Introduction to Data Mining

Lab 1: Introduction to Weka

Name: Phạm Đức Đạt

ID Student: ITITIU20184

1.1. Introduction

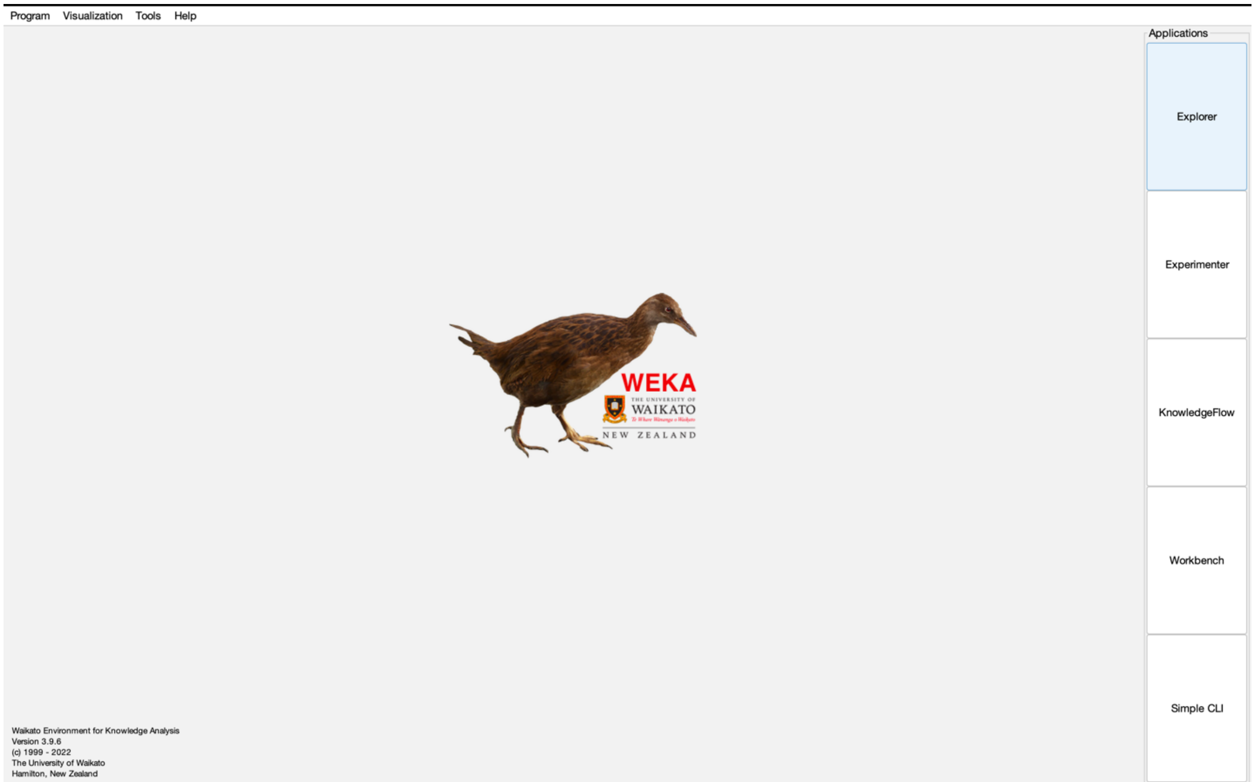
Weka is an open-source software available at www.cs.waikato.ac.nz/ml/weka. Weka stands for the Waikato Environment for Knowledge Analysis. It offers clean, spare implementation of the simplest techniques, designed to aid understanding of the data mining techniques. It also provides a work-bench that includes full, working, state-of-the-art implementations of many popular learning schemes that can be used for practical data mining or for research.

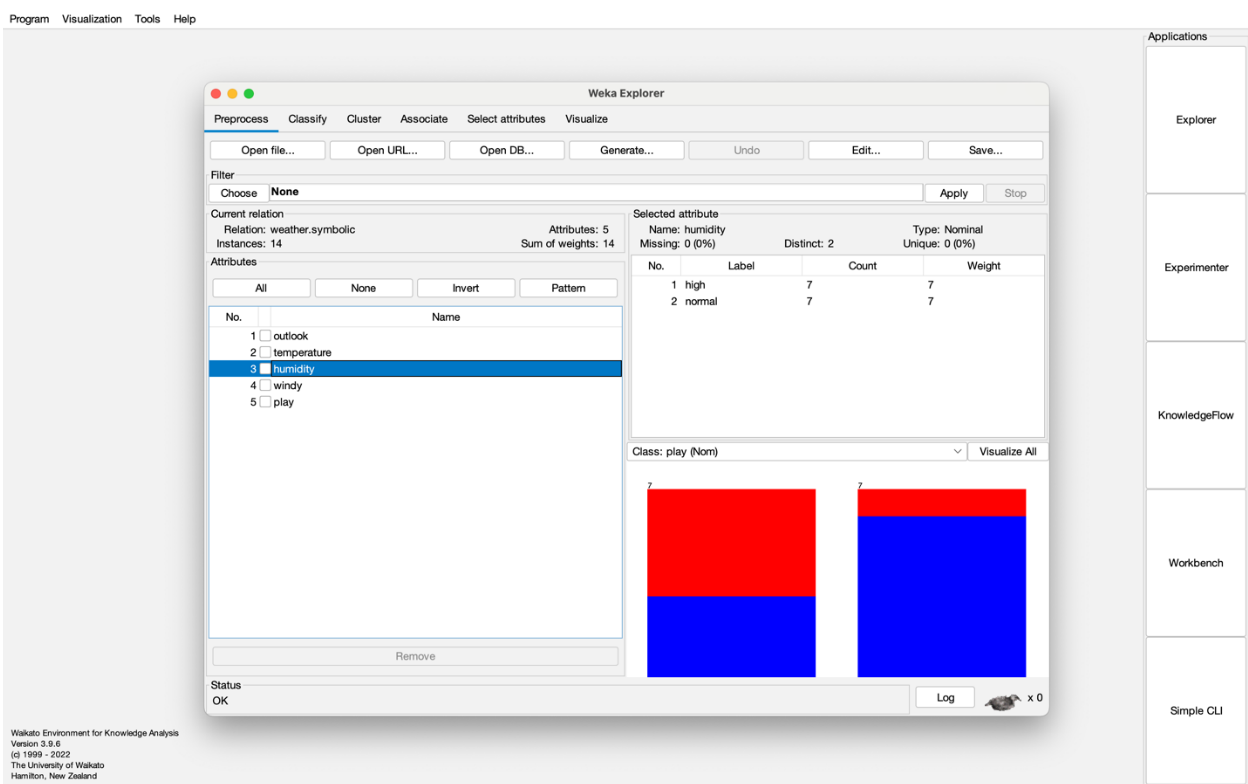
In the first class, we are going to get started with Weka: exploring the "Explorer" interface, exploring some datasets, building a classifier, using filters, and visualizing your dataset. (See the lecture of class 1 by Ian H. Witten, [1])

Task: Taking notes how you find the Explorer, and answering questions in the following sections

1.2. Exploring the Explorer

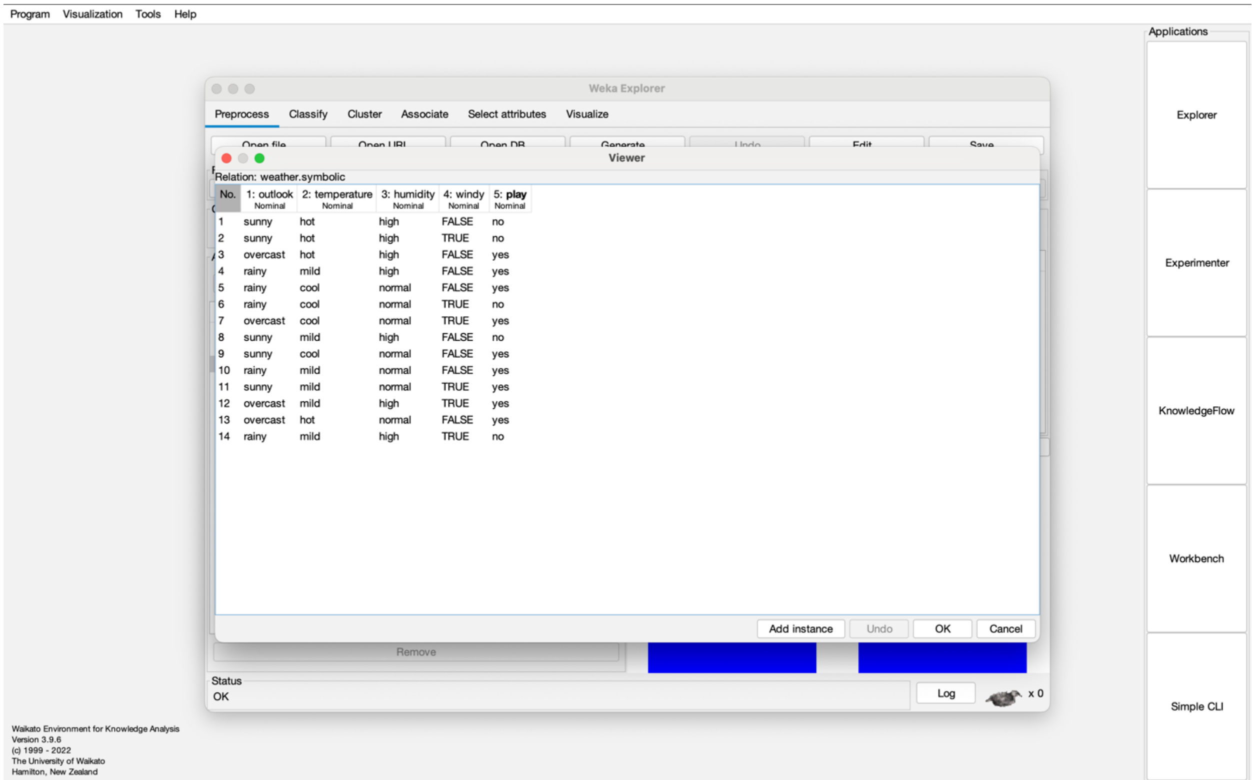
Follow the instructions in [1]





1.3. Exploring datasets

Follow the instructions in [1]



In dataset weather.nominal.arff, how many attributes are there in the relation? What are their values? What is the class and its values? Counting instances for each attribute value.

Dataset	Attributes	Values	#Instances
Relation: weather.nominal #Instances:14 #Attributes: 5	Outlook	sunny overcast rainy	5 4 5 Distinct: 3
	Temparature	hot mid cold	4 6 4 Distinct: 3
	Humidity	high normal	7 7 Distinct: 2
	Windy	TRUE, FALSE	6 8 Disctinct: 2
Class (Play)	Play	Yes No	9 5 Distinct: 2

Similarly, examine datasets: weather.numeric.arff and glass.arff.

Dataset	Attributes	Values	#Instances
Relation: weather #Instances: 14 #Attributes: 5	Outlook	sunny overcast rainy	5 4 5 Distinct: 3
	Temparature	Minimum 64 Maximum 85 Mean 73.571 StdDev 6.572	Distinct: 12
	Humidity	Minimum 65 Maximum 96 Mean 81.643 StdDev 10.285	Distinct: 10
	Windy	True False	6 10 Distinct: 2
Class (Play)	Play	Yes No	9 5 Distinct: 2

Dataset	Attributes	Values	#Instances
Relation: glass #Instances: 214 #Attributes:10	RI	Minimum 1.511 Maximum 1.534 Mean 1.518 StdDev 0.003	
	Na	Minimum 10.73 Maximum 17.38 Mean 13.408 StdDev 0.817	
	Mg	Minimum 0 Maximum 4.49 Mean 2.685 StdDev 1.442	
	Al	Minimum 0.29 Maximum 3.5 Mean 1.445 StdDev 0.499	
	Si	Minimum 69.81 Maximum 75.41 Mean 72.651 StdDev 0.775	
	K	Minimum 0 Maximum 6.21	

		Mean 0.497 StdDev 0.652	
	Ca	Minimum 5.43 Maximum 16.19 Mean 8.957 StdDev 1.423	
	Ba	Minimum 0 Maximum 3.15 Mean 0.175 StdDev 0.497	
	Fe	Minimum 0 Maximum 0.51 Mean 0.057 StdDev 0.097	
Class	Type	build wind float build wind non-float vehic wind float vehic wind non-float containers tableware headlamps	70 76 17 0 13 9 29 Distince: 6

Create a file of ARFF format and examine it.

Dataset	**Attributes**	**Values**	**#Instances**
Relation: attandanse #Instances: 5 #Attributes: 5	Name	Nominal	Continuous
	Class	Nominal	continuous
	Average_Score	Numeric	Continuous
	Extracurricular	Nomnal	Continuous
	Special_Achievement	Nominal	Continuous
Class	Attandance	Yes No	Continuous

1.4. Building a classifier

Follow the instructions in [1]

Examine the output of J48 vs. RandomTree applied to dataset glass.arff

Algorithm	Pruned/unpruned	minNumObj	No. of Leaves	Correctly Classified Instances
J48	unpruned	15	8	131 (61.215%)
Random Tree	N/A	15	11	134 (62.6168%)

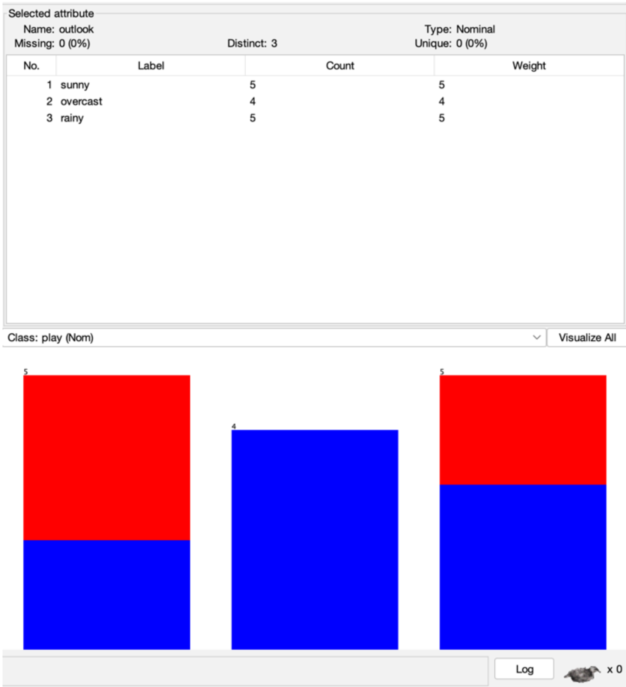
Evaluate the confusion matrix every time running an algorithm.

```
=== Confusion Matrix ===

  a  b  c  d  e  f  g   <-- classified as
 43 25  1  0  0  0  1 |  a = build wind float
 19 41  3  0  5  6  2 |  b = build wind non-float
  9  6  2  0  0  0  0 |  c = vehic wind float
  0  0  0  0  0  0  0 |  d = vehic wind non-float
  0  0  0  0 11  1  1 |  e = containers
  1  0  0  0  0  8  0 |  f = tableware
  1  0  0  0  1  1 26 |  g = headlamps
```

```
=== Confusion Matrix ===

  a  b  c  d  e  f  g   <-- classified as
 48 19  3  0  0  0  0 |  a = build wind float
 17 48  3  0  6  1  1 |  b = build wind non-float
  6  4  7  0  0  0  0 |  c = vehic wind float
  0  0  0  0  0  0  0 |  d = vehic wind non-float
  0  6  0  0  5  1  1 |  e = containers
  0  3  0  0  1  3  2 |  f = tableware
  2  3  0  0  0  1 23 |  g = headlamps
```

1.5. Using a filter

Follow the instructions in [1], and remark



_Use a filter to remove an attribute →

- What are attributeIndices? – Range of attributes to be acted upon by the filter.

_Remove instances where humidity is high →

- What are nominalIndices? - Range of label indices to be used for selection on nominal attribute.

_Fewer attributes, better classification:

This is not true for all cases. If it is true, then it is highly possible that the removed attributes prove to be no more than unnecessary complications to the model, or it is because the model cannot find the global optimum by including those attributes. However, in cases where important attributes are removed (such as attribute=size-measures to classify cats or tigers) then there will be major blows that deteriorate the classification results. Either way, the notion that fewer attributes can lead to better classification requires observations and experiments to confirm, it depends both on the model and the set of attributes.

Follow the instructions in [1], review the outputs of J48 applied to glass.arff:

Filter	**Leaf size**	**Correctly Classified Instances**	**Remark**
Original	30	66.8224%	Fewer attributes, higher accuracy
Remove Fe	26	67.2897%	

Remove all attributes except RI and MG	21	68.6916%	

1.6. Visualizing your data

Follow the instructions in [1], how do you find “Visualize classifier errors”?

After running **J48** for *iris.arff*, determine:

- How many instances are predicted wrong? - 9 (given J48 classifier - unpruned - minNumObj=15).
- What are they?

Instance	Predicted class	Actual class
119	Iris-virginica	Iris-versicolor
98	Iris-versicolor	Iris-setosa
15	Iris-virginica	Iris-versicolor
109	Iris-versicolor	Iris-virginica
73	Iris-virginica	Iris-versicolor