DBTech EXT* Virtual Laboratory Workshop on Business Intelligent (BI) & Knowledge Discovery from Databases (KDD)

Education and Culture DG
Lifelong Learning Programme

Hands-On Laboratory Exercises – WEKA®

WEKA® Hands-On Laboratory Exercise No. 1

(by L. Karamitopoulos and D.A. Dervos)

This exercise is a revised version that includes a pre-processing phase

A. The Case Study

The file **weather_revised.csv** concerns the conditions that are suitable for playing some sport game. Each one data line corresponds to a specific day and includes the values of four weather attributes:
outlook: {1-sunny, 2-overcast, 3-rainy}
temperature: Fahrenheit degrees
humidity: percentages
windy: {0-No, 1-Yes}

In addition to that, each one data line includes the value of the attribute *play,* which indicates whether the game is held or not.
play: {0-the game is not held, 1-the game is held}.

The objective of this case study is to determine the suitable weather conditions under which the game is more likely to be held. This kind of knowledge will be represented in the form of **Association Rules**.

B. Description of the Dataset

Open **WEKA Explorer**, and then, open the file weather_revised.csv (**Open file…**).

At the left side of the window (**Figure 1**), we are informed that there are 14 instances (days) and 5 attributes. Also, we notice that the first attribute is selected (*outlook*), and its type (numeric), the corresponding statistics and histogram are shown at the right side of the window. Similarly, we may select the attribute *play* at the left side and identify its type (nominal) and its values at the right side of the window along with the frequency distribution and the bar chart. The colors in diagrams are dictated by the attribute that WEKA considers as the class attribute just above the diagrams (by default, WEKA identifies the last attribute in the dataset as the class attribute).

Remember that in order to generate association rules with WEKA, all attributes must be nominal. Notice, however, that the attributes *temperature* and *humidity* are numeric. In addition to that, WEKA identifies the attributes *outlook* and *windy* as numeric while we know that they are nominal (the numeric values are just codes). This fact implies that we must pre-process the dataset.

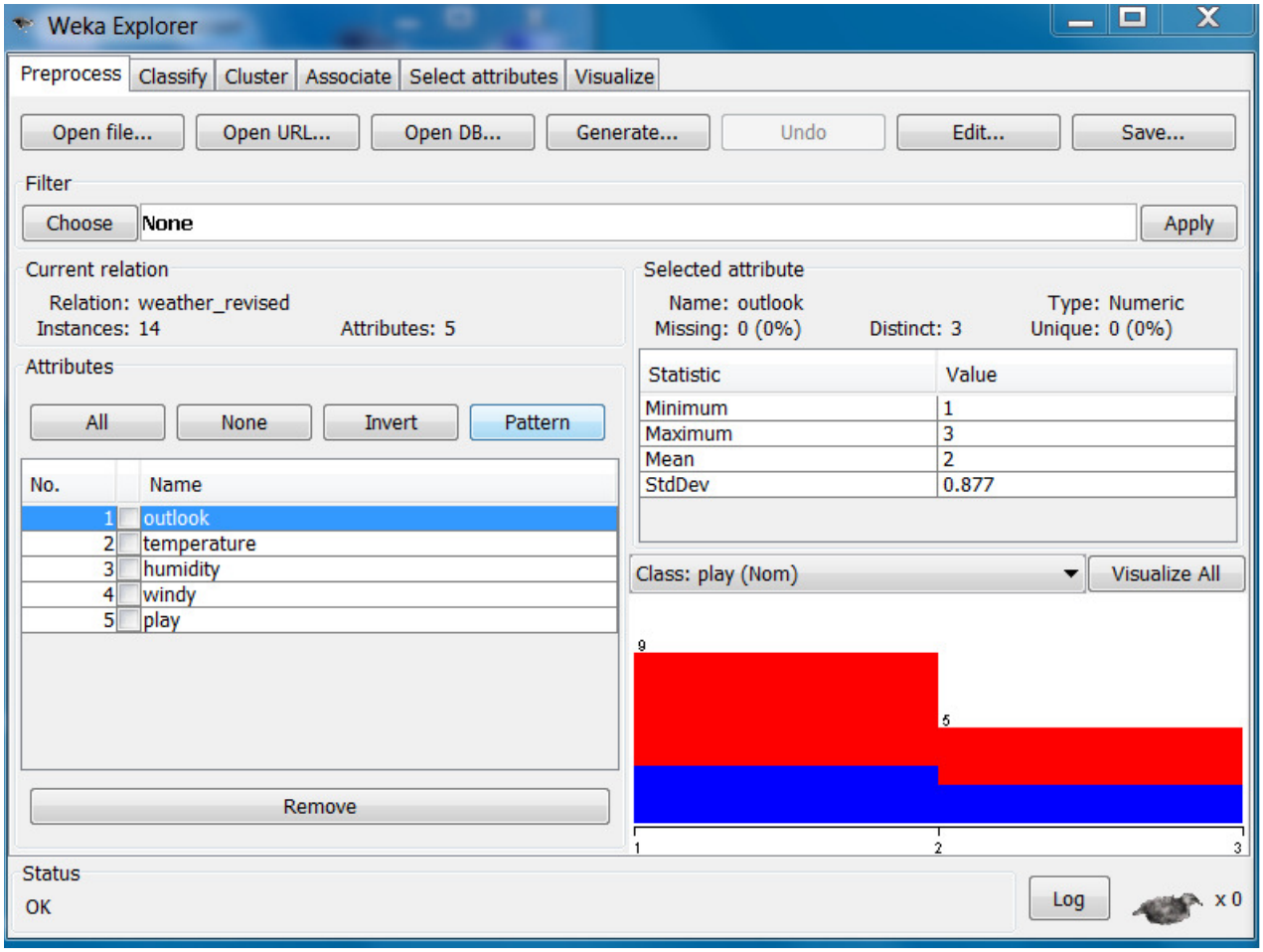


Figure 1. Identifying the basic features of the dataset

C. Data Pre-processing

As it is stated in the previous section, we must convert the numeric attributes into nominal. There are two different transformations that need to be applied.

First, the code (numeric) values of the attributes *outlook* and *windy* must be considered as "labels" rather than numbers. The filter that we use is called *NumericToNominal.*

Click on the button **Choose** at the upper left part of the Preprocess tab and follow the path: **filters – Unsupervised – Attribute – NumericToNominal.**

Click on the selected object (**Figure 2**), in order to edit its properties (the settings of the NumericToNominal filter) and set attributeIndices as 1,4 (No. of the attributes *outlook* and *windy*).

Click on **OK** ant then on **Apply** button. Check the type and the statistics of the transformed attributes.

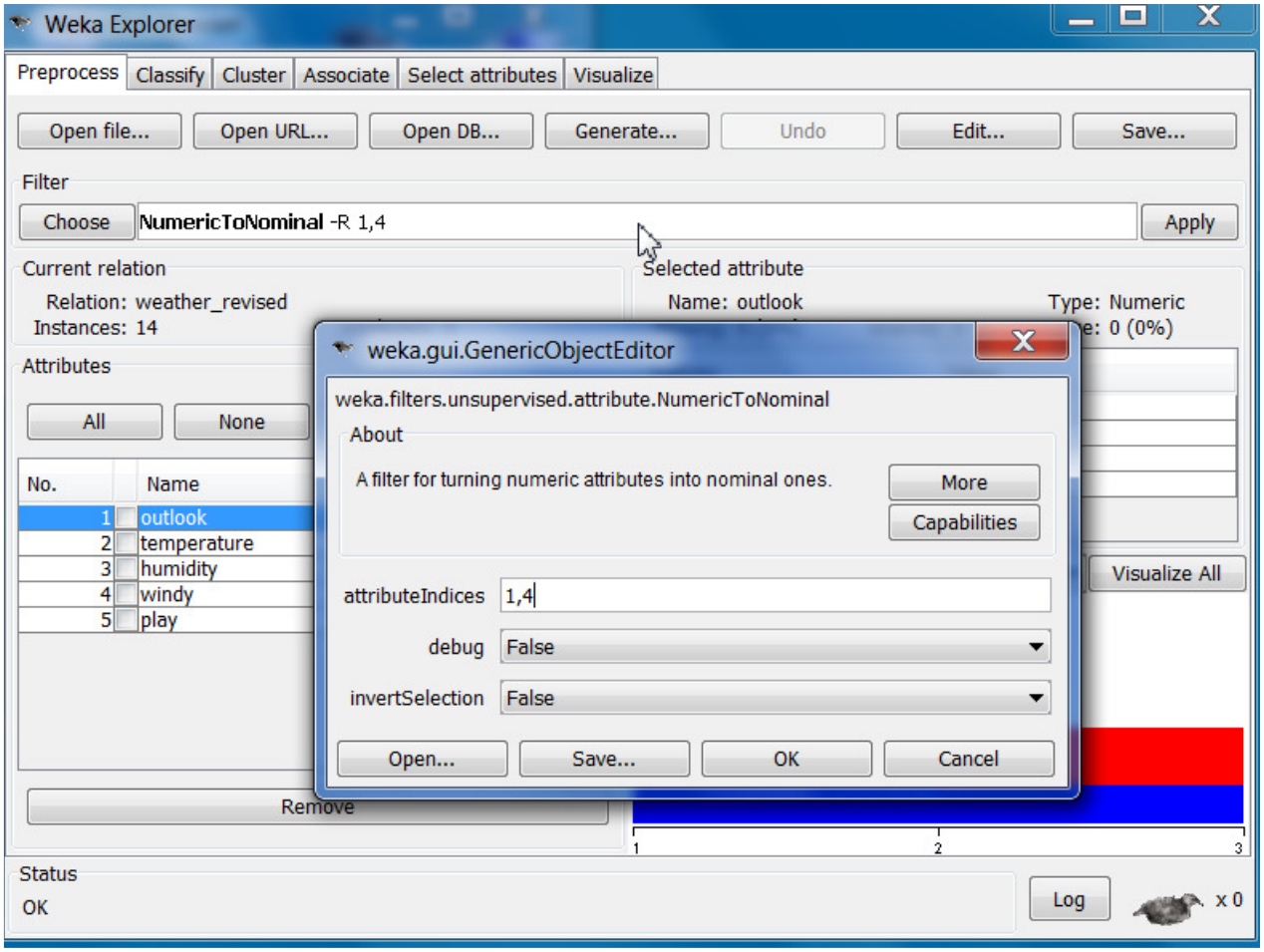


Figure 2. Editing the properties of NumericToNominal

Second, the numeric attributes *temperature* and *humidity* must also be converted to nominal. The procedure in this case is different from the one in the previous paragraph. One approach is to discretize these attributes by grouping their values into intervals. The filter that we use is called *Discretize.*

Select this filter by following the path: **Choose – filters – Unsupervised – Attribute – Discretize.**

Click on the selected object (**Figure 3**), in order to edit its properties (the settings of the Discretize filter) and set *attributeIndices* as 2,3 (No. of the attributes *temperature* and *humidity*). You may control the number of the derived intervals by setting the corresponding number in *Bins* (set it equal to 2). Also, you may force the intervals to have approximately the same number of values (useEqualFrequency=True).

Click on **OK** and then on **Apply** button. Check the type and the statistics of the transformed attributes.

At this phase, all attributes are nominal, and thus, we may proceed to the generation of the association rules.

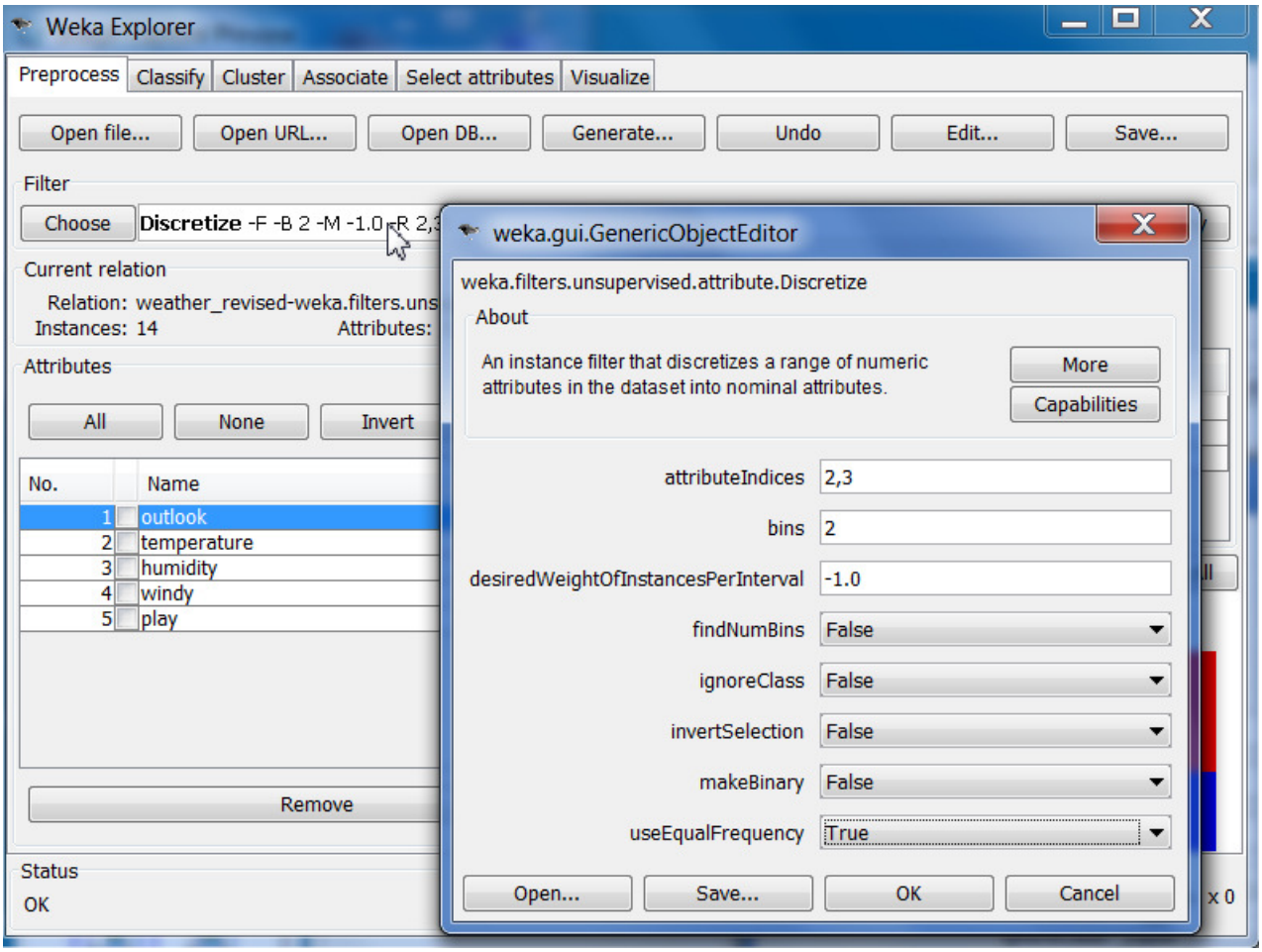


Figure 3. Editing the properties of Discretize

D. Generating Association Rules (applying the Apriori algorithm)

We select the **Associate** tab and **Choose** the Apriori algorithm.

Click on the selected object (**Figure 4**), in order to edit its properties (the settings of the Apriori algorithm). Suppose that we desire to find at most the first 20 association rules (**numRules**) with minimum support (**lowerBoundMinSupport**) set to 20% and minimum confidence (**minMetric**) set to 70%. Note that by setting the **metricType** to **Confidence**, the derived rules will be presented in decreasing order with respect to the value of confidence. Also, the frequent itemsets will appear in the results by setting **outputItemSets** = True. Leave the other settings as they are.

Click on **OK** button, and then click on **Start** button.

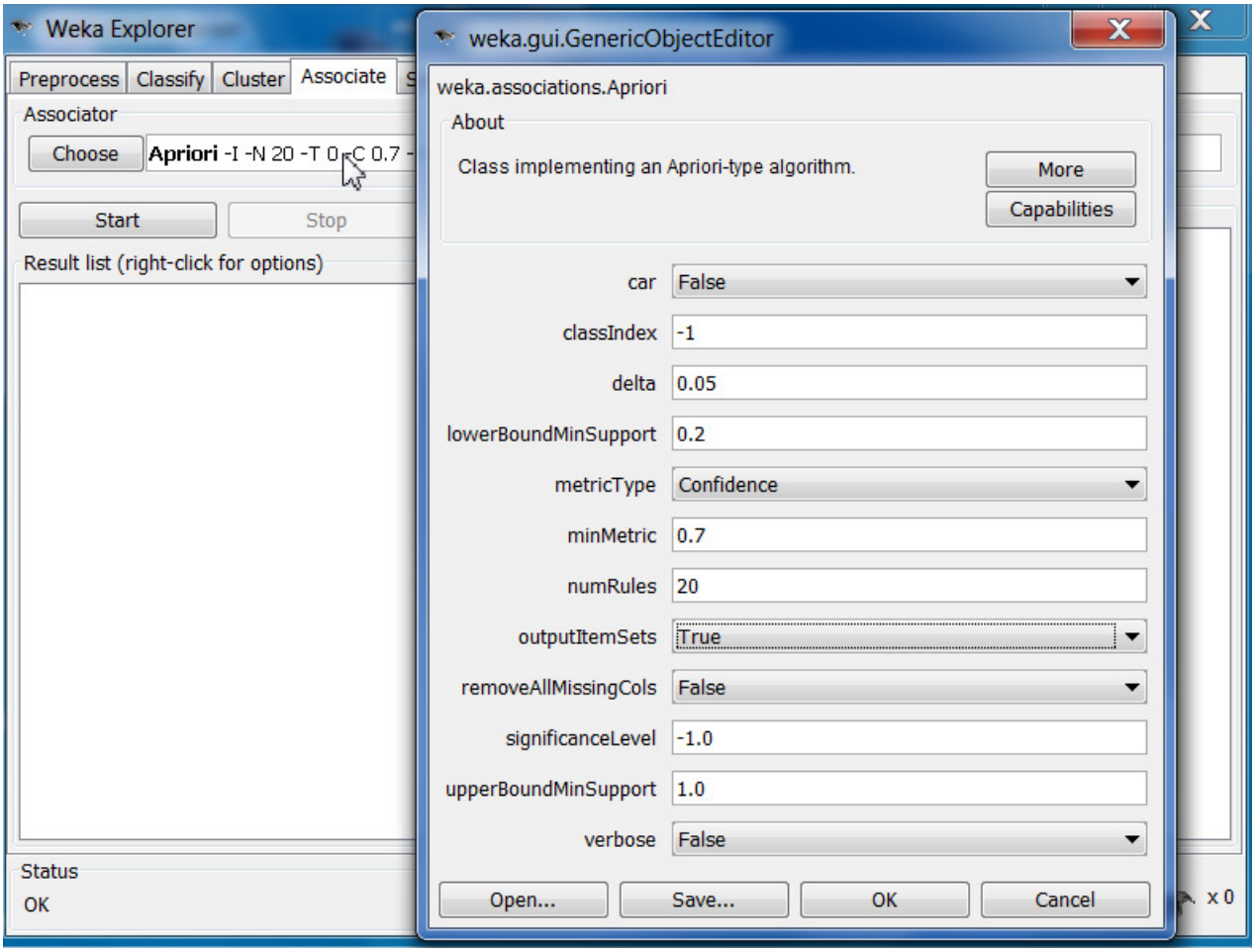


Figure 4. Editing the properties of the Apriori algorithm

In **Figure 5**, the generated rules are presented. As we can see, the algorithm has produced 17 rules that satisfy the requirements. Note that although we asked for at most 20 rules to be generated, the algorithm returned only 17 rules. Obviously, this means that all rules that satisfy the required conditions are presented in this list.

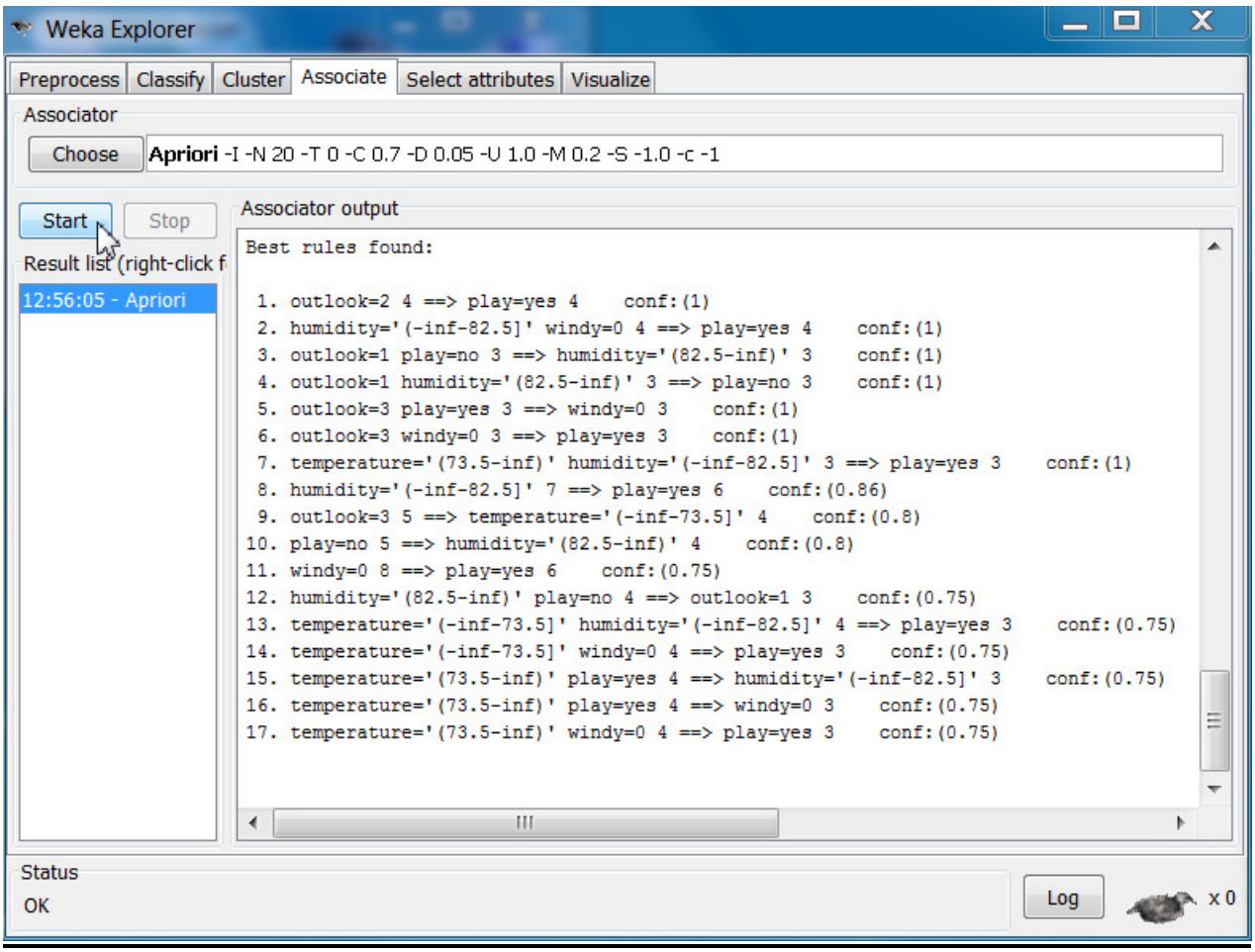


Figure 5. The generated association rules

E. Interpretation of the Generated Association Rules

The last rule (No. 17) in the list of the previous section is the following:

17. *temperature='(73.5-inf)' windy=0 4 ==> play=yes 3* conf:(0.75)

This rule states that:

When the temperature is above 73.5 degrees and it is not windy, then the game is held.

The number (4) at the left side of the rule indicates that there are 4 instances overall (out of 14) with *temperature='(73.5-inf)'* AND *windy=0*.

The number (3) at the right side of the rule indicates that, there are 3 instances with *play=yes*, out of the 4 instances with *temperature='(73.5-inf)' windy=0*. In other words, there are 3 instances with *temperature='(73.5-inf)'* AND *windy=0* AND *play=yes*, out of the 14 instances (whole dataset).

F. Calculation and Interpretation of the Basic Metrics

Considering the observations from the previous section, we may calculate (by hand) the basic metrics for this rule (No. 17), and provide their appropriate interpretations.

Support $\left(\text{temperature} = (73.5 - \text{inf})\ \text{windy=0}\ \text{==>}\ \text{play} = \text{yes}\right)$ =

$$= \frac{\left|\{\ \text{temperature} = (73.5 - \text{inf}), \text{windy} = 0,\ \text{play} = \text{yes}\}\ \right|}{\left|\text{dataset}\right|} = \frac{3}{14} = 0.2143$$

In 21.43% of the days, the temperature is above 73.5 degrees, it is not windy and the game is held.

Confidence $\left(\text{temperature} = (73.5 - \text{inf})\ ,\ \text{windy=0} \text{==>} \text{play} = \text{yes}\right) =$

$$= \frac{\left|\{\ \text{temperature} = (73.5 - \text{inf}), \text{windy} = 0, \text{play} = \text{yes}\}\right|}{\left|\{\text{temperature} = (73.5 - \text{inf}), \text{windy} = 0\}\right|} = \frac{3}{4} = 0.75$$

In 75% of the days with temperature above 73.5 degrees and no wind, the game is held.

Lift $\left(\text{temperature} = (73.5 - \text{inf}), \text{windy} = 0 \text{==>} \text{play} = \text{yes}\right) =$

$$= \frac{\textbf{Confidence}\ (\text{temperature} = (73.5 - \text{inf}), \text{windy} = 0 \text{==>} \text{play} = \text{yes})}{\textbf{Support}\{\text{play=yes}\}} = \frac{0.75}{9/14} = 1.17$$

One way to find the **Support** of { play=yes} is to look over the frequent itemsets L(1), which are shown in the same output (above the rules). We notice that there are 9 instances out of 14 with play=yes. Try to find/verify the value of Lift for this rule by using WEKA.

When we know that the temperature is above 73.5 degrees and it is not windy, the chance of the game to be held 1.17 times higher than in the case where we do not know anything about the temperature and windy.

G. Generating "Useful" Association Rules

According to the objective of this case study, we need to determine the suitable weather conditions under which the game is more likely to be held. Therefore, we are interested in those rules that have "play = yes" in the right side of a rule. In order to force the algorithm to generate association rules with the attribute *play* to be the only one to appear at the right side of a rule, we have to declare it into the properties of the Apriori algorithm.

In order to do so, we must set the parameter **car** (classification association rules) equal to **TRUE**, as in **Figure 6**. In addition to that, set **minMetric=0.9**.

Click on the **OK** button, and then click on the **Start** button.

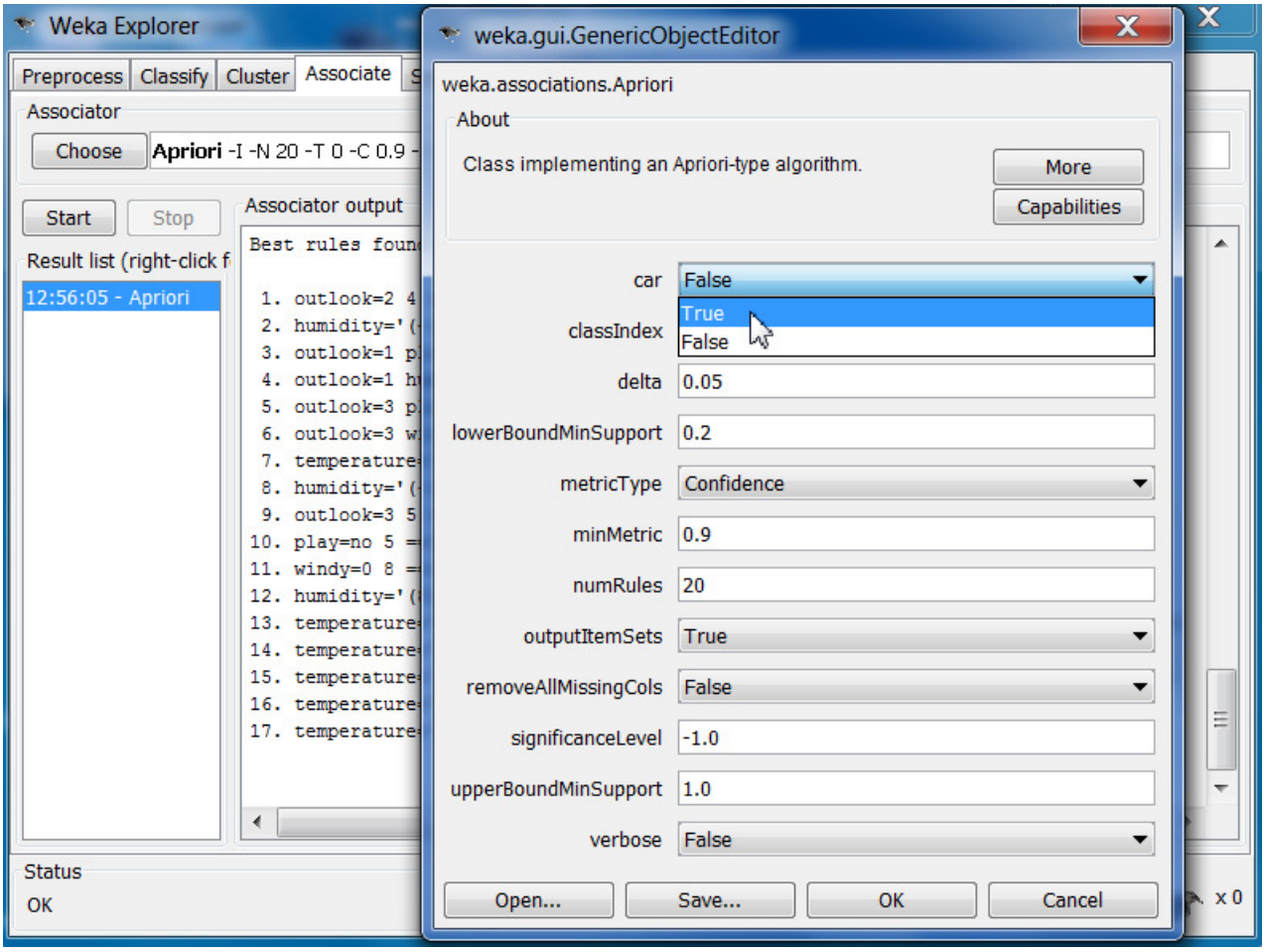


Figure 6. Generate rules with a classification attribute

The generated rules are the following:

1. outlook=2 4 ==> play=yes 4 conf:(1)
2. humidity='(-inf-82.5]' windy=0 4 ==> play=yes 4 conf:(1)
3. outlook=1 humidity='(82.5-inf)' 3 ==> play=no 3 conf:(1)
4. outlook=3 windy=0 3 ==> play=yes 3 conf:(1)
5. temperature='(73.5-inf)' humidity='(-inf-82.5]' 3 ==> play=yes 3 conf:(1)

We are interested in the rules that have *play=yes* on the right side, that is:

1. outlook=2 4 ==> play=yes 4 conf:(1)
2. humidity='(-inf-82.5]' windy=0 4 ==> play=yes 4 conf:(1)
4. outlook=3 windy=0 3 ==> play=yes 3 conf:(1)
5. temperature='(73.5-inf)' humidity='(-inf-82.5]' 3 ==> play=yes 3 conf:(1)

According to the above results, we may state the following rules with respect to the objective of this case study:

- The game has more chances to be held when the outlook is overcast.
- The game has more chances to be held when the humidity is below 82.5% and it is not windy.
- The game has more chances to be held when the outlook is rainy and it is not windy.
- The game has more chances to be held when the temperature is above 73.5 degrees and the humidity is below 82.5%.

All of these rules have the same confidence (1), whereas the last 2 rules have lower support (3/14) than the first two rules (4/14). Would it be useful to calculate the value of Lift for each rule in order to compare their usefulness?

<<<<<< End of text >>>>>>