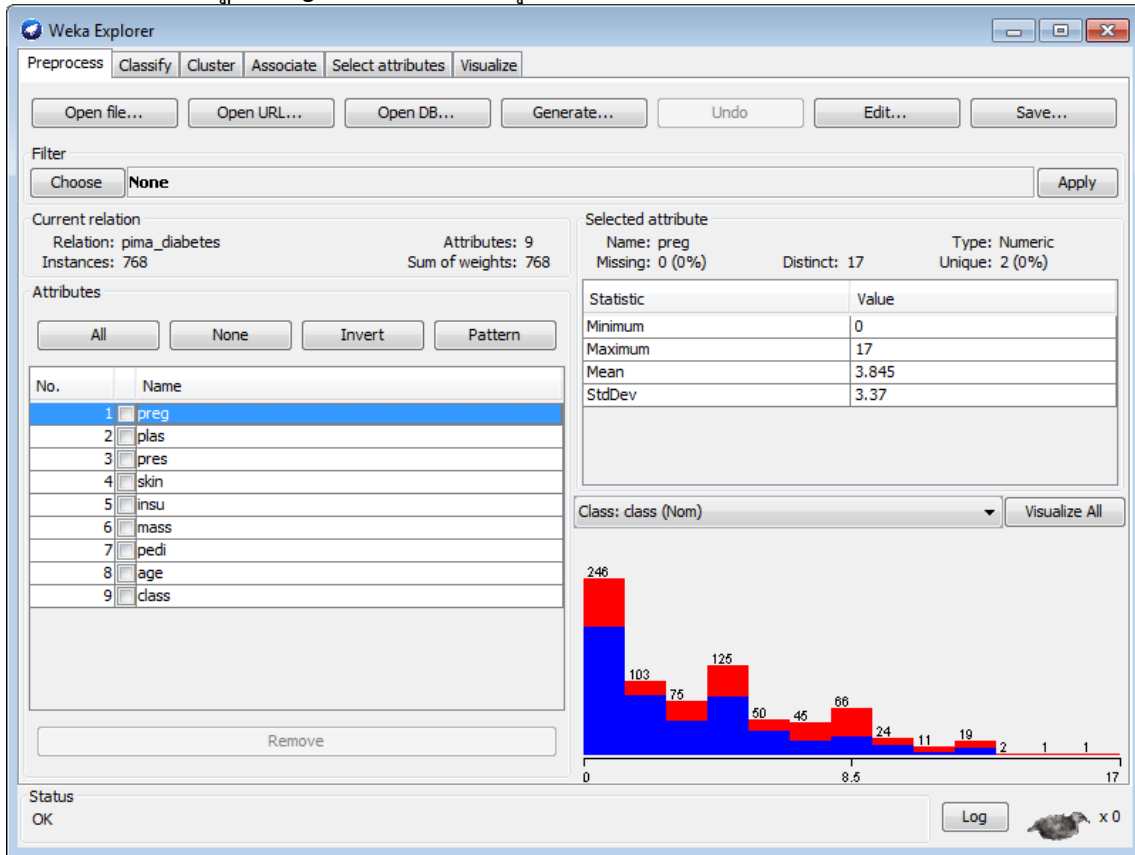
2. Edit เพื่อดู type และ ค่าของแต่ละตัวแปร

No.	1: preg Numeric	2: plas Numeric	3: pres Numeric	4: skin Numeric	5: insu Numeric	6: mass Numeric	7: pedi Numeric	8: age Numeric	9: class Nominal
1	6.0	148.0	72.0	35.0	0.0	33.6	0.627	50.0	tested_positive
2	1.0	85.0	66.0	29.0	0.0	26.6	0.351	31.0	tested_negative
3	8.0	183.0	64.0	0.0	0.0	23.3	0.672	32.0	tested_positive
4	1.0	89.0	66.0	23.0	94.0	28.1	0.167	21.0	tested_negative
5	0.0	137.0	40.0	35.0	168.0	43.1	2.288	33.0	tested_positive
6	5.0	116.0	74.0	0.0	0.0	25.6	0.201	30.0	tested_negative
7	3.0	78.0	50.0	32.0	88.0	31.0	0.248	26.0	tested_positive
8	10.0	115.0	0.0	0.0	0.0	35.3	0.134	29.0	tested_negative
9	2.0	197.0	70.0	45.0	543.0	30.5	0.158	53.0	tested_positive
10	8.0	125.0	96.0	0.0	0.0	0.0	0.232	54.0	tested_positive
11	4.0	110.0	92.0	0.0	0.0	37.6	0.191	30.0	tested_negative
12	10.0	168.0	74.0	0.0	0.0	38.0	0.537	34.0	tested_positive
13	10.0	139.0	80.0	0.0	0.0	27.1	1.441	57.0	tested_negative
14	1.0	189.0	60.0	23.0	846.0	30.1	0.398	59.0	tested_positive
15	5.0	166.0	72.0	19.0	175.0	25.8	0.587	51.0	tested_positive
16	7.0	100.0	0.0	0.0	0.0	30.0	0.484	32.0	tested_positive
17	0.0	118.0	84.0	47.0	230.0	45.8	0.551	31.0	tested_positive
18	7.0	107.0	74.0	0.0	0.0	29.6	0.254	31.0	tested_positive
19	1.0	103.0	30.0	38.0	83.0	43.3	0.183	33.0	tested_negative
20	1.0	115.0	70.0	30.0	96.0	34.6	0.529	32.0	tested_positive
21	3.0	126.0	88.0	41.0	235.0	39.3	0.704	27.0	tested_negative
22	8.0	99.0	84.0	0.0	0.0	35.4	0.388	50.0	tested_negative
23	7.0	196.0	90.0	0.0	0.0	39.8	0.451	41.0	tested_positive
24	9.0	119.0	80.0	35.0	0.0	29.0	0.263	29.0	tested_positive

ตัวแปร class (dependent/target) มี type เป็น nominal มี 2 ค่า คือ tested negative/tested positive

ตัวแปร classifier (independent/predictor) มี 8 ตัว มี type เป็น numeric ทั้งหมด

- ดู descriptive statistics ของแต่ละตัวแปรที่ Weka สรุปให้ พร้อมแสดงกราฟ (histogram) หรือคลิก Visualize All เพื่อดู histogram ของตัวแปรทุกตัวพร้อมกัน



ข้อมูลนี้มีทั้งหมด 768 ตัว (Instances/Cases/N)

ตัวแปร 1 ถึง 8 เป็นตัวแปร classifier มี type เป็น numeric ดังนั้น descriptive statistics จึงเป็น Minimum, Maximum, Mean, Standard Deviation (StdDev)

ตัวแปร class จะเป็นตัวแปรสุดท้ายเสมอ มี type เป็น nominal

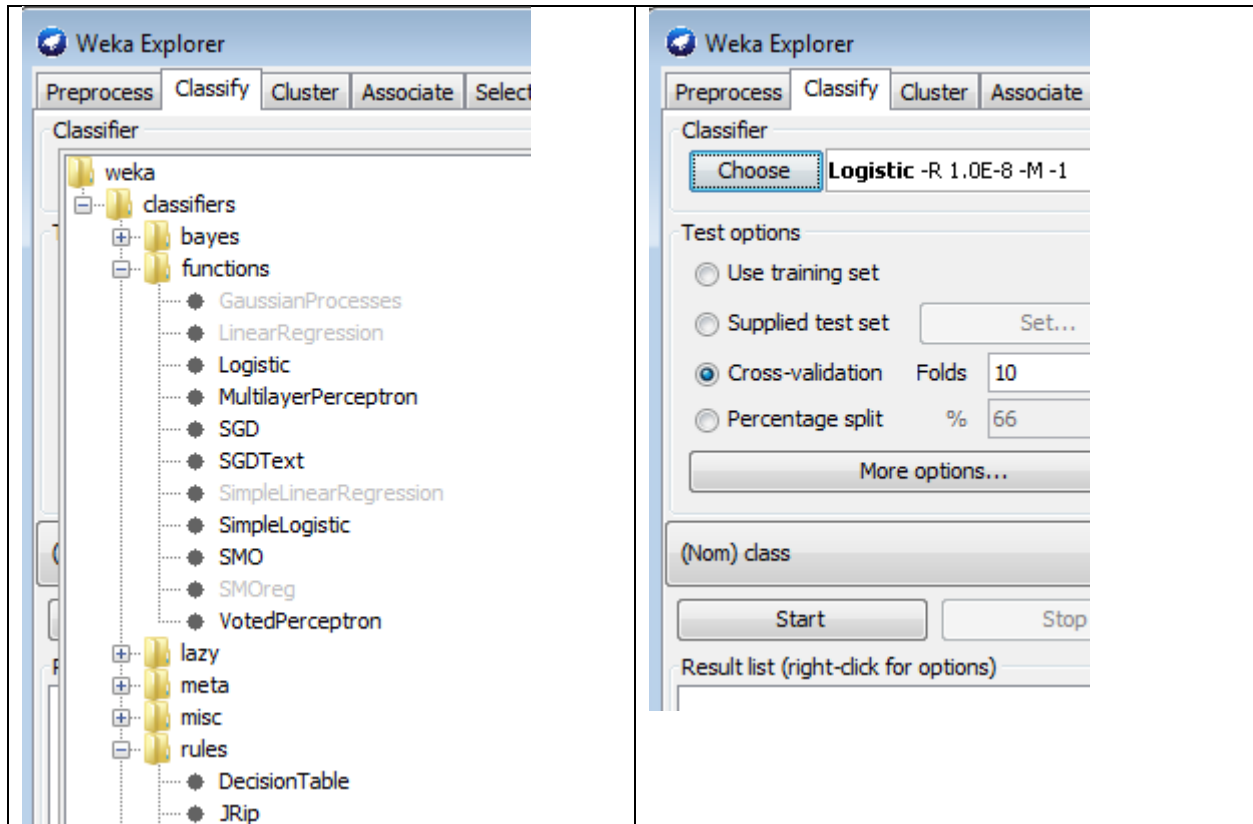
ลองคลิก Visualize All จะเห็นกราฟความสัมพันธ์ของตัวแปร class กับตัวแปร classifier แต่ละตัว

- Visualize scatter plot (ตัวแปร 2 ตัว) โดยคลิกที่แท็บ Visualize

- Fit logistic regression model

เลือกแท็บ Classify > Choose Classifier = functions > Logistic (Regression)

Default Test option = Cross validation Fold 10



กด Start จะได้ผลลัพธ์

6. การแปลผล

$$6.1 \text{ Model: Class} = e^{-0.1232 * \text{preg}} + e^{-0.0352 * \text{plas}} + e^{0.0133 * \text{pres}} + e^{-0.0006 * \text{skin}} + e^{0.0012 * \text{insu}} + e^{-0.0897 * \text{mass}} + e^{-0.9452 * \text{pedi}} + e^{-0.0149 * \text{age}} + 8.4047 \text{ หรือ}$$

$$\text{Class} = 0.8841 * \text{preg} + 0.9654 * \text{plas} + 1.0134 * \text{pres} + 0.9994 * \text{skin} + 1.0012 * \text{insu} + 0.9142 * \text{mass} + 0.3886 * \text{pedi} + 0.9852 * \text{age}$$

Classifier output		Odds Ratios...	
Logistic Regression with ridge Coefficients...		Variable	Class
Variable	tested_negative	tested_negative	
preg	-0.1232	preg	0.8841
plas	-0.0352	plas	0.9654
pres	0.0133	pres	1.0134
skin	-0.0006	skin	0.9994
insu	0.0012	insu	1.0012
mass	-0.0897	mass	0.9142
pedi	-0.9452	pedi	0.3886
age	-0.0149	age	0.9852
Intercept	8.4047		

6.2 Evaluation:

Correctly Classified Instances	593	77.2135 %
Incorrectly Classified Instances	175	22.7865 %

6.3 Deployment: แทนค่าลงในสมการข้อ 6.1

คลิกขวาที่ Result แล้ว Save model

- ให้ทดสอบทั้ง 2 test options (Use training set, Cross-validation Folds 10 จะได้ว่า Weka ใช้ full training set ในการสร้างโมเดล ซึ่งผลลัพธ์ของโมเดลจะเหมือนกัน แต่ accuracy จะไม่เหมือนกัน ในวิธีการ Cross-validation Folds 10 Weka จะแบ่งข้อมูลทั้งหมดออกเป็น 10 ส่วน และรัน 10 ครั้ง โดยครั้งที่ 1 ใช้ ส่วนที่ 1 เป็น test set ส่วนที่เหลือทั้งหมดอีก 9 ส่วนเป็น training set
ครั้งที่ 2 ใช้ ส่วนที่ 2 เป็น test set ส่วนที่เหลือทั้งหมดอีก 9 ส่วนเป็น training set
จนถึงครั้งที่ 10 ใช้ ส่วนที่ 10 เป็น test set ส่วนที่เหลือทั้งหมดอีก 9 ส่วนเป็น training set โดย evaluation accuracy จะเป็นค่าเฉลี่ยของทั้ง 10 ครั้ง
จากนั้นจะรันโมเดลอีกครั้งโดยใช้ full training set ทั้งหมดในการรันโมเดล

WS#3: Classification by ML algorithm

Data set: diabetes

1. Preprocess > Open file เลือก diabetes
2. Fit classification model Classifier = rules > **ZeroR** (baseline accuracy)

การแปลผล

Model: Class = tested_negative

```
=== Classifier model (full training set) ===  
  
ZeroR predicts class value: tested_negative
```

Evaluation:

```
Correctly Classified Instances      500      65.1042 %  
Incorrectly Classified Instances    268      34.8958 %  
  
=== Confusion Matrix ===  
  
   a  b  <-- classified as  
500  0  |  a = tested_negative  
268  0  |  b = tested_positive
```

Deployment: tested negative

3. Fit classification model Classifier = rules > OneR

การแปลผล

Model: ไม่ make sense

```
=== Classifier model (full training set) ===  
  
plas:  
  < 114.5 -> tested_negative  
  < 115.5 -> tested_positive  
  < 127.5 -> tested_negative  
  < 128.5 -> tested_positive  
  < 133.5 -> tested_negative  
  < 135.5 -> tested_positive  
  < 143.5 -> tested_negative  
  < 152.5 -> tested_positive  
  < 154.5 -> tested_negative  
  >= 154.5      -> tested_positive  
  
(587/768 instances correct)
```

Evaluation: (cross-validation)

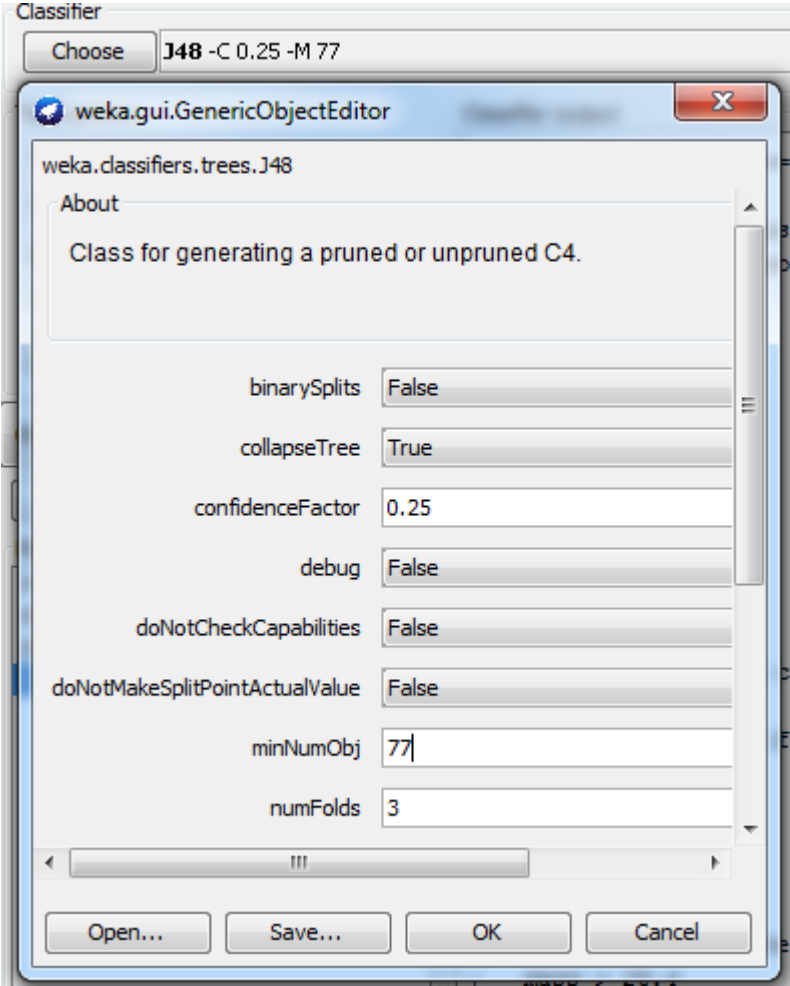
```
Correctly Classified Instances      549      71.4844 %  
Incorrectly Classified Instances    219      28.5156 %
```

```
=== Confusion Matrix ===
      a   b  <-- classified as
433  67 | a = tested_negative
152 116 | b = tested_positive
```

Deployment: ใช้กฎ (apply rule)

4. Fit classification model Classifier = trees > J48

หมายเหตุ ตาม default parameter จะได้จะได้ 20 กฎ ซึ่งบางกฎ ก็เป็นจริงสำหรับข้อมูลจำนวนน้อย ดังนั้นควรจะปรับ parameter “minNumObj” เช่นถ้าต้องการให้กฎเป็นจริงสำหรับข้อมูล 10% ก็ปรับ minNumObj = 77 โดยคลิกที่ **J48** แก้ minNumObj



การแปลผล

Model: คลิกขวาที่ Result เลือก Visualize tree

```
=== Classifier model (full training set) ===  
  
J48 pruned tree  
-----  
  
plas <= 127: tested_negative (485.0/94.0)  
plas > 127  
| mass <= 30: tested_negative (78.0/26.0)  
| mass > 30: tested_positive (205.0/57.0)  
  
Number of Leaves :      3  
  
Size of the tree :      5
```

```
graph TD
    Root((plas)) -- "<= 127" --> L[tested_negative (485.0/94.0)]
    Root -- "> 127" --> R((mass))
    R -- "<= 30" --> RL[tested_negative (78.0/26.0)]
    R -- "> 30" --> RR[tested_positive (205.0/57.0)]
```

Evaluation: (cross-validation)

```
=== Stratified cross-validation ===  
=== Summary ===  
  
Correctly Classified Instances      570           74.2188 %  
Incorrectly Classified Instances    198           25.7813 %  
-----  
=== Confusion Matrix ===  
  
   a  b  <-- classified as  
445  55 |  a = tested_negative  
143 125 |  b = tested_positive
```

Deployment: คลิกขวาที่ Result แล้ว Save model

7. ให้เปรียบเทียบผลการทดลองด้วย Classifier ตัวอื่น ๆ เช่น rules > PART, rules > JRIP (จะดำเนินการใน Experimenter Interface)

WS#4: Association Problem Type

Data set: supermarket

1. Preprocess > Open file เลือก supermarket

Edit เพื่อดูการจัดเก็บข้อมูล

at	20: canned fruit Nominal	21: canned vegetables Nominal	22: breakfast food Nominal	23: cigs-tobacco pkts Nominal
		t		
t		t		
t				
	t		t	
	t		t	
t	t		t	
	t			
t	t			
t	t		t	
t	t			
			t	t
			t	
t	t		t	
t				
			t	t
				t
	t		t	

แถวหนึ่ง ๆ คือตะกร้าสินค้าหนึ่ง ๆ t (=true) มีสินค้าชนิดนั้น

2. Fit Association model โดยเลือกแท็บ Associate

The image shows two screenshots from the Weka GUI. The left screenshot shows the 'Associate' tab selected in the top menu, and the 'Apriori' algorithm chosen in the 'Associator' list. The right screenshot shows the 'weka.gui.GenericObjectEditor' dialog for the 'Apriori' class, with the following parameters set:

- car: False
- classIndex: -1
- delta: 0.05
- lowerBoundMinSupport: 0.2
- metricType: Confidence
- minMetric: 0.9
- numRules: 10

เลือก Associator algorithm เป็น

Apriori

คลิกที่ Apriori (รูปทางขวา)

ปรับ parameter ที่สำคัญ 3 ตัวคือ

-lowerBoundMinSupport = 0.2

-minMetric (min Confidence) = 0.9

-numRules = 10

ผลลัพธ์: ไม่พบกฎใด ๆ เลย

ลองปรับ

-minMetric (min Confidence) = 0.85

การแปลผล

Model:

Best rules found:

```

1. biscuits=t frozen foods=t fruit=t vegetables=t 1039 ==> bread and cake=t 929 <conf:(0.89)> lift:(1.24) lev:(0.04) [182] conv:(2.49)
2. fruit=t vegetables=t total=high 1050 ==> bread and cake=t 938 <conf:(0.89)> lift:(1.24) lev:(0.04) [182] conv:(2.49)
3. fruit=t total=high 1243 ==> bread and cake=t 1104 <conf:(0.89)> lift:(1.23) lev:(0.05) [209] conv:(2.49)
4. biscuits=t total=high 1228 ==> bread and cake=t 1082 <conf:(0.88)> lift:(1.22) lev:(0.04) [198] conv:(2.34)
5. milk-cream=t total=high 1217 ==> bread and cake=t 1071 <conf:(0.88)> lift:(1.22) lev:(0.04) [195] conv:(2.32)
6. biscuits=t margarine=t vegetables=t 1054 ==> bread and cake=t 925 <conf:(0.88)> lift:(1.22) lev:(0.04) [166] conv:(2.27)
7. frozen foods=t total=high 1273 ==> bread and cake=t 1117 <conf:(0.88)> lift:(1.22) lev:(0.04) [200] conv:(2.27)
8. biscuits=t margarine=t fruit=t 1073 ==> bread and cake=t 938 <conf:(0.87)> lift:(1.21) lev:(0.04) [165] conv:(2.21)
9. party snack foods=t total=high 1120 ==> bread and cake=t 979 <conf:(0.87)> lift:(1.21) lev:(0.04) [172] conv:(2.21)
10. vegetables=t total=high 1270 ==> bread and cake=t 1110 <conf:(0.87)> lift:(1.21) lev:(0.04) [195] conv:(2.21)

```

Evaluation: Min Support และ Confidence

Deployment: เอา กฎ ไปใช้

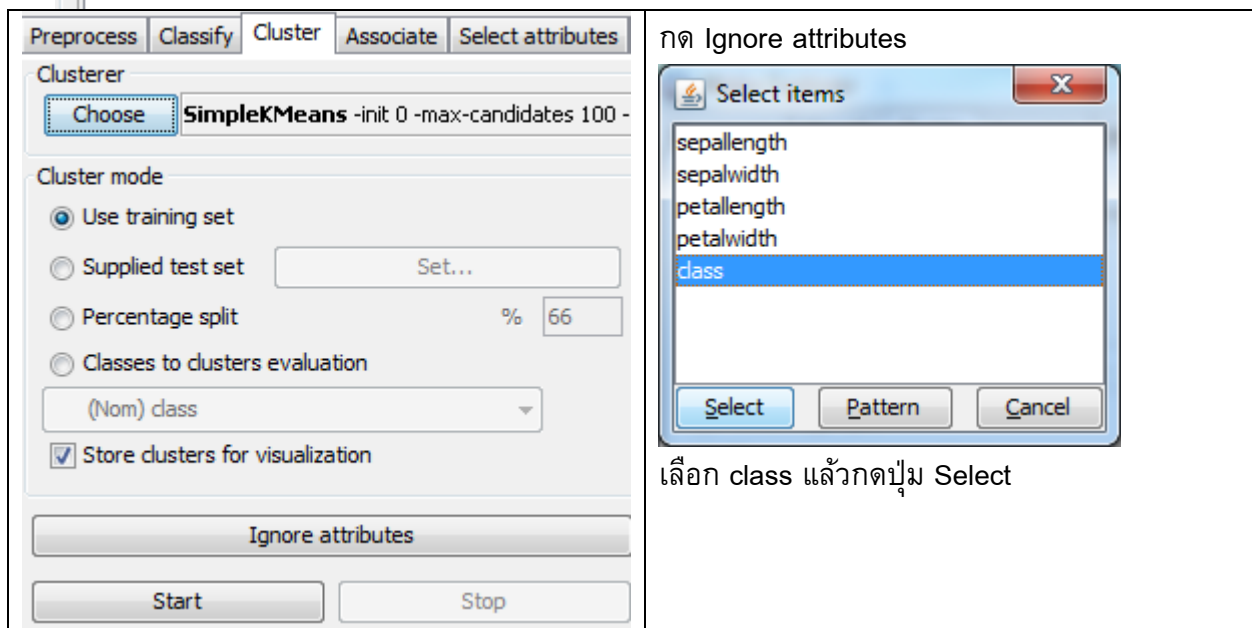
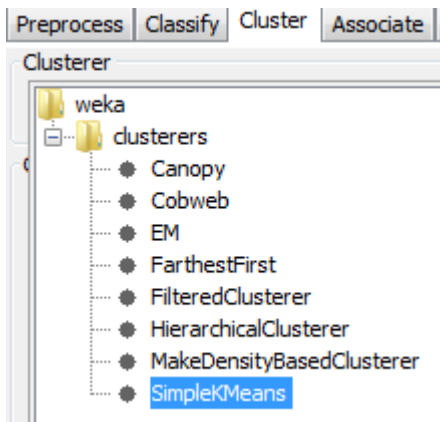
WS#5: Clustering

Data set: iris

1. Preprocess > Open file เลือก **iris**

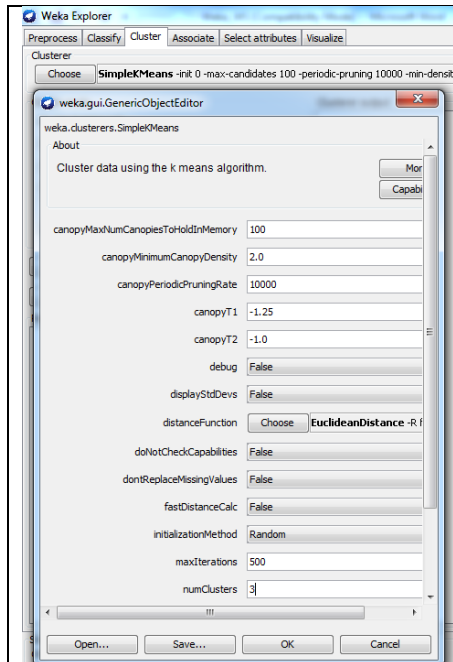
ในที่นี้ iris มีตัวแปร class แต่เราจะไม่ใช่ในการ clustering แต่จะใช้ในการเปรียบเทียบกับค่าจากโมเดล

2. Fit Clustering model โดยเลือกแท็บ Cluster



กด Ignore attributes

เลือก class แล้วกดปุ่ม Select



คลิก SimpleKMeans
ปรับ numCluster = 3

การแปลผล

Model:

Final cluster centroids:				
Attribute	Full Data (150)	Cluster#		
		0 (61)	1 (50)	2 (39)
sepalength	5.8433	5.8885	5.006	6.8462
sepalwidth	3.054	2.7377	3.418	3.0821
petalength	3.7587	4.3967	1.464	5.7026
petalwidth	1.1987	1.418	0.244	2.0795

Evaluation:

Within cluster sum of squared errors: 6.998114004826762

Deployment: คลิกขวาที่ Result แล้ว Save model

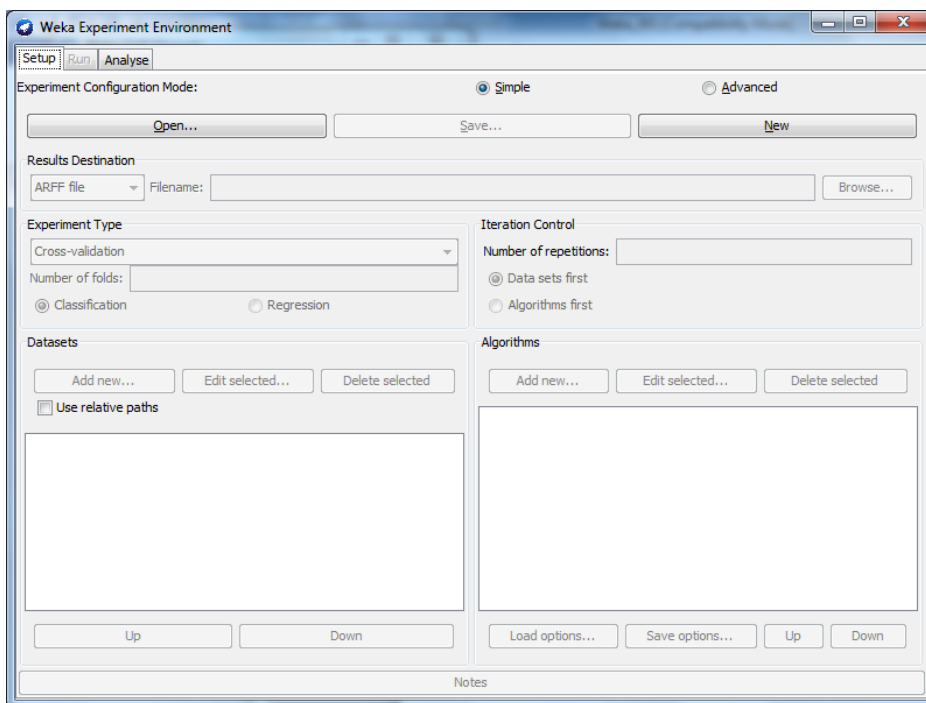
WS#6 จงเปรียบเทียบการวิเคราะห์โจทย์ Classification โดยใช้หลาย algorithms

Experiment 1

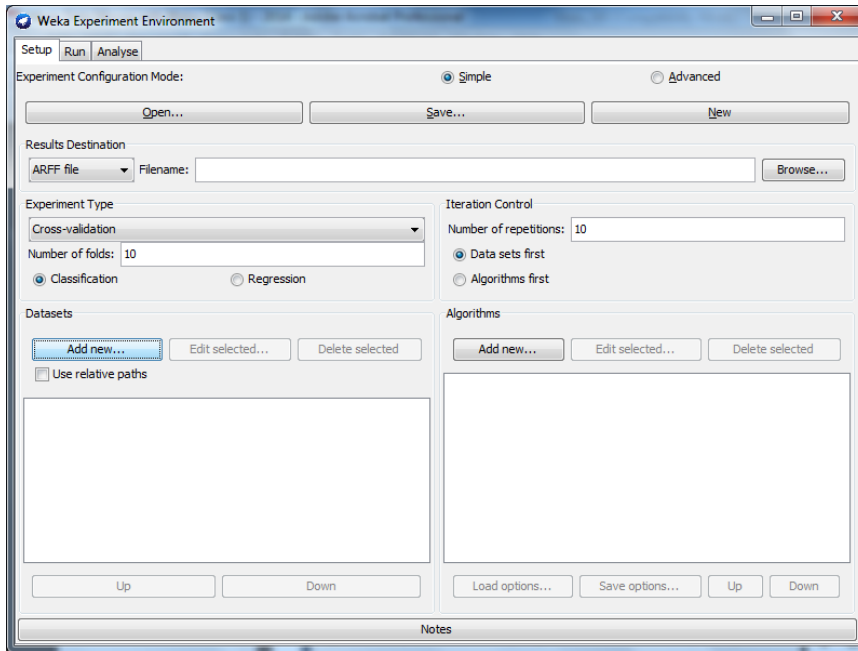
Data set: iris

Algorithms: trees > J48, rules > OneR, rules > ZeroR

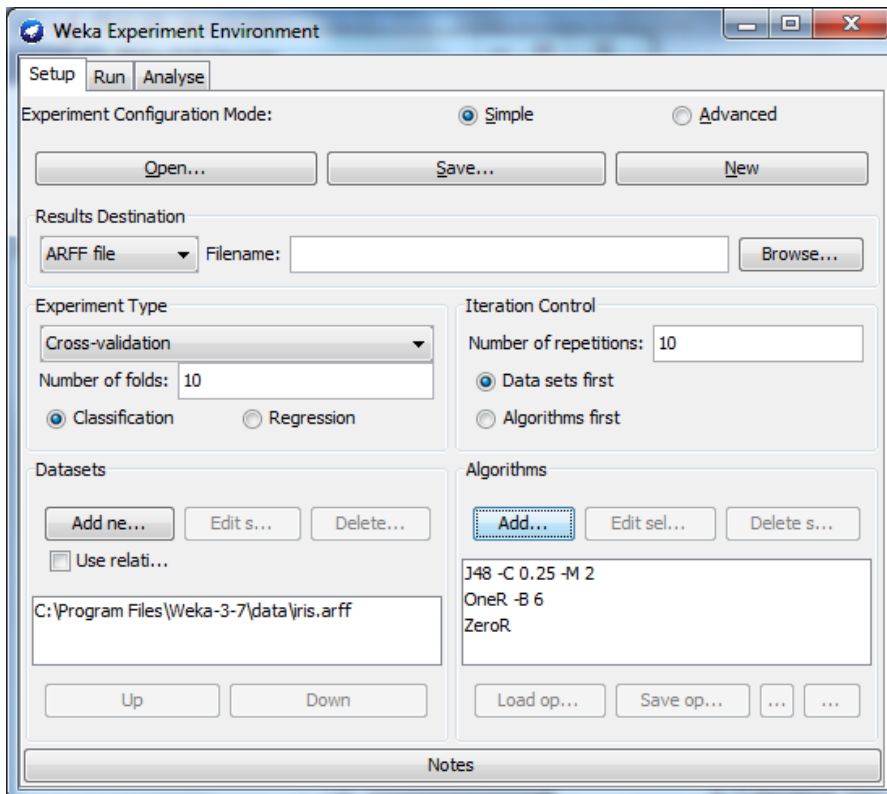
1. Weka GUI Chooser เลือก Experimenter



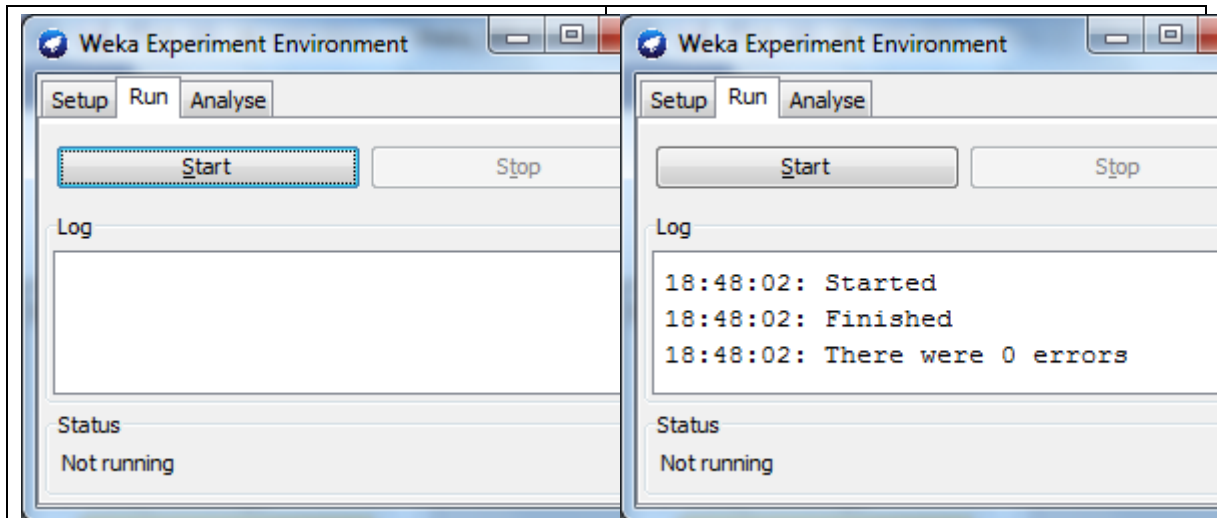
2. คลิก New



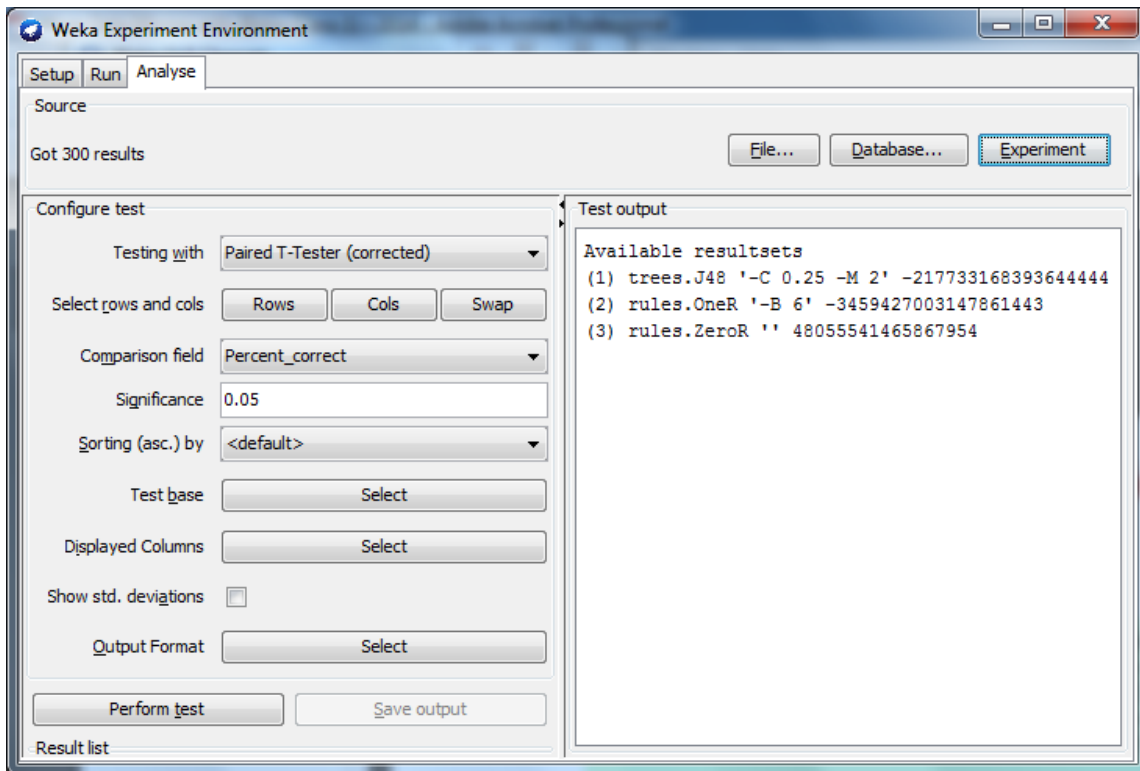
3. ที่ Datasets คลิก Add new แล้วเปิดเลือก iris
ที่ Algorithms คลิก Add new แล้วเลือก Choose: rules > ZeroR, rules > OneR, trees > J48
(Add new 3 ครั้ง ตาม default)



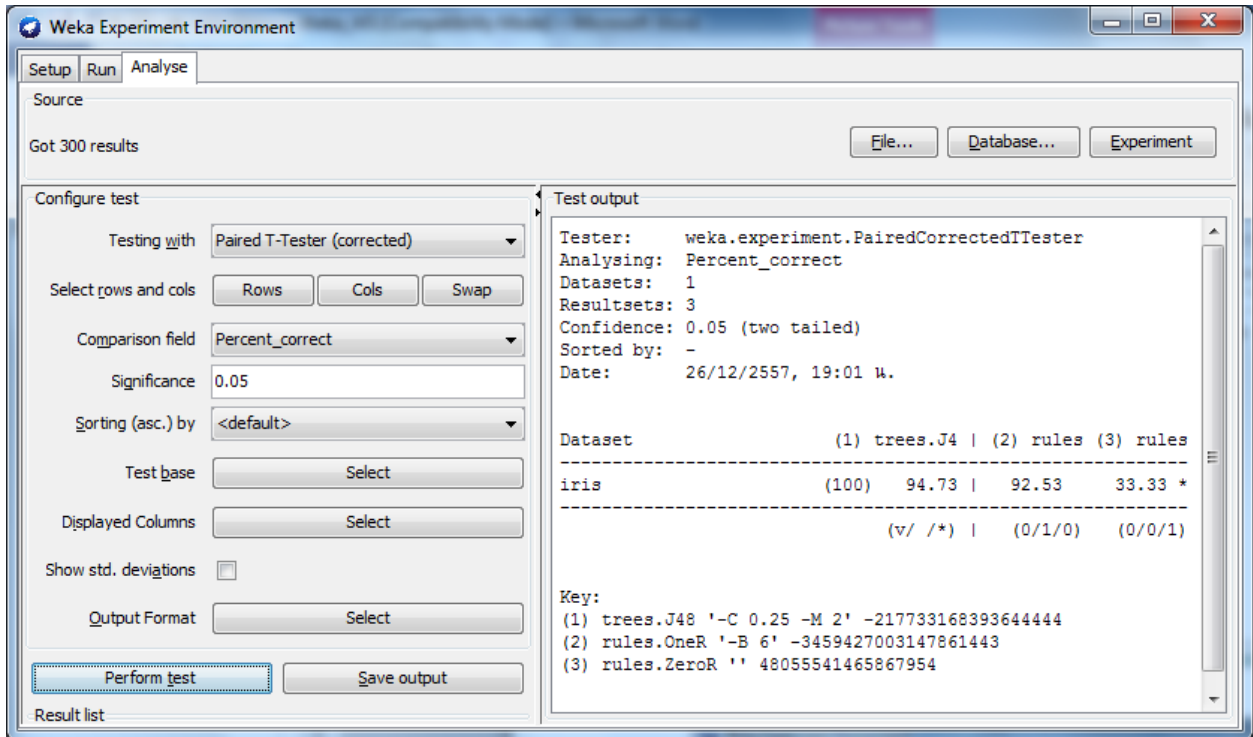
4. ไปที่แท็บ Run คลิก Start



5. ไปที่แท็บ Analyze คลิก Experiment



6. คลิก Perform test



7. การแปลผลจาก Witten

```

Dataset          (1) trees.J4 | (2) rules (3) rules
-----
iris              (100)  94.73 |  92.53   33.33 *
-----
                   (v/ /*) | (0/1/0)  (0/0/1)

Key:
(1) trees.J48
(2) rules.OneR
(3) rules.ZeroR
    
```

v significantly better
* significantly worse

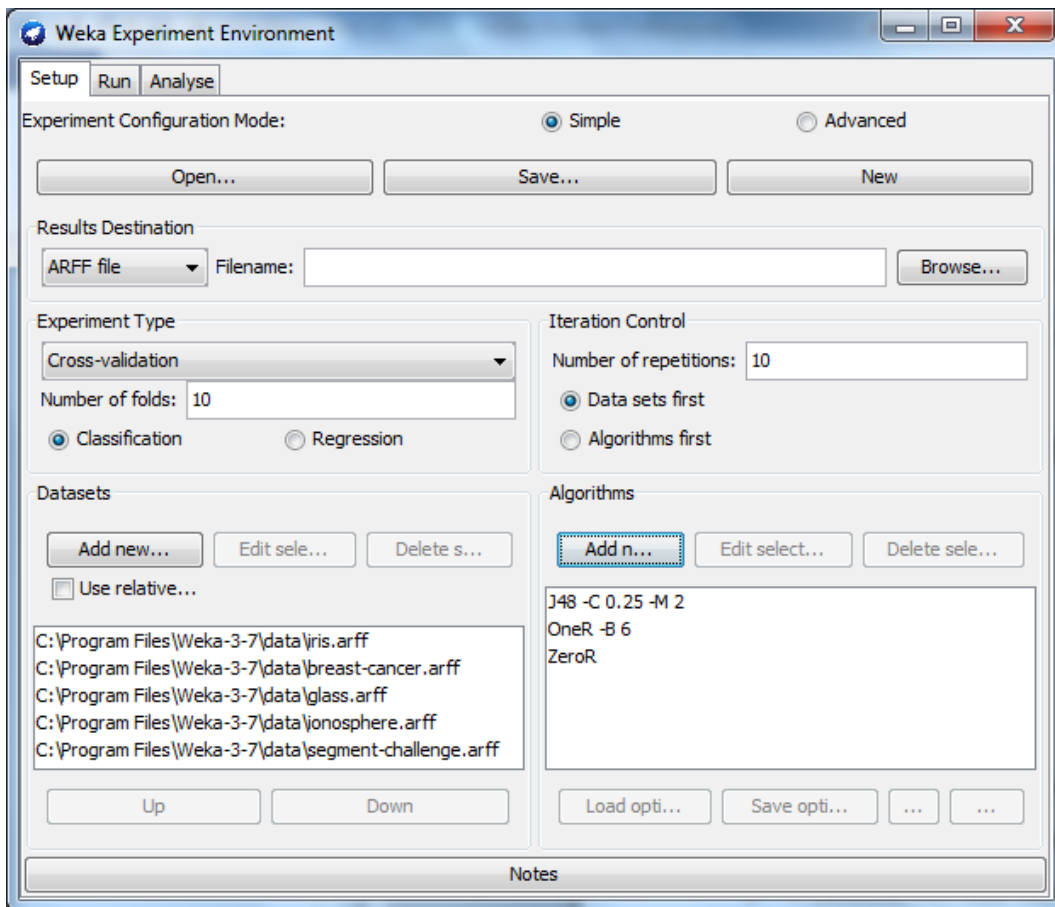
- ❖ ZeroR (33.3%) is significantly worse than J48 (94.7%)
- ❖ Cannot be sure that OneR (92.5%) is significantly worse than J48
- ❖ ... at the 5% level of statistical significance
- ❖ J48 seems better than ZeroR: pretty sure (5% level) that this is not due to chance

Experiment 2:

Data set: iris, breast-cancer, glass, ionosphere, segment-challenge

Algorithms: trees > J48, rules > OneR, rules > ZeroR

1. ไปที่แท็บ Set up คลิก New
2. ที่ Datasets คลิก Add new แล้วเปิดเลือก iris, breast-cancer, glass, ionosphere, segment-challenge (Add new 5 ครั้ง)
ที่ Algorithms คลิก Add new แล้วเลือก Choose: rules > ZeroR, rules > OneR, trees > J48 (Add new 3 ครั้ง ตาม default)



3. ไปที่แท็บ Run คลิก Start
4. ไปที่แท็บ Analyze คลิก Experiment

5. คลิก Perform test

The screenshot shows the Weka Experiment Environment window. The 'Configure test' panel on the left has the 'Perform test' button highlighted with a blue border. The 'Test output' panel on the right displays the results of a Paired T-Tester (corrected) test. The test compares 'trees.J48' and 'rules.OneR' across five datasets: iris, breast-cancer, Glass, ionosphere, and segment. The results show that 'trees.J48' generally performs better than 'rules.OneR' across most datasets.

Test output:

```

Tester:      weka.experiment.PairedCorrectedTTester
Analysing:   Percent_correct
Datasets:    5
Resultsets:  3
Confidence:  0.05 (two tailed)
Sorted by:   -
Date:        26/12/2557, 19:26 น.

Dataset      (1) trees.J4 | (2) rules (3) rules
-----
iris          (100)  94.73 |  92.53   33.33 *
breast-cancer (100)  74.28 |  66.91 *  70.30
Glass         (100)  67.63 |  57.40 *  35.51 *
ionosphere    (100)  89.74 |  82.28 *  64.10 *
segment       (100)  95.71 |  64.35 *  15.73 *

(v/ /*) | (0/1/4) (0/1/4)

Key:
(1) trees.J48 '-C 0.25 -M 2' -217733168393644444
(2) rules.OneR '-B 6' -3459427003147861443
(3) rules.ZeroR '' 48055541465867954
    
```

6. เปลี่ยน Test base Select เป็น rules.OneR

The screenshot shows the Weka Experiment Environment window after the 'Test base' has been changed to 'rules.OneR'. The 'Test output' panel now shows the results of a Paired T-Tester (corrected) test comparing 'trees.J48' and 'rules.OneR'. The results show that 'rules.OneR' performs better than 'trees.J48' across all five datasets.

Test output:

```

Tester:      weka.experiment.PairedCorrectedTTester
Analysing:   Percent_correct
Datasets:    5
Resultsets:  3
Confidence:  0.05 (two tailed)
Sorted by:   -
Date:        26/12/2557, 19:50 น.

Dataset      (2) rules.On | (1) trees (3) rules
-----
iris          (100)  92.53 |  94.73   33.33 *
breast-cancer (100)  66.91 |  74.28 v  70.30
Glass         (100)  57.40 |  67.63 v  35.51 *
ionosphere    (100)  82.28 |  89.74 v  64.10 *
segment       (100)  64.35 |  95.71 v  15.73 *

(v/ /*) | (4/1/0) (0/1/4)

Key:
(1) trees.J48 '-C 0.25 -M 2' -217733168393644444
(2) rules.OneR '-B 6' -3459427003147861443
    
```