**Practical No 4**

**4.1- Introduction to weka**

**4.2 - Classification using WEKA**

**4.3 - Implementation of Apriori Algo using weka**

**4.1) INTRODUCTION TO WEKA.**

Weka is a collection of machine learning algorithms for data mining tasks. Weka contains tools for data pre- processing, classification, regression, clustering, association rules, and visualization.

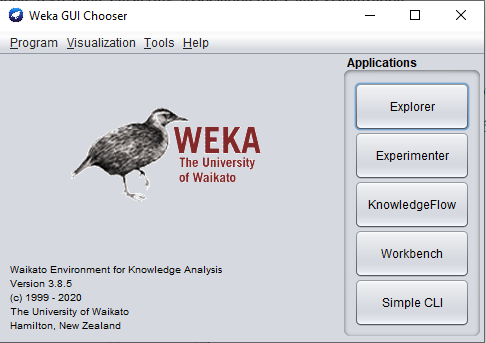
**The buttons can be used to start the following applications:**

**Explorer:** An environment for exploring data with WEKA.

**Experimenter:** An environment for performing experiments and conducting statistical tests between learning schemes.

**Knowledge Flow:** This environment supports essentially the same functions as the Explorer but with a drag-and-drop interface. One advantage is that it supports incremental learning. **Simple CLI:** Provides a simple command-line interface that allows direct execution of WEKA

commands for operating systems that do not provide their own command line interface.



1. Pre-process: It is used to choose and modify the data.

2. Classify: It is used to apply classification algorithms.

3. Cluster: Through this option we can learn different clustering algos for data.

4. Associate: It helps us to learn association rules for data.

5. Select Attributes: It helps us to select most relevant attributes in the data.

6. Visualize: It helps us to view the interactive 2D plot of the data.

**Loading Data**

The first three buttons at the top of the pre-process section enable us to load data into WEKA:

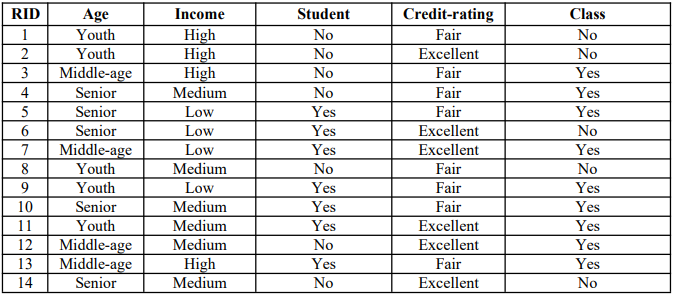
1. Open file.... Brings up a dialog box allowing us to browse for the data file on the local file system.

2. Open URL.... Asks for a Uniform Resource Locator address for where the data is stored.

3. Open DB.... Reads data from a database. (Note that to make this work you might have to edit the file in weka/experiment/DatabaseUtils.props.)

**4.2) IMPLEMENTATION OF CLASSIFICATION TECHNIQUES.**

A) Create an. ARFF (Attribute Relation File Format) file and construct decision tree for the following data.



**Step 1) Creating ARFF file: Open Notepad and type the following code and save it as electronics. arff.**

@relation electronics

@attribute age{youth,middle\_age,senior}

@attribute income{high,medium,low}

@attribute student{yes,no}

@attribute credit{fair,excellent}

@attribute class{yes,no}

@data

youth,high,no,fair,no

youth,high,no,excellent,no

middle\_age,high,no,fair,yes

senior,medium,no,fair,yes

senior,low,yes,fair,yes

senior,low,yes,excellent,no

middle\_age,low,yes,excellent,yes

youth,medium,no,fair,no

youth,low,yes,fair,yes

senior,medium,yes,fair,yes

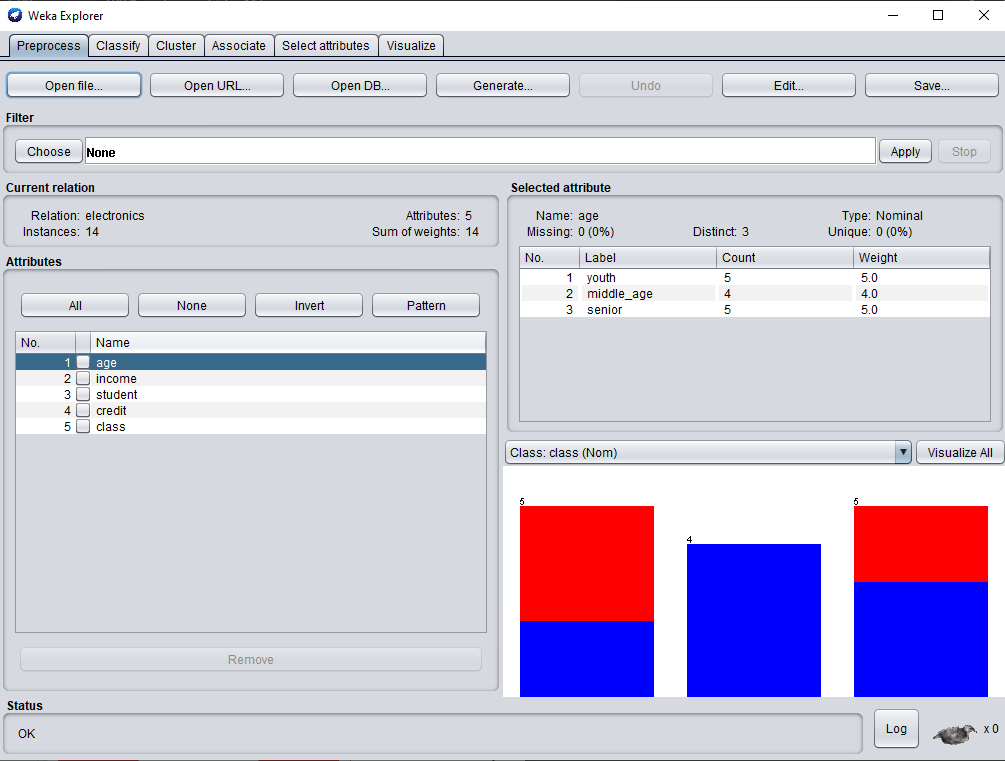
youth,medium,yes,excellent,yes

middle\_age,medium,no,excellent,yes

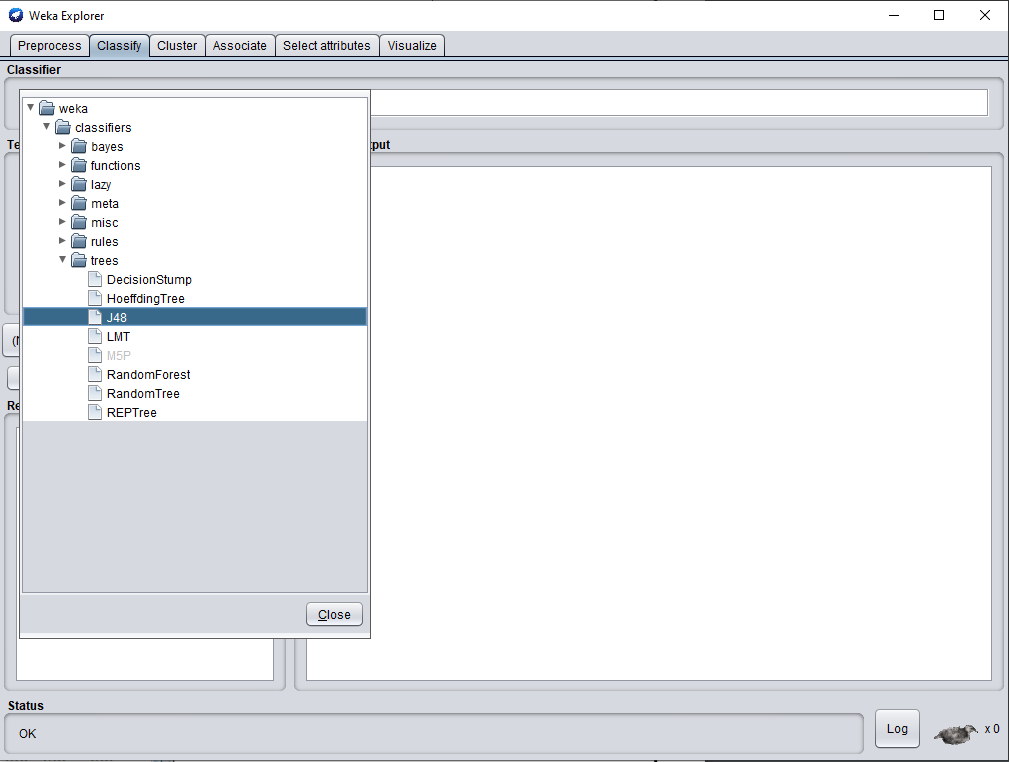
middle\_age,high,yes,fair,yes

senior,medium,no,excellent,no

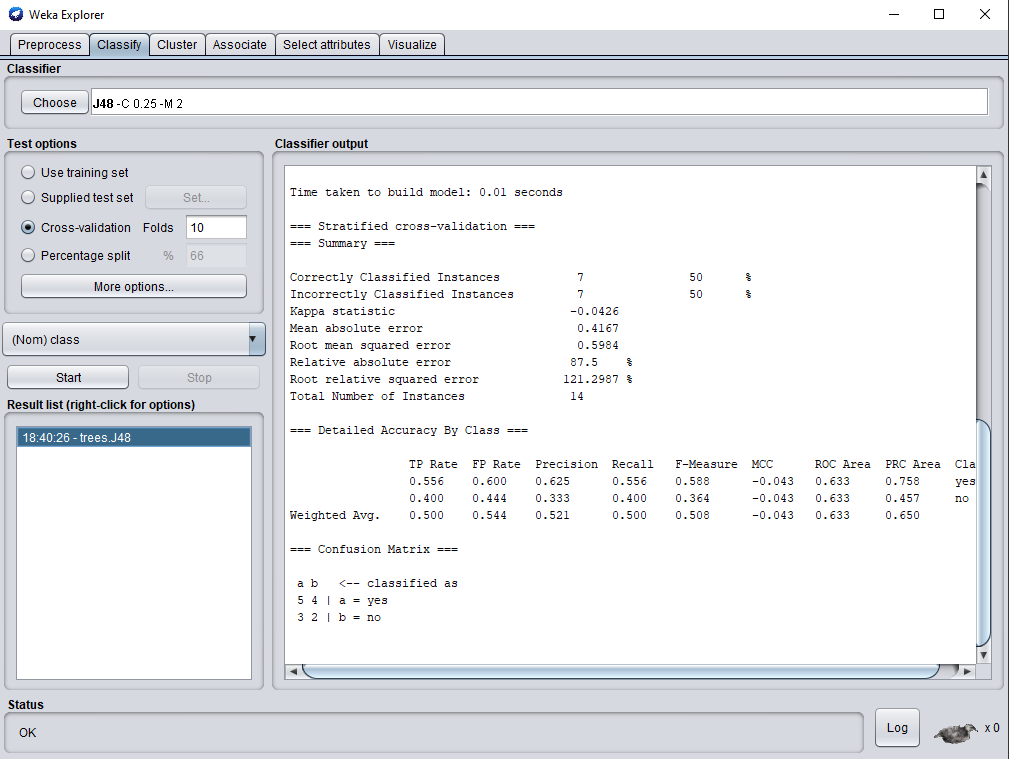
**Step 2) Open this file in WEKA by clicking on open file button.**



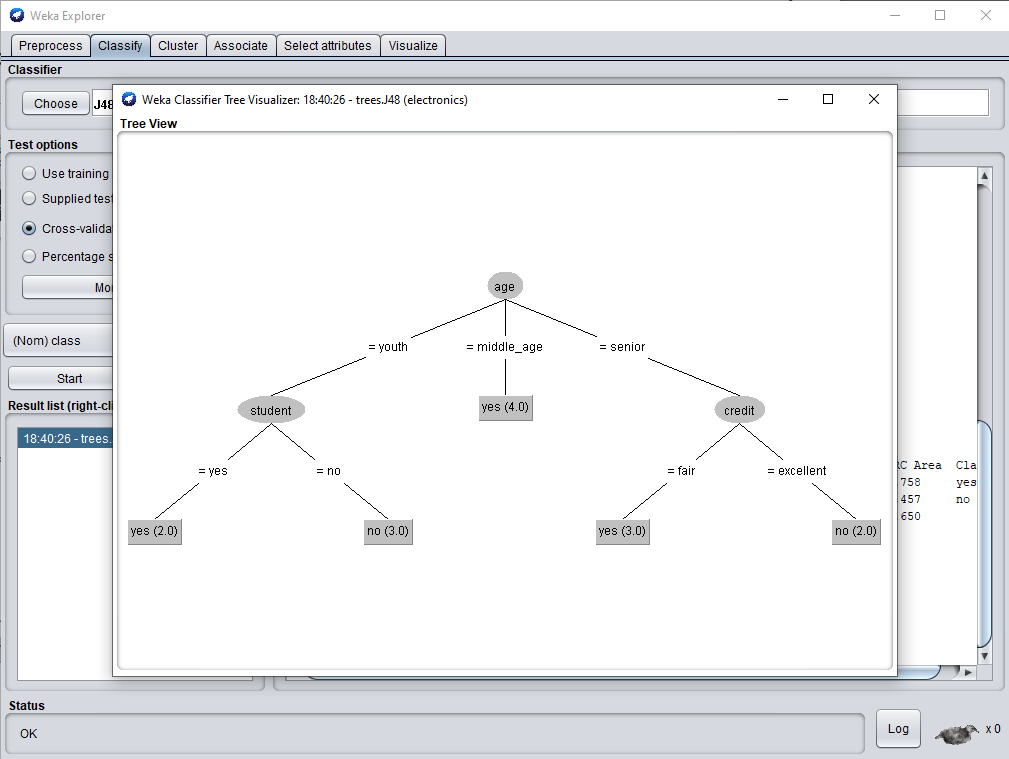
**Step 3) click on classify tab and then click on choose file and under tree select J48.**



**Step 4) Click on Start.**

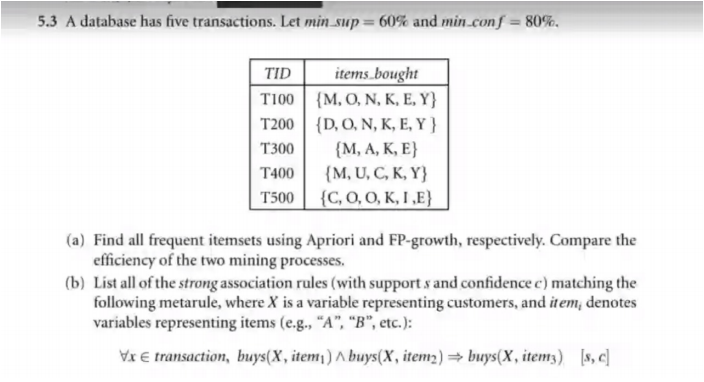


**Step 5) Right Click on J48 tree and select visualize tree.**



**4.3) Implementation of Apriori Algo using weka**

**Q1**



**Data file:**

@relation ItemDoc

@attribute A {TRUE, FALSE}

@attribute C {TRUE, FALSE}

@attribute D {TRUE, FALSE}

@attribute E {TRUE, FALSE}

@attribute I {TRUE, FALSE}

@attribute K {TRUE, FALSE}

@attribute M {TRUE, FALSE}

@attribute N {TRUE, FALSE}

@attribute O {TRUE, FALSE}

@attribute U {TRUE, FALSE}

@attribute Y {TRUE, FALSE}

@data

FALSE,FALSE,FALSE,TRUE,FALSE,TRUE,TRUE,TRUE,TRUE,FALSE,TRUE

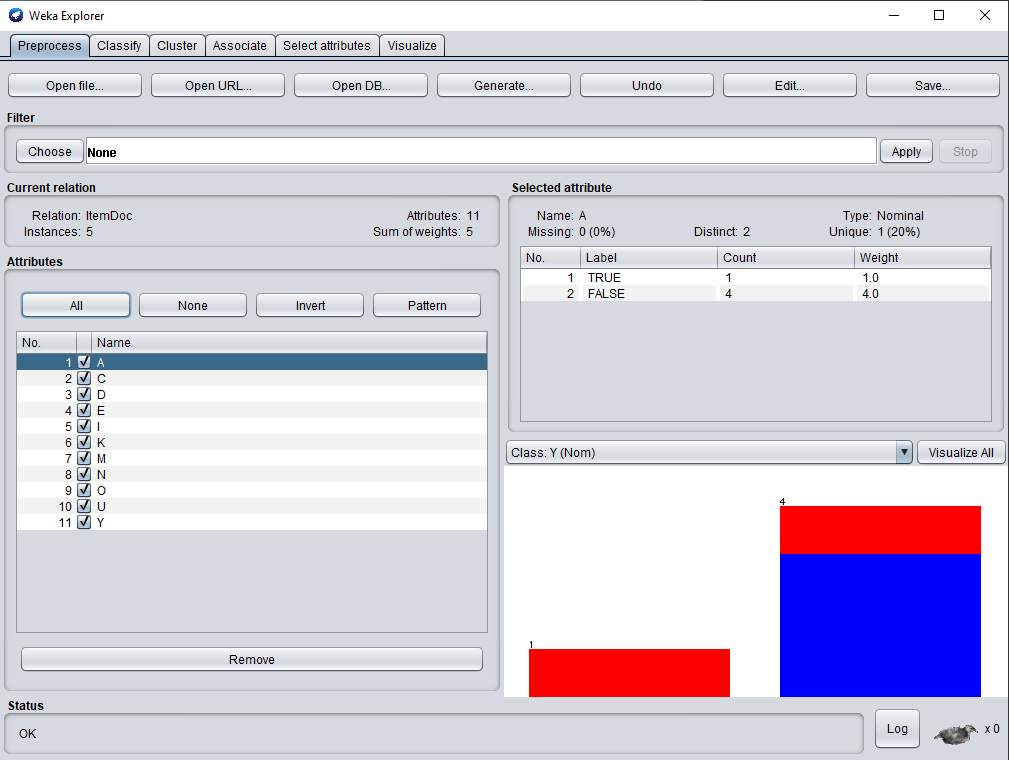
FALSE,FALSE,TRUE,TRUE,FALSE,TRUE,FALSE,TRUE,TRUE,FALSE,TRUE

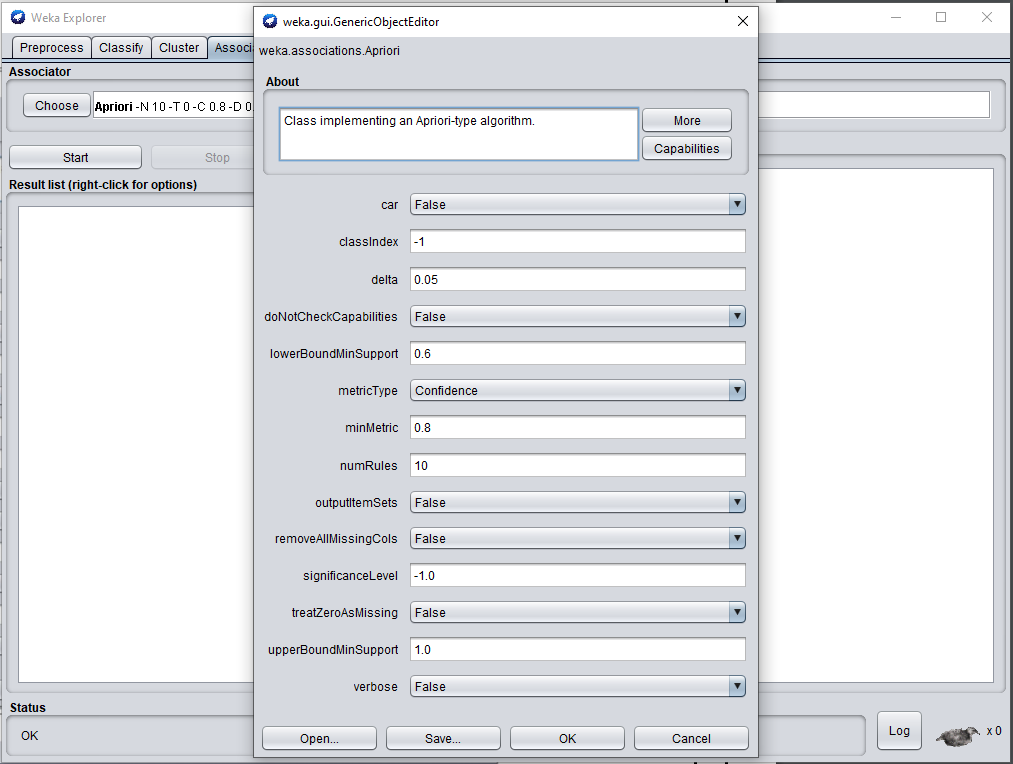
TRUE,FALSE,FALSE,TRUE,FALSE,TRUE,TRUE,FALSE,FALSE,FALSE,FALSE

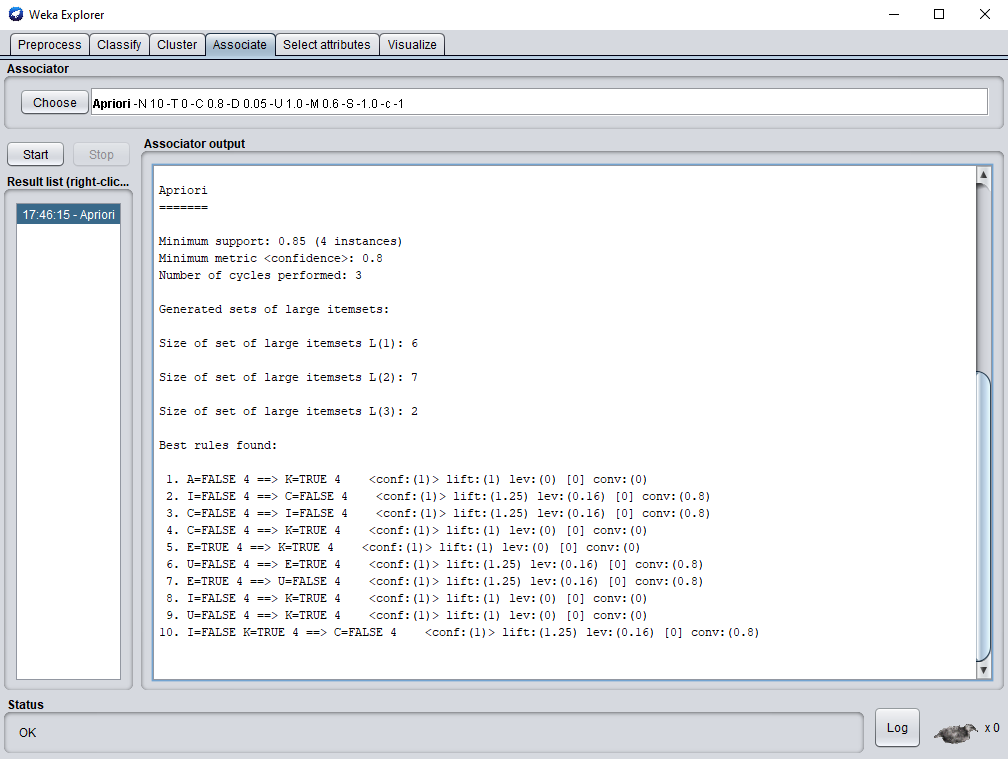
FALSE,FALSE,TRUE,FALSE,FALSE,TRUE,TRUE,FALSE,FALSE,TRUE,TRUE

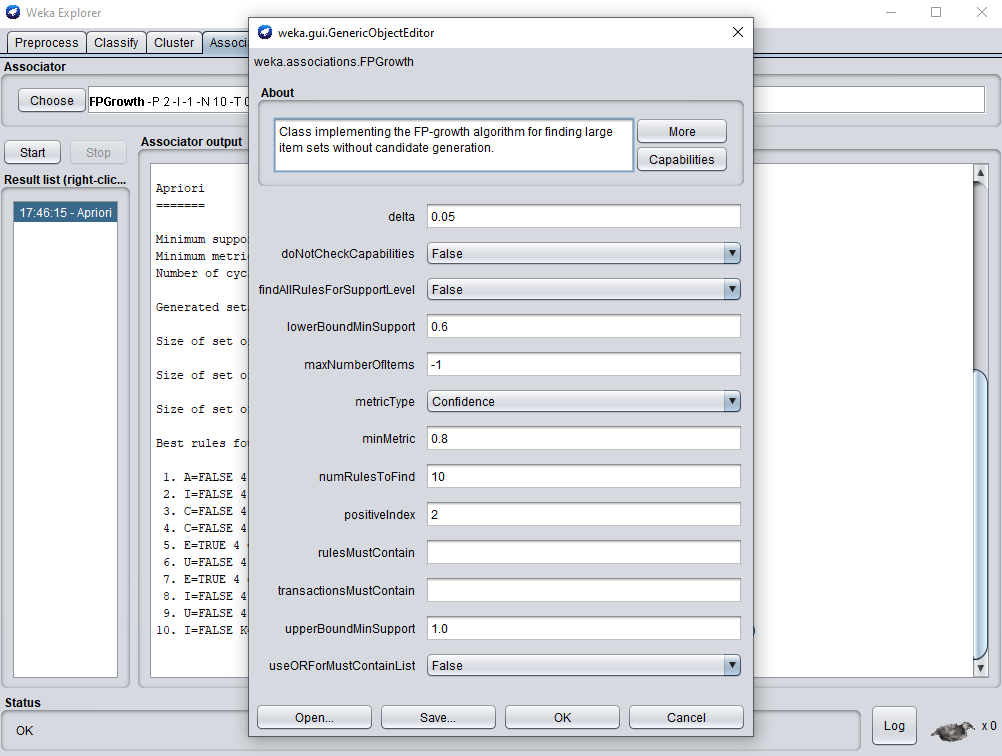
FALSE,TRUE,FALSE,TRUE,TRUE,TRUE,FALSE,FALSE,TRUE,FALSE,FALSE

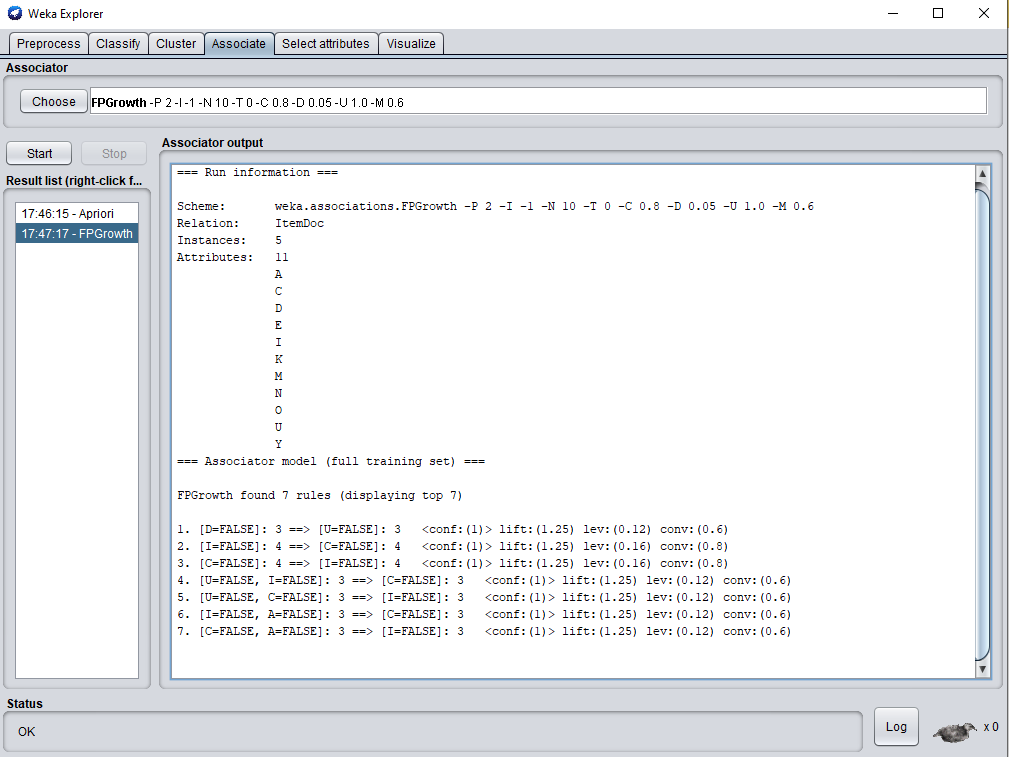
**Output:**



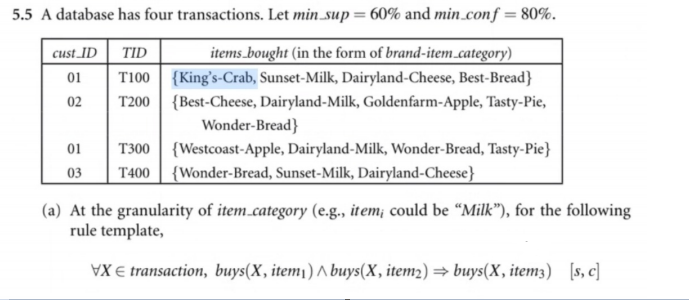








**Q2.**



**Data file:**

@relation ItemDoc1

@attribute A {TRUE, FALSE}

@attribute B {TRUE, FALSE}

@attribute C {TRUE, FALSE}

@attribute D {TRUE, FALSE}

@attribute E {TRUE, FALSE}

@attribute F {TRUE, FALSE}

@data

100,TRUE,TRUE,TRUE,TRUE,FALSE,FALSE

200,FALSE,TRUE,TRUE,TRUE,TRUE,TRUE

300,FALSE,TRUE,TRUE,FALSE,TRUE,TRUE

400,FALSE,TRUE,TRUE,FALSE,TRUE,FALSE

**Output:**

