



BIDUSA
Big Data Unites Sciences and Arts



Cofinanciado por
la Unión Europea



Convert data into arff file Session 3

In this session, we will learn how to create an arff file from a spreadsheet.



Erasmus+

Enriqueciendo vidas, abriendo mentes.

A.- ORIGIN POINT

We start from the basis of having our data in the Google spreadsheet, which we left done at the end of the previous session.

	A	B	C	D	E	F	G
1	Timestamp	Score	Select your course or occ	Insert your current age w	What is your gender?	1. How many deputies ar	2. Which of these prograi
2	28/03/2024 11:47:17	1 / 20	1ESO	13	Female	100	Galileo
3	28/03/2024 11:48:20	4 / 20	2ESO	15	Male	705	Erasmus+
4	28/03/2024 11:49:38	4 / 20	1ESO	15	Nonbinary	751	Eurovisión
5	28/03/2024 11:50:30	3 / 20	1ESO	13	Male	775	Frontex
6	28/03/2024 11:51:42	11 / 20	Teacher	53	Male	751	Eurovisión

B.- DATA PROCESSING AND DOWNLOAD

In Google's own spreadsheet we are going to process the data according to our needs with the following steps. It is important to check how many columns we have in the form. According to the Spanish form, that we are using, we have from column A to column Y, that is, 25 columns. 20 questions plus date, score, course or occupation, age and gender. We will eliminate the Timestamp column, so we will finally have 24 columns and, in the case of Spain, only 12 of them will have a score.

Our goal is to enclose all data in single quotes to convert it to the string data type and avoid problems with whitespace, accents, and control characters. We will do this in a new spreadsheet.

1. We will move the Score column to the last position, column Y.
2. Let's create a new spreadsheet in the plus icon in the lower left area.
3. In the new spreadsheet we go to cell **A1** and we will write the following function `""&'Form responses 1'!B1&""` to copy the data from cell **B1** and enclose it in single quotes.
4. We copy and drag the relative reference A1 so that it does the same with the rest of the cells up to column X, one less than Y since Timestamp does not interest us, and up to as many records as we have. It should be like this.

	A	B	C	D	E	F	G	H	I	J
1	'Select your course or occupation'	'Insert your curre	'What is your ger'	'1. How many de	'2. Which of thes	'3. How many st	'4. Which of the	'5. How many of	'6. What is the E	'7. Which EU cot
2	'1ESO'	'13'	'Female'	'100'	'Galileo'	'6'	'Mark Zuckerber	'One (English)	'Things can only	'Slovenia'
3	'2ESO'	'15'	'Male'	'705'	'Erasmus+'	'12'	'Bono'	'Three (English,	'Ode to joy'	'Germany'
4	'1ESO'	'15'	'Nonbinary'	'751'	'Eurovisión'	'28'	'Pope Francisco'	'Ten'	'We are the char	'France'
5	'1ESO'	'13'	'Male'	'775'	'Frontex'	'50'	'All of them'	'Twenty-four'	'One nation und	'Greece'
6	'Teacher'	'53'	'Male'	'751'	'Eurovisión'	'12'	'All of them'	'Twenty-four'	'Ode to joy'	'Germany'

5. Now we can only download this new spreadsheet by clicking on **File/Download/Comma-separated values (csv)**



Excel xlsx format download: possible problem with Score

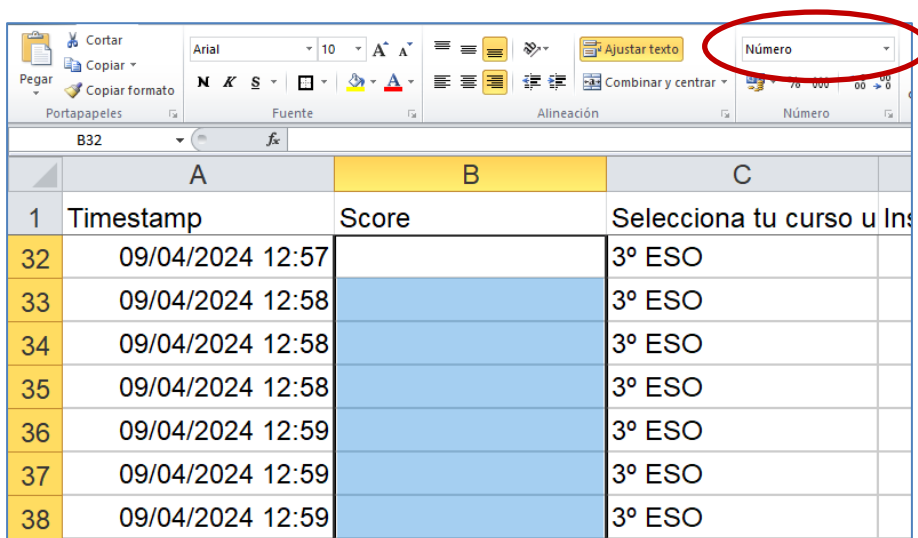
It is common that when downloading date-like columns in xlsx format, such as Score, it changes them to date format. To solve it we will follow the following steps:

A.- We create a **new sheet** and paste only the data to be converted into it.

B.- In an adjacent column we are going to use the **DAY** function, in Spanish **DIA**, which allows us to extract the day from a date, in our case the score, and converts it to a number in the first cell. Then we drag to do it in all of them. We see that it is already a number since it aligns it to the right.

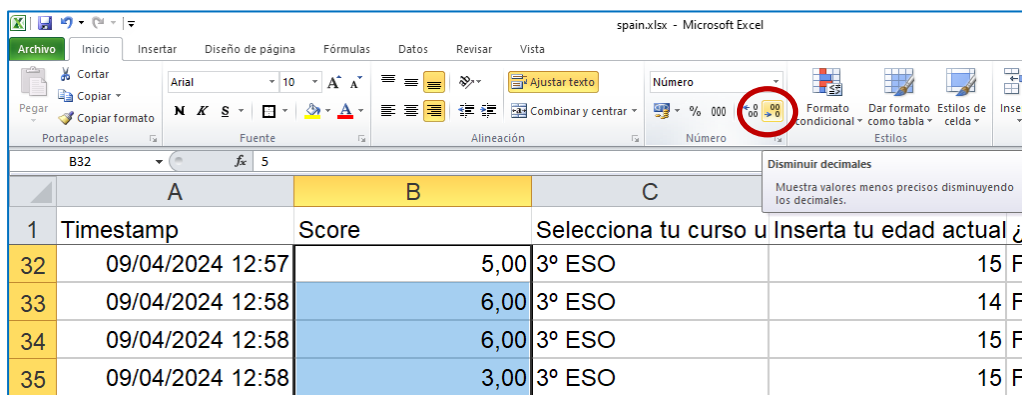
	A	B
1	05-dic	=DIA(A1)
2	06-dic	DIA(núm.de.serie)
3	06-dic	6
4	03-dic	3
5	05-dic	5
6	07-dic	7
7	05-dic	5

C.- Now we will go to our original sheet, we select the range of the data, we delete it and put the number format.



	A	B	C
1	Timestamp	Score	Selecciona tu curso u Ins
32	09/04/2024 12:57		3º ESO
33	09/04/2024 12:58		3º ESO
34	09/04/2024 12:58		3º ESO
35	09/04/2024 12:58		3º ESO
36	09/04/2024 12:59		3º ESO
37	09/04/2024 12:59		3º ESO
38	09/04/2024 12:59		3º ESO

D.- Then we copy the range of formatted numbers and on the original sheet with the selected range, we select the **special paste** with the right button to click on **paste only values**. They will appear with decimals, we remove them in the indicated button.



	A	B	C
1	Timestamp	Score	Selecciona tu curso u Inserta tu edad actual ¿C
32	09/04/2024 12:57	5,00	3º ESO
33	09/04/2024 12:58	6,00	3º ESO
34	09/04/2024 12:58	6,00	3º ESO
35	09/04/2024 12:58	3,00	3º ESO

E.- If we wanted to keep the **Score / 12** format then we have to select the entire range of cells, click on the right button and then on **Cell Format**, within the **category** select **Custom** and overwrite the **Type** value with **0"/ 12 "** to keep the starting number, it would be 0, and add a literal with **"/12"**

	A	B	
1	Timestamp	Score	Seleccio
35	09/04/2024 12:58	3 / 12	3º ESO
36	09/04/2024 12:59	5 / 12	3º ESO
37	09/04/2024 12:59	7 / 12	3º ESO
38	09/04/2024 12:59	5 / 12	3º ESO
39	09/04/2024 12:59	4 / 12	3º ESO
40	09/04/2024 12:59	3 / 12	3º ESO
41	09/04/2024 12:59	3 / 12	3º ESO
42	09/04/2024 13:00	8 / 12	3º ESO
43	09/04/2024 13:00	4 / 12	3º ESO
44	09/04/2024 13:00	8 / 12	3º ESO

F.- Then we would delete the new sheet and we could convert to the csv format with Excel itself using **Save As** and selecting **CSV (Comma Separated Value) (*.csv)** and continuing with the steps on the following pages.

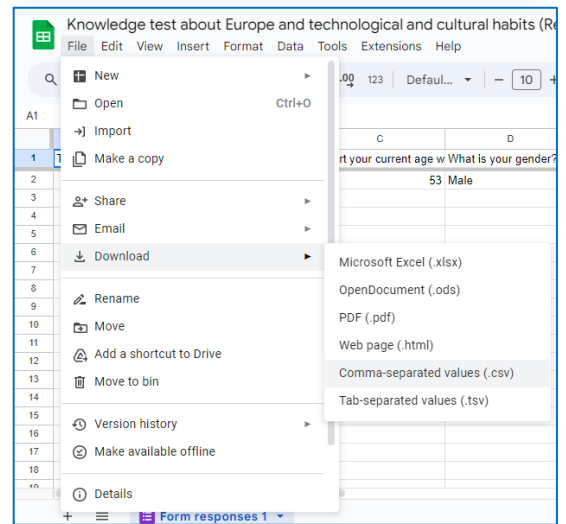
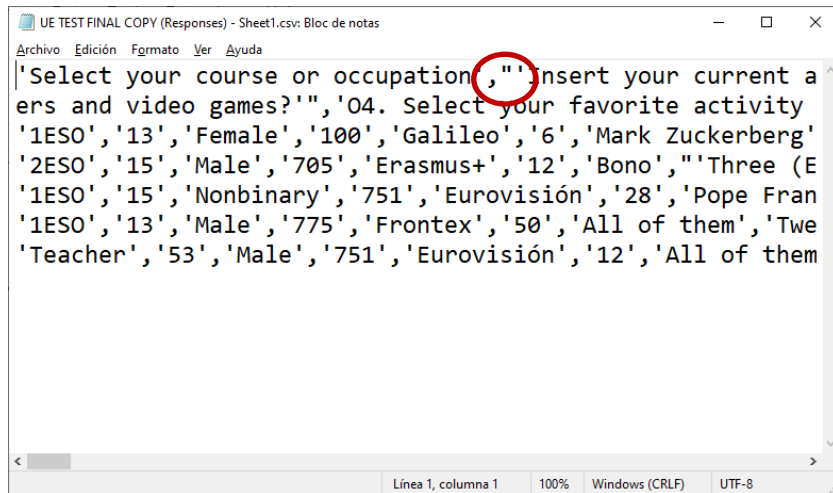


Absence of values: null is ?

This is not our case, but it is possible that in certain datasets we find the absence of null values. For example, if we do not know the age, occupation, etc. In these cases, Weka has a special character to indicate that there is a null, it is the **?**, which we must insert as is.

Once the data has been downloaded in csv format, we must associate the file format with the notepad. To do this, we will right-click and choose *Open with*, and select *Notepad*. This way, forever, all csv files will be opened directly with notepad.

Our data will appear as follows.



The data appears raw apparently, but within the data we have the following order,

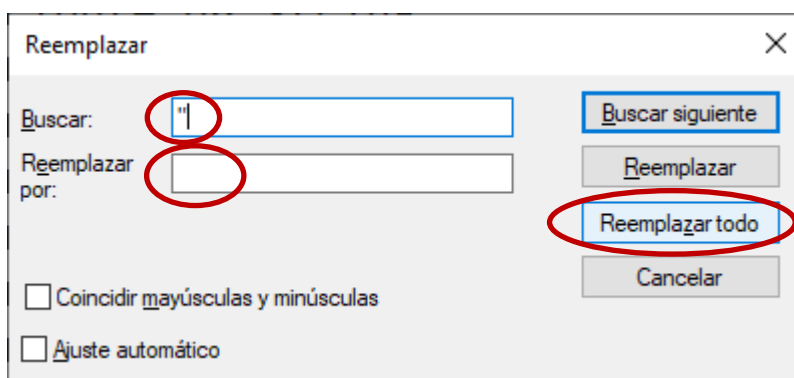
- ✓ The first row is the question headings, the *columns*.
- ✓ The following *rows* are the data itself, although we will only have a single row now.

Now we are ready to convert the csv file to arff format.



Be careful with double quotes

When downloading the file in csv format, it is very likely that some phrase will be enclosed in double quotes. Knowing that we do not have any double quotes that we want to keep in our data, we can then go to the top menu **Edit/Replace** and where we put double quotes we will put a blank space, then clicking on **Replace all**.



C.- CONVERT INTO ARFF FILE

Our goal in this section is to achieve the following structure compatible with arff files, which allows us to open the file with the Weka program.

```
*weather.numeric.arff: Bloc de notas
Archivo Edición Formato Ver Ayuda
@relation weather

@attribute outlook string
@attribute temperature numeric
@attribute humidity numeric
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
sunny,85,85,FALSE,no
sunny,80,90,TRUE,no
overcast,83,86,FALSE,yes
rainy,70,96,FALSE,yes

Línea 3, columna 26 100% UNIX (LF) UTF-8
```

The arff file is divided into 3 sections, each identified by a type of tag that always begins with @.

@relation name: every arff file must begin with this declaration in its first line, which will name the file.

@attribute name data_type: then we will include a line for each attribute or column that we have in our data set, indicating its name and the corresponding data type that it may be,

- ✓ **Numerical Attributes:** they can be real numbers, taking the point as a decimal separator, with numeric or real. We can also express whole numbers with integers. For example, *numeric, real, integer*.
- ✓ **Text attributes:** takes the string data type of the Java language for textual expressions. For example, *string*.
- ✓ **Date attributes:** we put date and, optionally, we will indicate the date format optionally, which can be of the type "yyyy-MM-dd HH:mm:ss", taking the Java parameters. For example, *date "yyyy-MM-dd"*
- ✓ **Nominal attributes:** These are data types defined by ourselves using a list of values that are separated by commas and enclosed in braces. They are the ones we should use whenever we have textual values. We can write them ourselves, if there are few values, or much better, let Weka do it with an attribute filter, we will see it in another session. For example, *{sunny, cloudy, rainy}*

@data: are the data records, each one on one line. We'll make sure that all rows have the same number of columns and that that number matches the number of **@attribute** statements we added earlier.



Exception to data types - A Priori Algorithm

Usually we must assign the correct data type to each attribute, however the A Priori algorithm only allows attributes that are textual and nominal. Therefore, we will put all the attributes as string.

We will follow the following steps for the conversion.

1. We will write the name of the file putting **@relation 'test EU'** as the first line
2. We will separate the different sections with carriage returns.
3. We will put each column in a row.

4. We must prepend **@attribute** in front of all columns and remove the final commas from each question.
5. We will change the text of the first 3 general questions about the course or occupation, age and gender to **course_occupation, age, gender**.
6. We must also simplify the names of the columns to avoid long names by **q1, q2, oq1, oq2, ...**
7. All columns must be textual due to **A Priori algorithm** so we will mark them as **string**.
8. We will put the **@data** section before all the data but before we must delete the row with the column names.

The result that we should obtain will be the following.

```

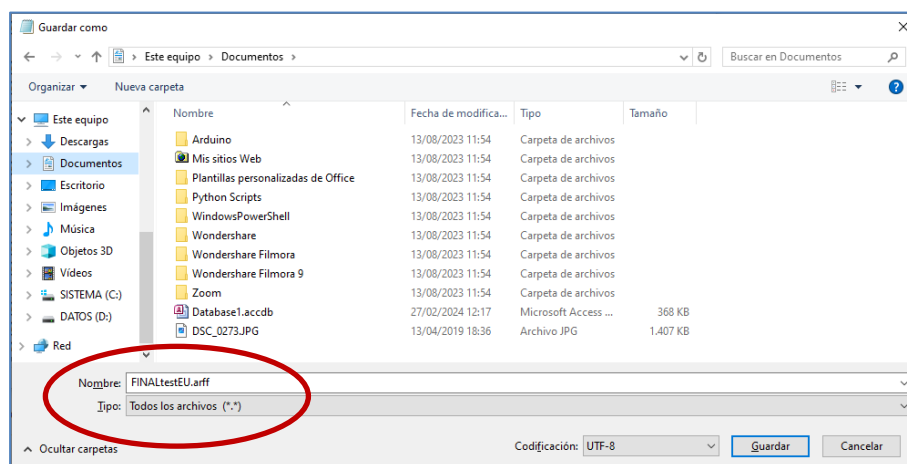
@relation 'test EU'

@attribute 'course_occupation' string
@attribute 'age' string
@attribute 'gender' string
@attribute q1 string
@attribute q2 string
@attribute q3 string
@attribute q4 string
@attribute q5 string
@attribute q6 string
@attribute q7 string
@attribute q8 string
@attribute q9 string
@attribute q10 string
@attribute q11 string
@attribute q12 string
@attribute oq1 string
@attribute oq2 string
@attribute oq3 string
@attribute oq4 string
@attribute oq5 string
@attribute oq6 string
@attribute oq7 string
@attribute oq8 string
@attribute 'Score' string

@data
'1ESO','13','Female','100','Galileo','6','Mark Zuckerberg','One (English)','Things can only get better','Slovenia','Türkiye','The Euro'

```

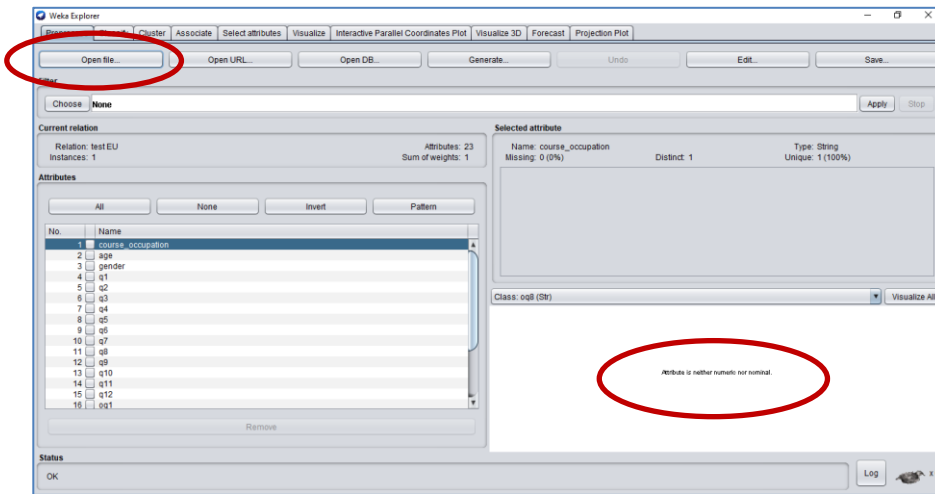
9. Now we have to save the file as arff. To do this we will go to **File/Save as...** and we will choose the name that we have with the arff extension, for example **finaltestEU.arff**, and in **type** we will put **All files (*.*)** and click on **Save**.



10. In the event that we have the test divided into several forms, we will only have to copy and paste the new data after the last existing ones since the columns will be the same.

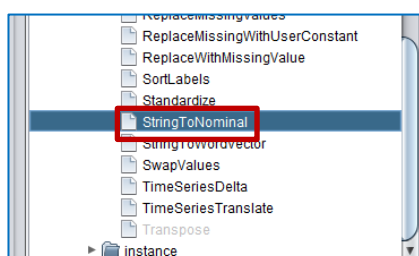
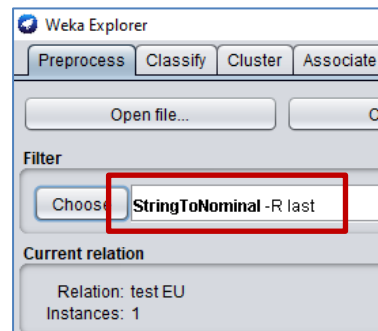
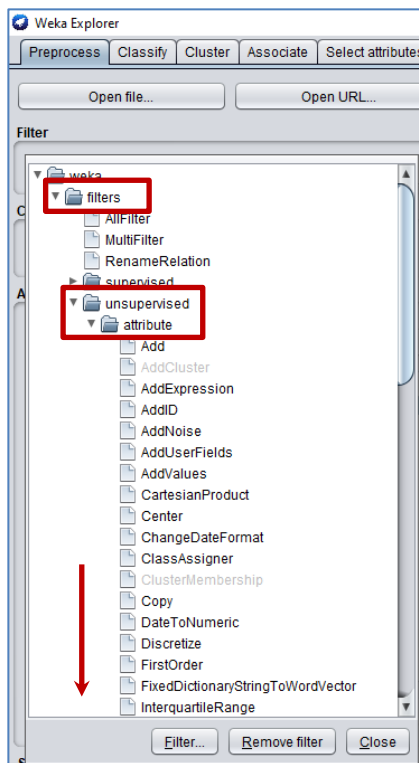
D.- OPEN ARFF FILE WITH WEKA

Once the file is finished, we will open Weka and press the *Explorer* button to enter into graphical interface. In the graphical interface we will load the path file where it was saved by clicking on the *Open File ...* button, and the image that will appear will be the following.

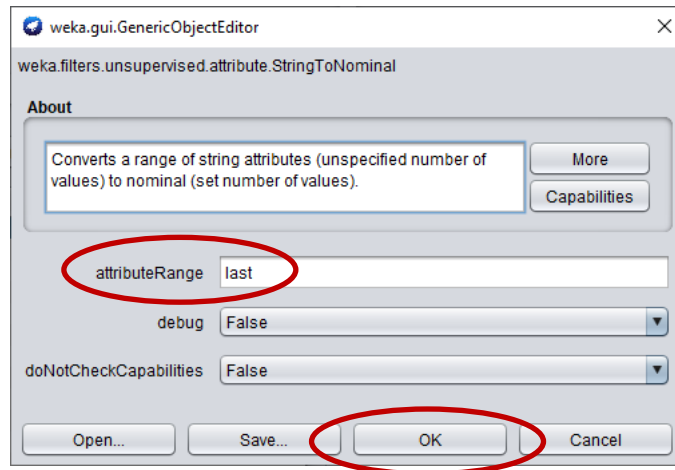


However, we do not get any histogram of each attribute. This is because attributes are required to be nominal, allowing only written values, and enclosed in curly braces. We will apply an attribute filter.

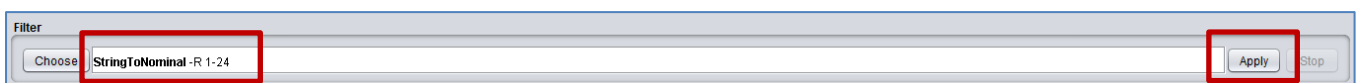
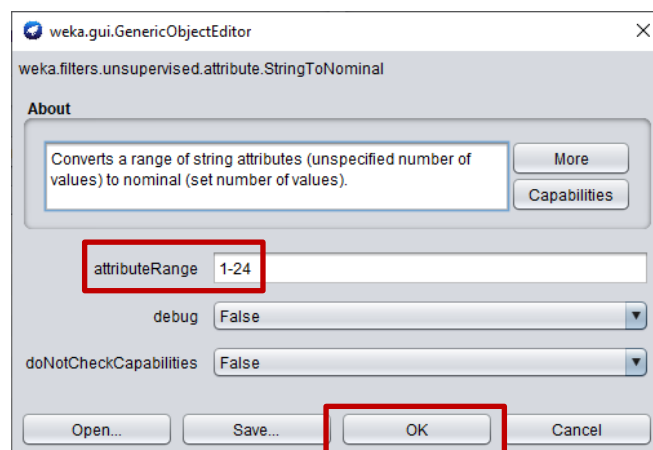
- ✓ In the same *Preprocess* tab we have *Filter* section we will press the *Choose* button and within the *filter/unsupervised/attribute* options to choose the *StringToNominal* filter, which should be written in the text box.



- ✓ Next we will click on the text box that says StringToNominal to configure the filter and the following window will appear.



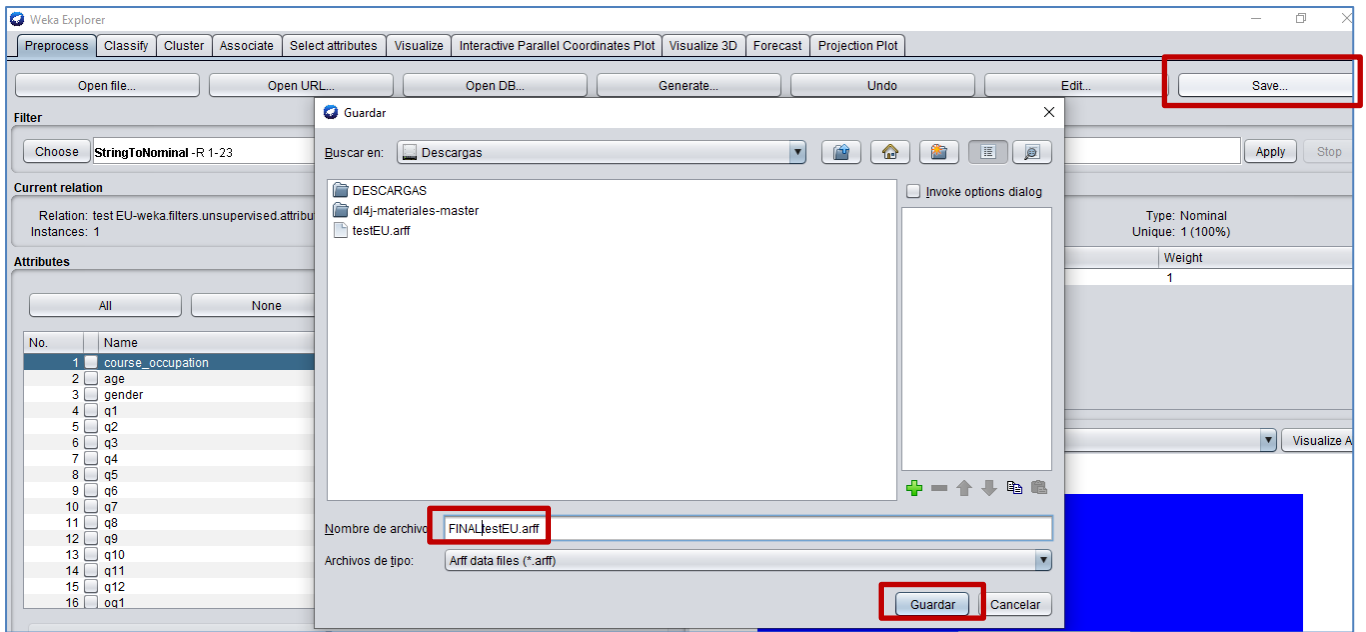
- ✓ In the *attributeRange* text box we must write the numbers of attributes that we want to nominalize. They are all the ones that we put as string in the arff file. Therefore, we must write in the text box **1-24** to indicate the interval between attribute 1 and 24, both inclusive. And then press the **OK** button. The filter will change to the image below.



- ✓ Then we will press the **Apply** button to the right of the filter and the histograms will appear.

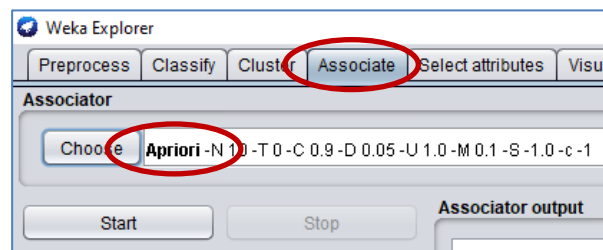
F.- SAVE DATASET

After making all the previous changes and before passing the algorithm, we must save the dataset with all the changes made. To do this, in the *Preprocess* tab we will click on the **Save...** button and we will save it with the name we want. This file will be our final dataset.

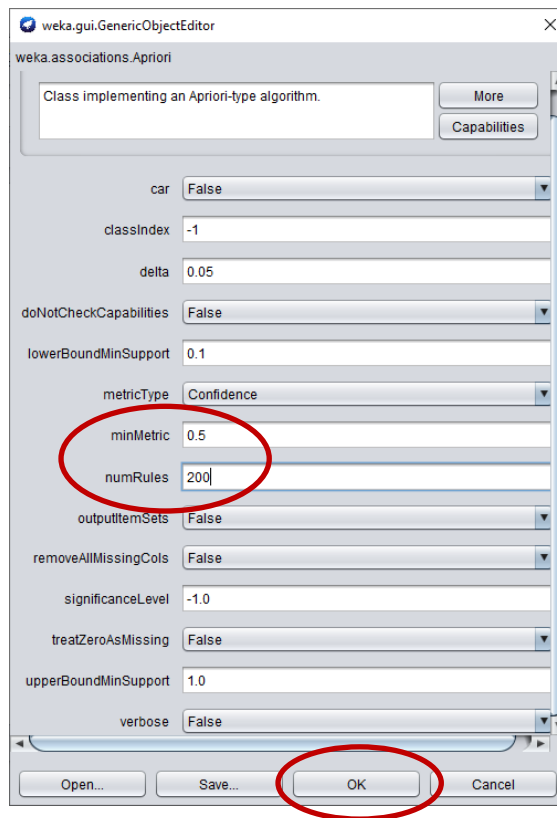


G.- A PRIORI ALGORITHM

Finally, we can now go to the *Associate* tab, choose the A Priori algorithm in the *Choose* button, click it in the name of text box and configure it.



- ✓ **minMetric**: we must indicate the minimum percentage, by one, that we want the rules it finds to comply with. For example, we will put 0.5.
- ✓ **numRules**: we indicate the number of rules that we want to appear. For example, we will put 200.



Then we click on **OK**, and we execute it on the **Start** button, we will automatically get 200 rules with a minimum of 50% probability, ordered by highest to lowest probability.

```
Best rules found:
1. age=13 2 ==> course_occupation=IESO 2 <conf:(1)> lift:(1.67) lev:(0.16) [0] conv:(0.8)
2. Score=4 2 ==> age=15 2 <conf:(1)> lift:(2.5) lev:(0.24) [1] conv:(1.2)
3. age=15 2 ==> Score=4 2 <conf:(1)> lift:(2.5) lev:(0.24) [1] conv:(1.2)
4. q3=12 2 ==> gender=Male 2 <conf:(1)> lift:(1.67) lev:(0.16) [0] conv:(0.8)
5. q4=All of them 2 ==> gender=Male 2 <conf:(1)> lift:(1.67) lev:(0.16) [0] conv:(0.8)
6. q5=Twenty-four 2 ==> gender=Male 2 <conf:(1)> lift:(1.67) lev:(0.16) [0] conv:(0.8)
7. q6=Ode to joy 2 ==> gender=Male 2 <conf:(1)> lift:(1.67) lev:(0.16) [0] conv:(0.8)
8. q7=Germany 2 ==> gender=Male 2 <conf:(1)> lift:(1.67) lev:(0.16) [0] conv:(0.8)
9. q12=The three previous countries 2 ==> gender=Male 2 <conf:(1)> lift:(1.67) lev:(0.16) [0] conv:(0.8)
10. oq2=None of them 2 ==> gender=Male 2 <conf:(1)> lift:(1.67) lev:(0.16) [0] conv:(0.8)
11. q2=EurovisiÃ³n 2 ==> q1=751 2 <conf:(1)> lift:(2.5) lev:(0.24) [1] conv:(1.2)
12. q1=751 2 ==> q2=EurovisiÃ³n 2 <conf:(1)> lift:(2.5) lev:(0.24) [1] conv:(1.2)
13. q8=Croatia 2 ==> q1=751 2 <conf:(1)> lift:(2.5) lev:(0.24) [1] conv:(1.2)
14. q1=751 2 ==> q8=Croatia 2 <conf:(1)> lift:(2.5) lev:(0.24) [1] conv:(1.2)
15. q9=The European Union as a whole 2 ==> q1=751 2 <conf:(1)> lift:(2.5) lev:(0.24) [1] conv:(1.2)
16. q1=751 2 ==> q9=The European Union as a whole 2 <conf:(1)> lift:(2.5) lev:(0.24) [1] conv:(1.2)
17. q10=Every five years 2 ==> q1=751 2 <conf:(1)> lift:(2.5) lev:(0.24) [1] conv:(1.2)
18. q1=751 2 ==> q10=Every five years 2 <conf:(1)> lift:(2.5) lev:(0.24) [1] conv:(1.2)
19. oq5=Laptop 2 ==> q1=751 2 <conf:(1)> lift:(2.5) lev:(0.24) [1] conv:(1.2)
20. q1=751 2 ==> oq5=Laptop 2 <conf:(1)> lift:(2.5) lev:(0.24) [1] conv:(1.2)
21. q8=Croatia 2 ==> q2=EurovisiÃ³n 2 <conf:(1)> lift:(2.5) lev:(0.24) [1] conv:(1.2)
22. q2=EurovisiÃ³n 2 ==> q8=Croatia 2 <conf:(1)> lift:(2.5) lev:(0.24) [1] conv:(1.2)
23. q9=The European Union as a whole 2 ==> q2=EurovisiÃ³n 2 <conf:(1)> lift:(2.5) lev:(0.24) [1] conv:(1.2)
24. q2=EurovisiÃ³n 2 ==> q9=The European Union as a whole 2 <conf:(1)> lift:(2.5) lev:(0.24) [1] conv:(1.2)
25. q10=Every five years 2 ==> q2=EurovisiÃ³n 2 <conf:(1)> lift:(2.5) lev:(0.24) [1] conv:(1.2)
```