AUTO-DataGenCARS
User Guide

Last update: 16/06/2021

Index

What is AUTO-DataGenCARS?

DataGenCARS is a complete Java-based synthetic dataset generator for the evaluation of both Context-Aware Recommendation Systems (*CARS*) and traditional Recommendation Systems (*RS*) to obtain the required datasets for any type of scenario desired. This tool allows a high flexibility in the generation of appropriate datasets for evaluating CARS.

The goal of this guide is to describe the main functionalities of *DataGenCARS*. Specifically, it explains the different functionalities, components and organization of the DataGenCARS application interface, named *AUTO-DataGenCARS* (*Advanced User orienTed tOol DataGenCARS*), the files that are used and how they are formed. This guide also shows examples of how to generate a dataset using this tool, thus demonstrating that it can be quite useful since there are very few data sets to evaluate CARS [1].

To run this tool, you will need to run the *AUTO-DataGenCARS* jar with this command:
java --add-opens=java.desktop/javax.swing=ALL-UNNAMED --illegal-access=deny -jar DGC.jar

1. Input and output data

DataGenCARS requires some input data to support the generation of the dataset. There exist three types of input files: scheme, generation and profile.

- The scheme files are composed by a list of *attributes* (name, data type and possible values) defining the different entities involved in the recommendation process, namely: *users, items and contexts*.

- The generation file contains parameters to configure the dataset generation process, like the number of ratings to generate or the minimum and maximum value of the ratings.

- There are two types of profiles, *user and item*. AUTO-DataGenCARS allows you to define the item profile as an attribute for the item scheme, however the user profile must be created in a different way.

  The user profile file contains identifiers of the user profiles and weights for each attribute defined (about the items and contexts). The sum of all the weights must equal one. However, DataGenCARS has implemented the automatic readjustment of the weights. The weights may have the associated symbols (+) or (-). The former indicates that the order of relevance of the attribute values starts with the furthest on the right [-…+], while the latter indicates the opposite [+…-]. Non-relevant attributes defined in the schemes have a value of 0.

user_profile.csv

```
id;director;movieCountry;time;daytype;season;location;weather;mood;other
1;(-) 0.1;(-) 0.3;(-) 0.4;0;0;(-) 0.1;0;(-) 0.1;0
2;(-) 0.2;(-) 0.3;(-) 0.2;0;0;(-) 0.1; (+) 0.2;0;0
3;0;(-) 0.3;(-) 0.3;0;(-) 0.2;0;0;(-) 0.2;0
4;0;(-) 0.3;(-) 0.3;(-) 0.1;0;0;(-) 0.1;(-) 0.2;0
5;0;(-) 0.4;0;(-) 0.1;0;(-) 0.3;(-) 0.2;0;0
```

Figure 1: Example of user profile file

AUTO-DataGenCARS interface allows the creation and modification of all these input data without the need to touch any external files.

DataGenCARS generates the output files *user.csv*, *item.csv*, *context.csv* and *ratings.csv* (see examples in the Figures 2, 3, 4 and 5, respectively). These files have a header with the names of the fields and the following lines represent the data generated for each field. Both the name of the fields and the data generated are separated by semicolons.

user.csv

```
userID;age;sex;city;country
1;30;F;Zaragoza;Spain
2;42;M;Granada;Spain
3;28;F;Valencia;Spain
...
```

Figure 2: Example of user file

item.csv

```
itemID;director;movieCountry
1;Takeshi Kitano;Japan
2;Tom Six;Netherlands
3;Don Siegel;United States
...
```

Figure 3: Example of item file

context.csv

```
contextID;time;season
1;night;spring
2;morning;autumn
3;afternoon;winter
...
```

Figure 4: Example of context file

ratings.csv

```
userID;itemID;contextID;rating
1;2;1;4
1;3;2;5
2;1;3;4
…
```

Figure 5: Example of rating file

The structure of the output files is defined by the corresponding scheme files. Output files can also be used as input files in some specific generations.

These files can be saved in the form of datasets in the current workspace, to be able to see their data, download the files or evaluate them as you please.

2. Functionalities of DataGenCARS and AUTO-DataGenCARS

DataGenCARS allows the automatic mapping of item schemes into Java classes (and vice versa), the definition of item and user profiles, the definition and creation of users, items and contexts through scheme files composed of defined attributes, the generation of completely synthetic datasets, the increasing of ratings in the existing dataset, the generation of a synthetic dataset similar to an existing one, the generation of a dataset from an initial sample of an existing dataset and the removing of unknown contextual information [2].

AUTO-DataGenCars not only provides this, but also adds all the facilities that the use of a graphical interface implies, along with other options to improve quality of life such as easy creation of attributes for schemes, chaining of generations, visualization of statistics, etc. The evaluation tool for the generated data is also an important addition, being able to evaluate these as such or modifying them through experimental options.

As an auxiliary functionality, the possibility of saving already generated datasets in a workspace has also been incorporated into the interface, either to be able to visualize the data when needed, or to use it easily in a dataset to be generated. User-created attributes can also be stored in a workspace, edited and used in schemes with ease.

	DataGenCARS	AUTO-DataGenCARS
Mapping of item schemes into Java classes (and vice versa)	✅	✅
Creation of users, items and contexts through scheme files	✅	✅
Generation of completely synthetic datasets	✅	✅
Increasing of ratings in the existing dataset	✅	✅
Generation of a synthetic dataset similar to an existing one	✅	✅
Generation of a dataset from an initial sample of an existing dataset	✅	✅
Ability to remove unknown contextual information	✅	✅
Graphic interface		✅
Automatic data generation for input files		✅
Ability to chain different types of generation		✅
Graph showing how the actions included in the workflow (e.g., enlarge an existing dataset + remove unknown values) interact through input and output files		✅
Viewing statistics through built-in graphs		✅
Export dataset to a Weka file		✅
Export and Import workflows		✅
Evaluation tools		✅
Evaluation tools with experimental settings		✅
Saving datasets in a workspace for easy viewing and later use		✅
Saving attributes in a workspace for easy viewing and later use		✅

3. Workspace, Projects and Workflows

The DataGenCARS application uses different components to make it as easy as possible to generate multiple datasets, whether they are identical, similar, or completely different.

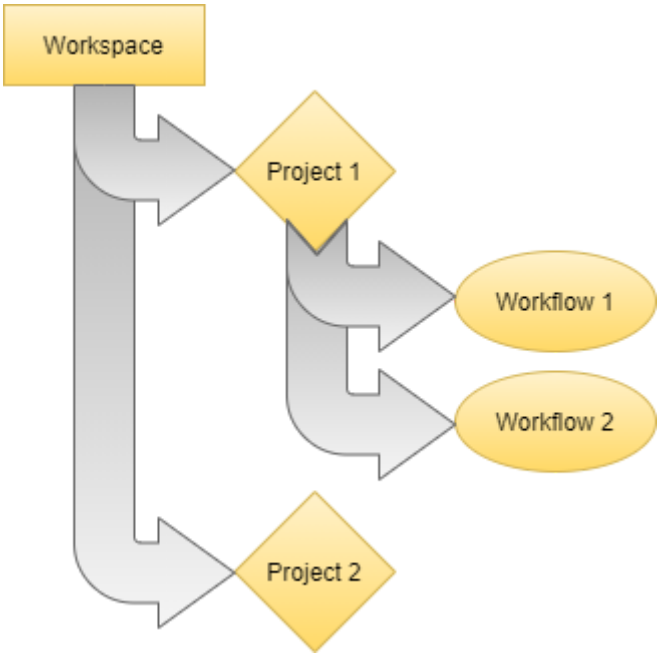

Figure 6: Workspace, Projects and Workflows

At the highest level are the workspaces, a folder selected by the user where everything necessary to run the application will be created, in addition to storing information on custom attributes and saved datasets.

Going down one level we find the projects, which allow us to group workflows as we see fit. These have their own folder within their corresponding workspace.

The workflows grouped in the projects are the elements where all the necessary data for the generation of a dataset, as well as its subsequent data visualization, are created and filled in. Just as the projects with the workspace, each workflow has its own folder inside the project's folder.

While projects can only be opened or deleted, workflows can also be moved between projects and be duplicated, as well as be opened and deleted. If needed, workflows can be exported as a *zip* file, and imported in another workspace in the same or another computer.

The directory tree can be used to open or delete a workflow/project, and also to move a workflow to another project. The menus at the top of the interface have the options to duplicate or move the current workflow (in the *Project* menu) or export and import a workflow to this workspace (in the *File* menu).

4. AUTO-DataGenCARS Components

AUTO-DataGenCARS interface is made up of three important elements, a tree of projects, workflows and datasets, a bottom bar with information on the status of the program and the main menu of the workflow. In addition, there are also other auxiliary menus to perform other functionalities. All these components will be detailed in the following sections:

4.1. Directory tree

Tree that represents the projects, workflows and datasets found in the current workspace. Through this tree we can change workflow / project, delete them or move a workflow from one project to another. We can also select a dataset to see its data in more detail.

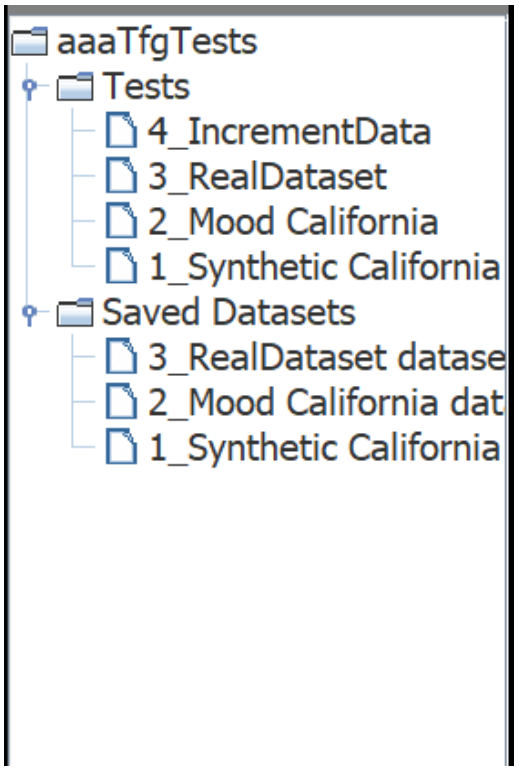

Figure 7: Directory tree

4.2. Bottom bar

Bottom bar of the application in which the current status of the interface is shown, both the current project/workflow and the internal status. A log can be opened from here as well to check the internal processes in more detail.

4.3. Workflow menu

Group of tabs where all the necessary information about the datasets and their generation are introduced, modified and displayed. The tabs are as follows:

4.3.1. Workflow main menu

In this tab the name, path and types of generation to use of the current workflow can be seen and changed, as well as being able to check a graph that details what elements are needed for each generation and what each file will be used for.

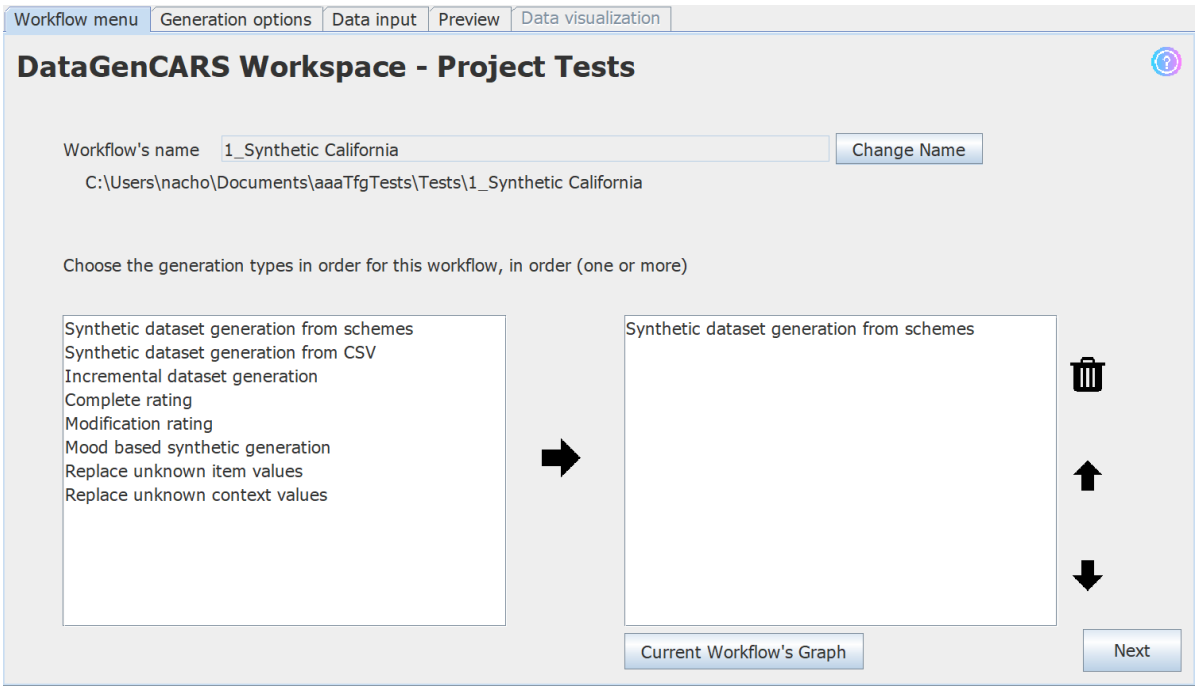

Figure 8: Workflow menu tab

4.3.2. Generation options

Tab to modify the basic configuration for the generation of the current workflow's dataset. Some examples of the data to modify are the number of elements to be generated, the minimum and maximum ratings or the percentage of users to generate with each defined user profile.

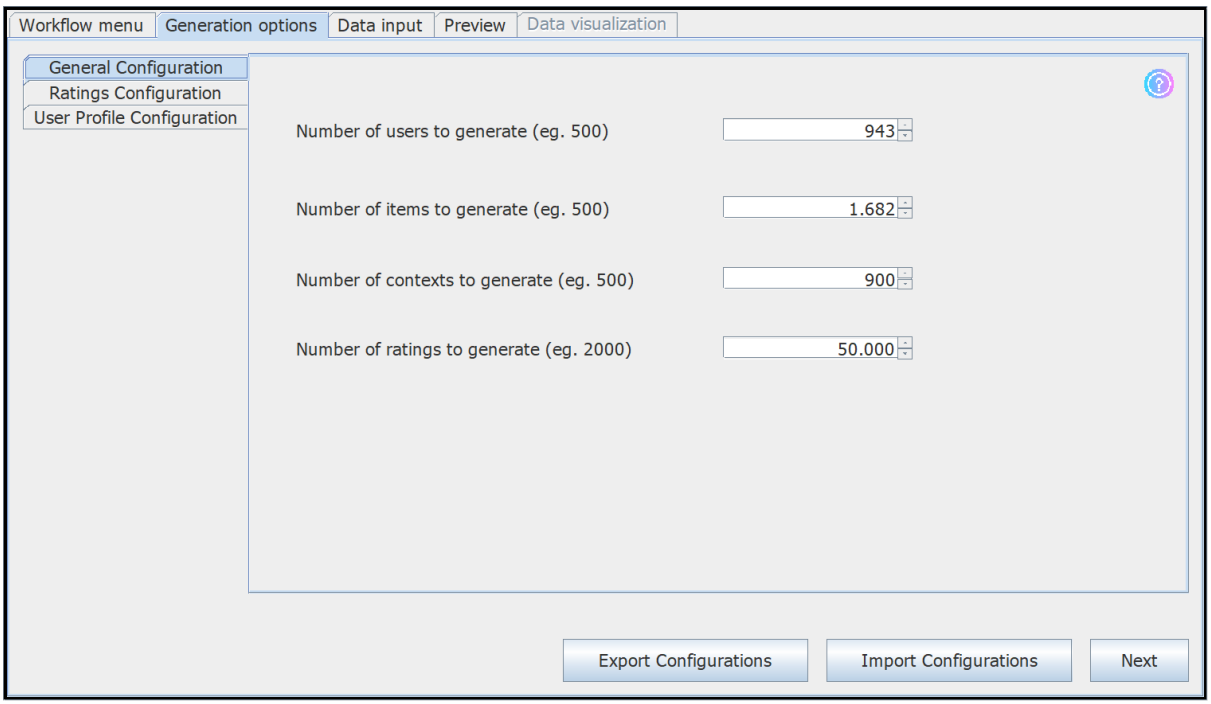

Figure 9: Generation options tab

4.3.3. Data input

Tab in which the necessary data is introduced to be able to generate a dataset, regardless of the type of the generation. You can fill in the user, item and context schemes, create user profiles based on the item and context attributes marked as important and add files in CSV format of already existing datasets or user profiles.

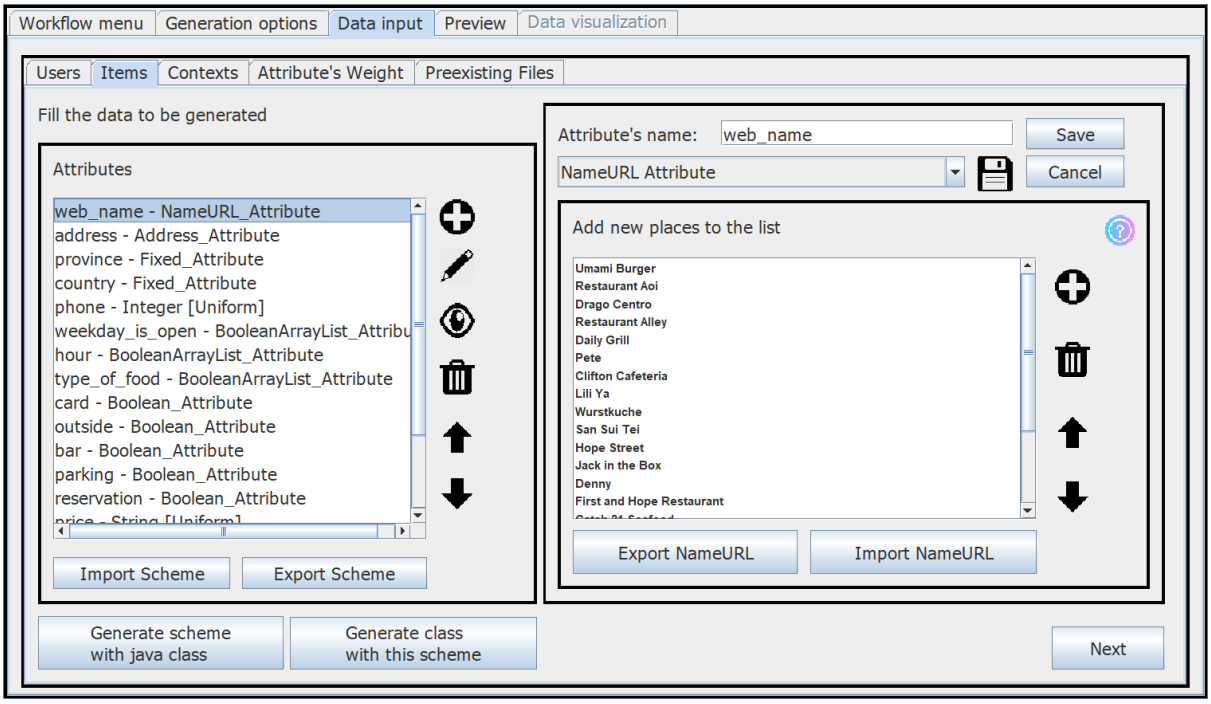

Figure 10: Data input tab

4.3.4. Preview

Last tab before generating the dataset. You can see information such as the generations to be carried out, the missing data to fill in (if any), the attributes of the schemes and statistics of the preexisting csv files introduced.

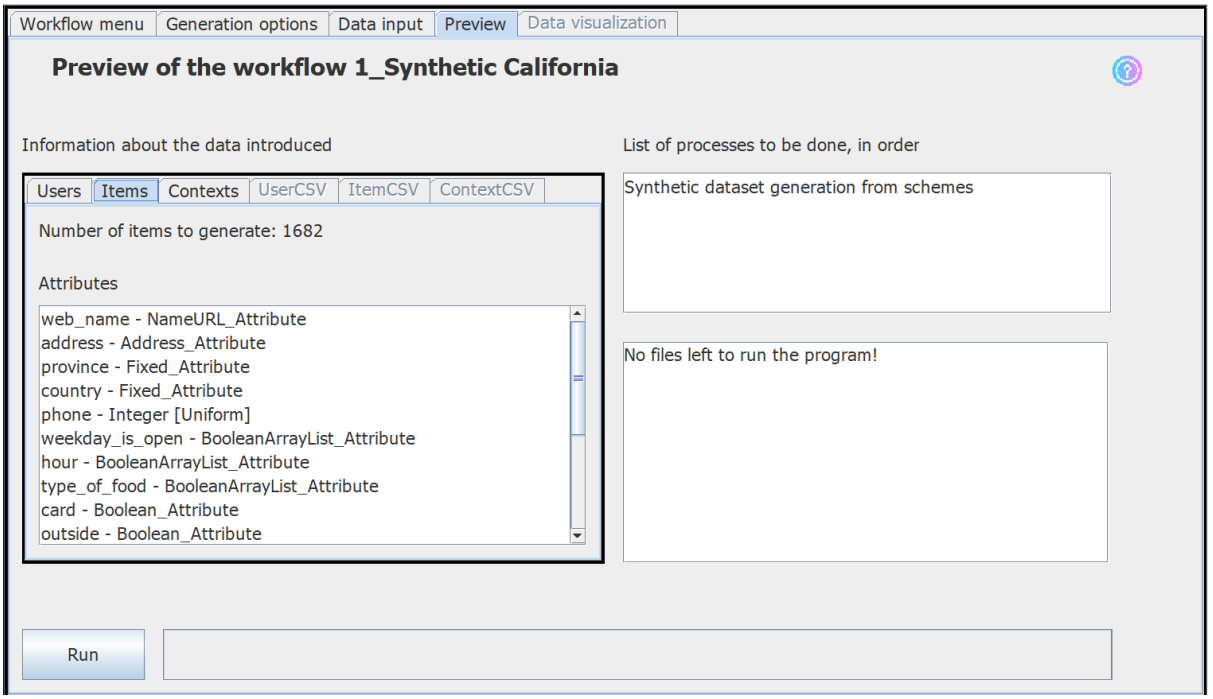

Figure 11: Preview tab

4.3.5. Data visualization

Accessible at the end of the dataset generation, different statistics about the newly generated dataset will be displayed here, in addition to allowing the download, evaluation and saving of the previously mentioned dataset in the workspace.

Figure 12: Data visualization tab

4.4. Additional menus

Different windows for additional functionalities. The evaluation, saved datasets and saved attributes menus fall into this category. Both the saved datasets and attributes menus show the ones stored in the current workspace.

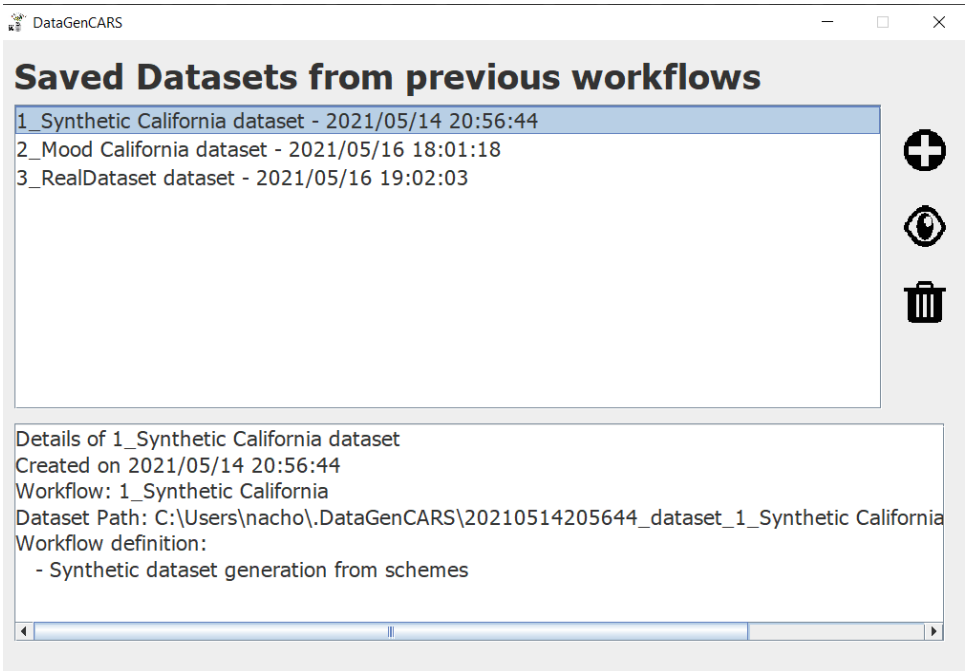

Figure 13: Saved datasets menu

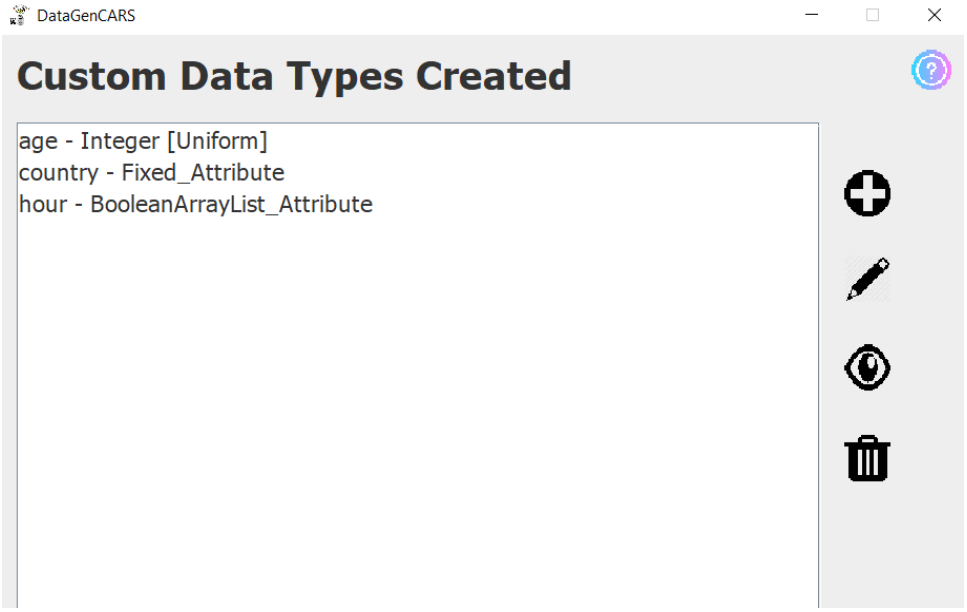

Figure 14: Saved attributes menu

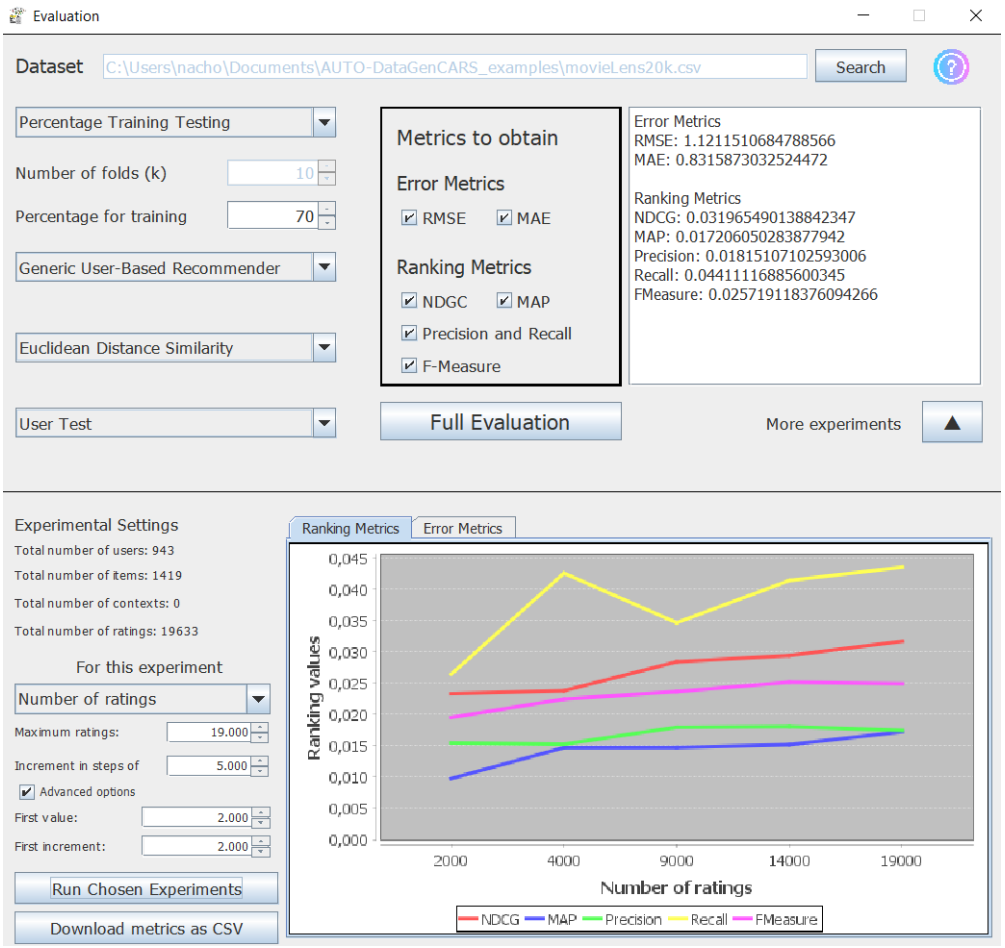

Figure 15: Evaluation menu

5. Examples of Dataset Generations with AUTO-DataGenCARS

Here are some practical examples of AUTO-DataGenCARS, using the key elements explained in the previous points.

5.1. Generating and exploiting a synthetic dataset

In the first example we are going to create a completely synthetic dataset with no previous data. This dataset will focus on a restaurant recommendation scenario for mobile users located in the state of California. The schemas of users, types of items, and contexts considered, are defined as follows:

- **Users**: *age, gender, occupation*.
- **Restaurants**: *web_name, address, province, country, phone, weekday_is_open, hour, type_of_food, card, outside, bar, parking, reservation, price, quality_food, quality_service, quality_price, global_rating*.
- **Contexts**: *transport_way* (walking, bicycle, car, public), *mobility* (stopped, moving), *weekday* (week, weekend), *season* (spring, summer, autumn, winter), *companion* (alone, friends, family, girlfriend, children), *temperature* (warm, hot, cold), *weather* (sunny, cloudy, rainy, snowing), *distance* (near, far), *time_of_day*(morning, night, afternoon).

We open a new workspace:

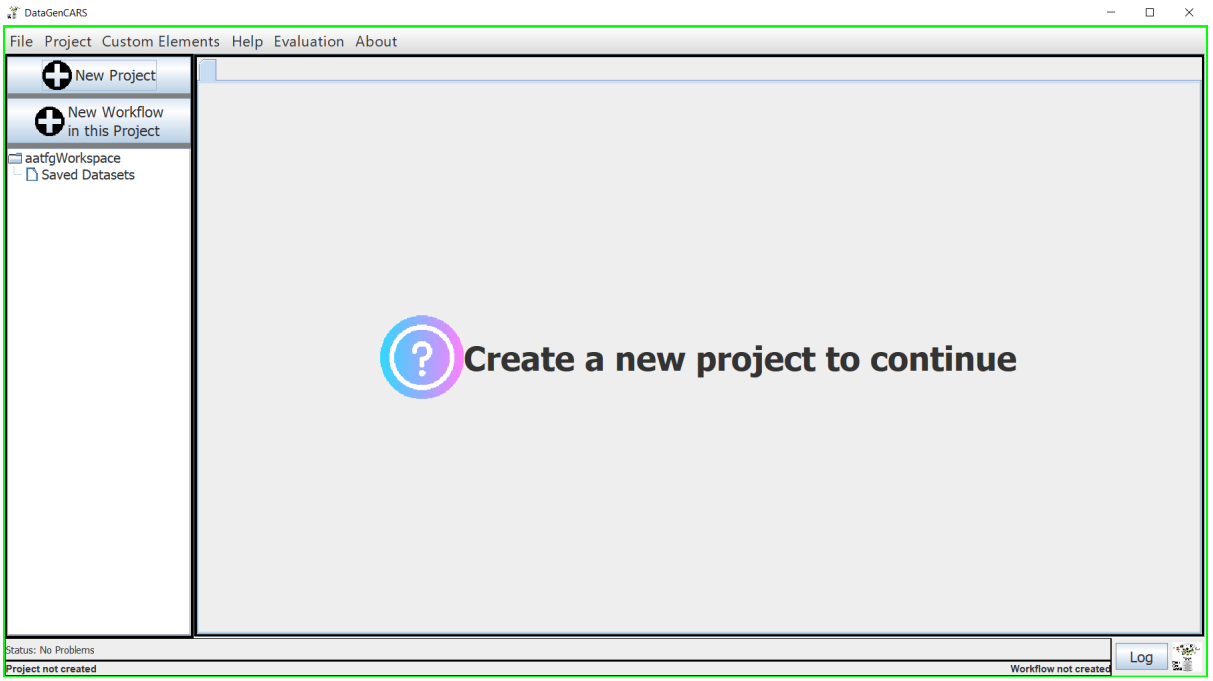

Figure 16: New workspace

To start we create a new project, and once created we create a new workflow in that project. With the workflow created, we name it and indicate that we want to use the *"Synthetic dataset generation from schemes"* generation. Generations can be added to the workflow by clicking the arrow icon pointing to the right.

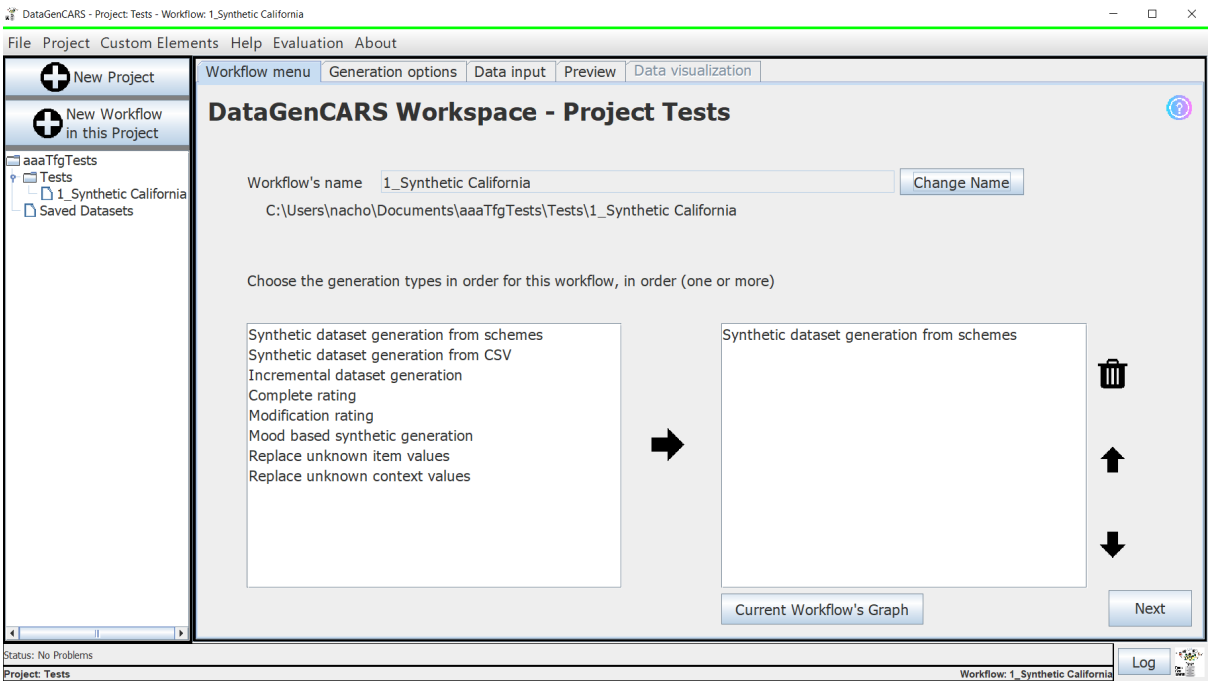

Figure 17: Workflow created

This scenario will consist of *943 users*, *1682 items* (which in this case will be restaurants) and *900 contexts*. We advance, either with the next button or by clicking on the tab names, to the *"Generation options"* tab and enter these figures in the general configuration tab.

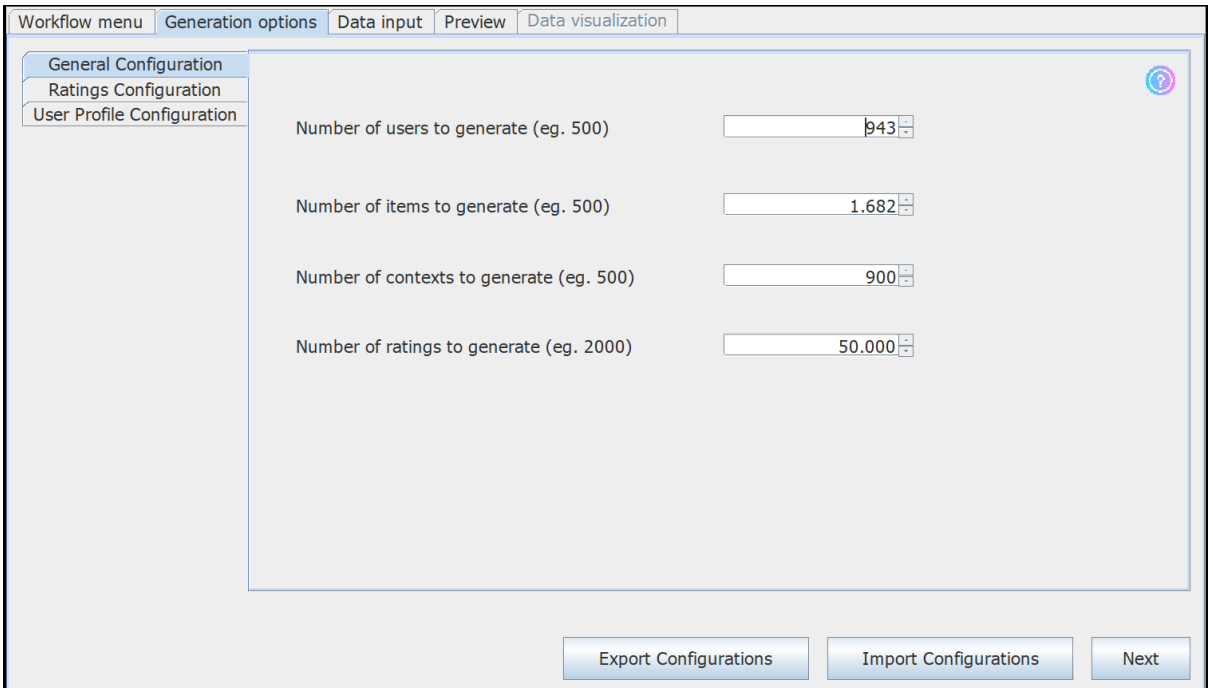

Figure 18: Introducing the general configuration

Specifically, we want to synthetically generate ratings whose values are between 1 and 5, and also labeled with a date, which will be in the range from 1980 to 2020. We enter these data in the ratings configuration tab.

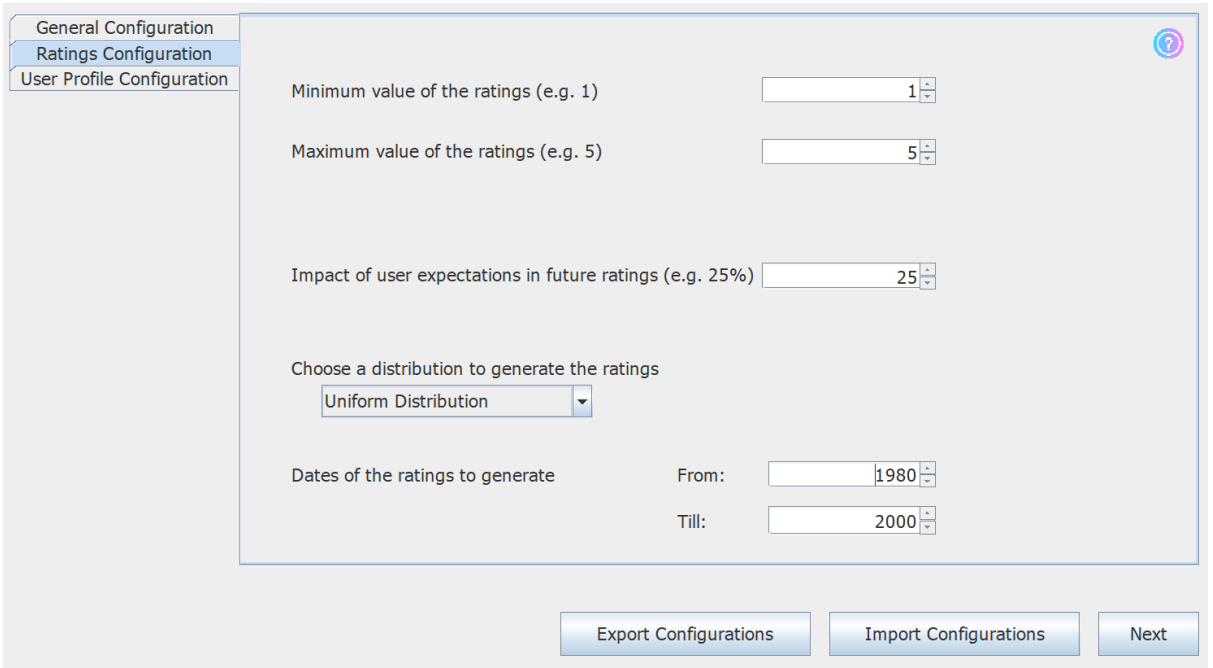

Figure 19: Introducing the ratings configuration

Next it is necessary to create the schemes of the users, the restaurants and the contexts. To do this, we enter the *"Data input"* tab and begin to create different attributes, such as the following one that represents the age of a user:

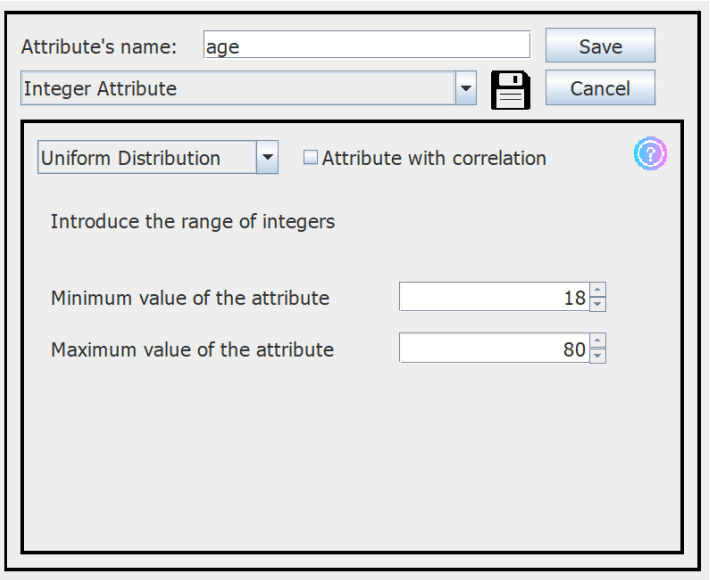

Figure 20: Age attribute

If we want to save this attribute for later, click on the save icon (the *floppy disk*). We can use this attribute as it has been created later, or we can view or modify it in the *"Custom Data Types"* window *(Custom Elements> Custom Data Types)*.

Once the necessary attributes have been created, the users' scheme would look like this:

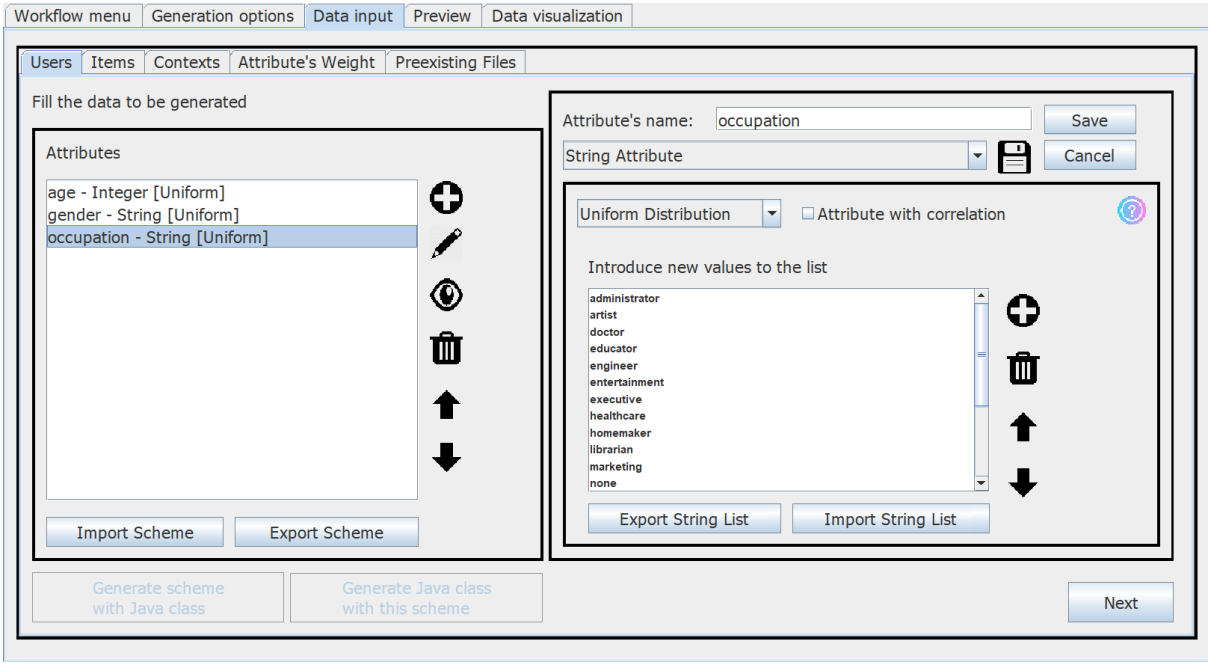

Figure 21: User's scheme

If we want to see an attribute that has already been created without modifying anything, we can click on the eye icon while the attribute is selected to see how said attribute is composed. If we want to modify it, use the pencil icon, if we want to erase it, the bin icon. We can also raise or lower it in the list with the arrow icons, and if we want to add another new attribute, we can press the plus icon (if another attribute is being edited or inspected).

The scheme of the restaurants would look like this:

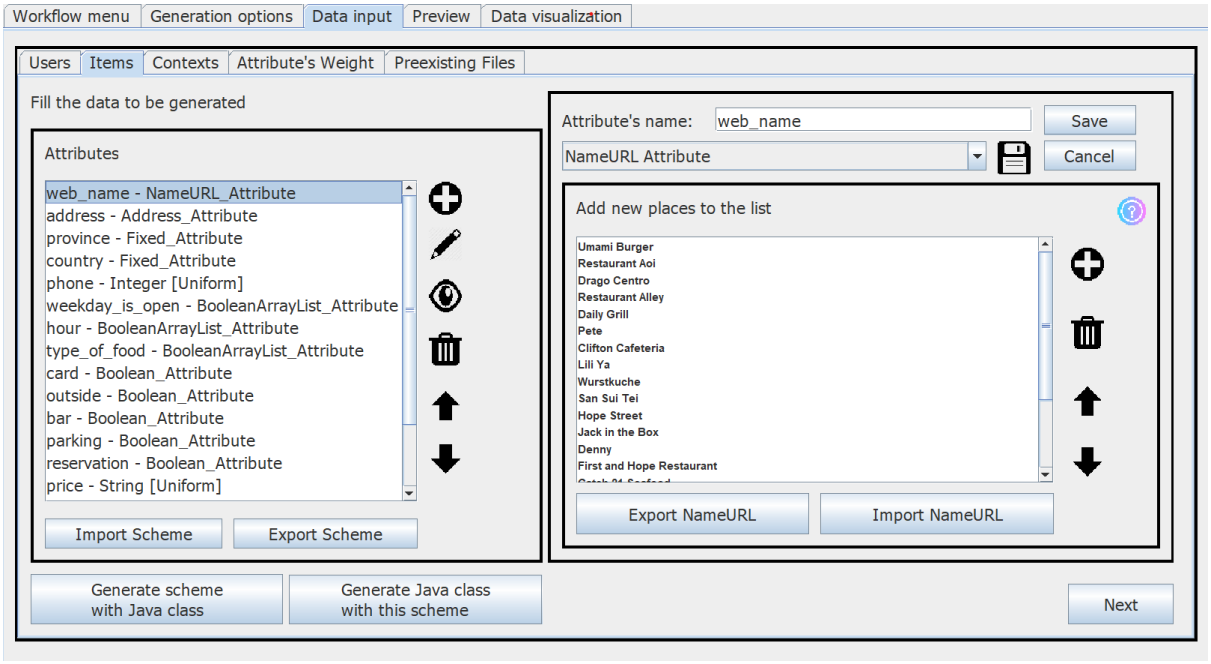

Figure 22: Restaurant's scheme

And the context's scheme:

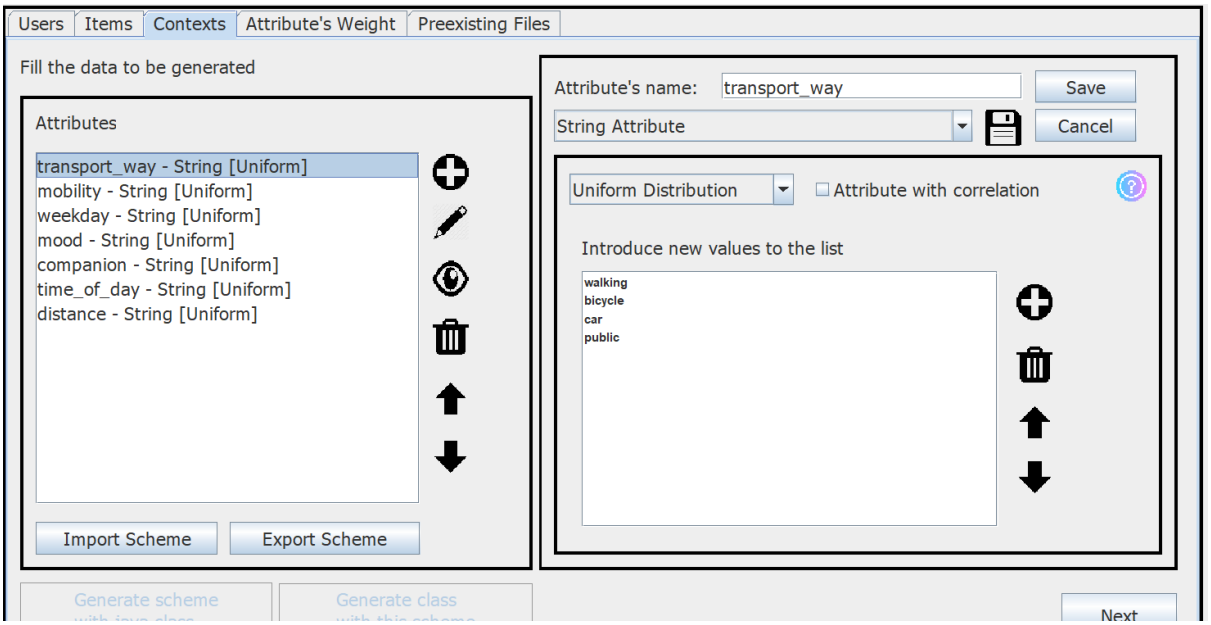

Figure 23: Context's scheme

Next, we move on to the *"Attribute's weight"* tab, in which we select which attributes of the restaurants and contexts are most relevant, and if they have a higher or lower ranking order.

We select the attributes transport_way and distance from the context of the users, and parking, price, quality_food, quality_service for restaurants.

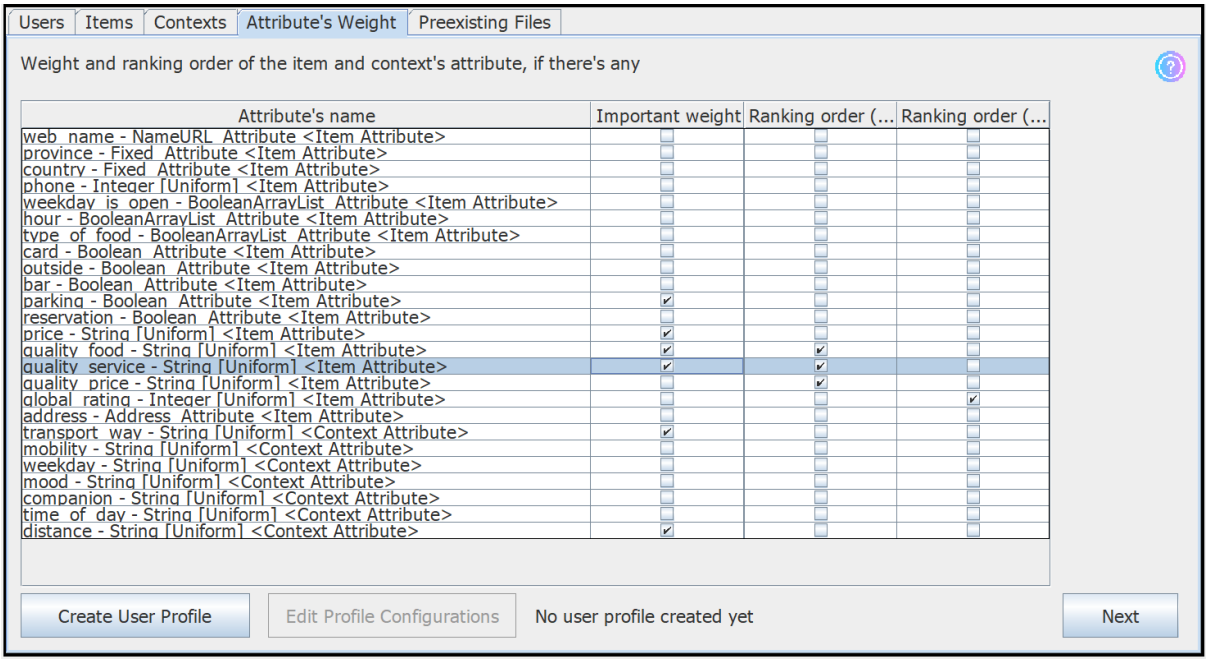

Figure 24: Attribute's weight

With these attributes selected we are going to create 5 different user profiles, and we are going to give different weights to each attribute according to the profile, remembering that the sum of all the weights must equal one (we can press *"Weight readjustment"* so that the weights are readjusted, or we can leave it as we want that they will be readjusted when executing). The created profiles would look like this:

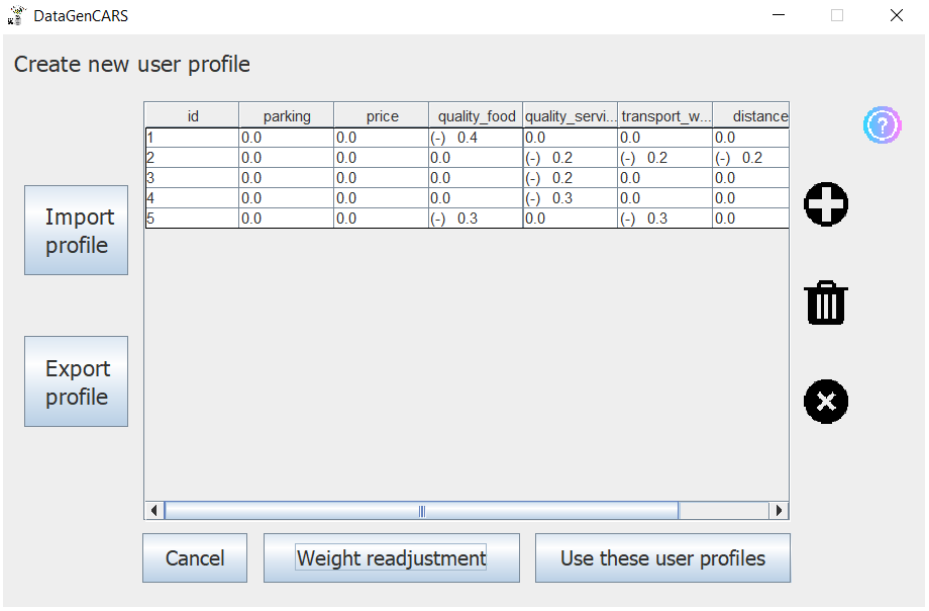

Figure 25: User profiles

Once saved, click on *"Edit Profile Configurations"* to edit if we want to change the percentage of users to create with each profile and the percentage of noise that there will be.

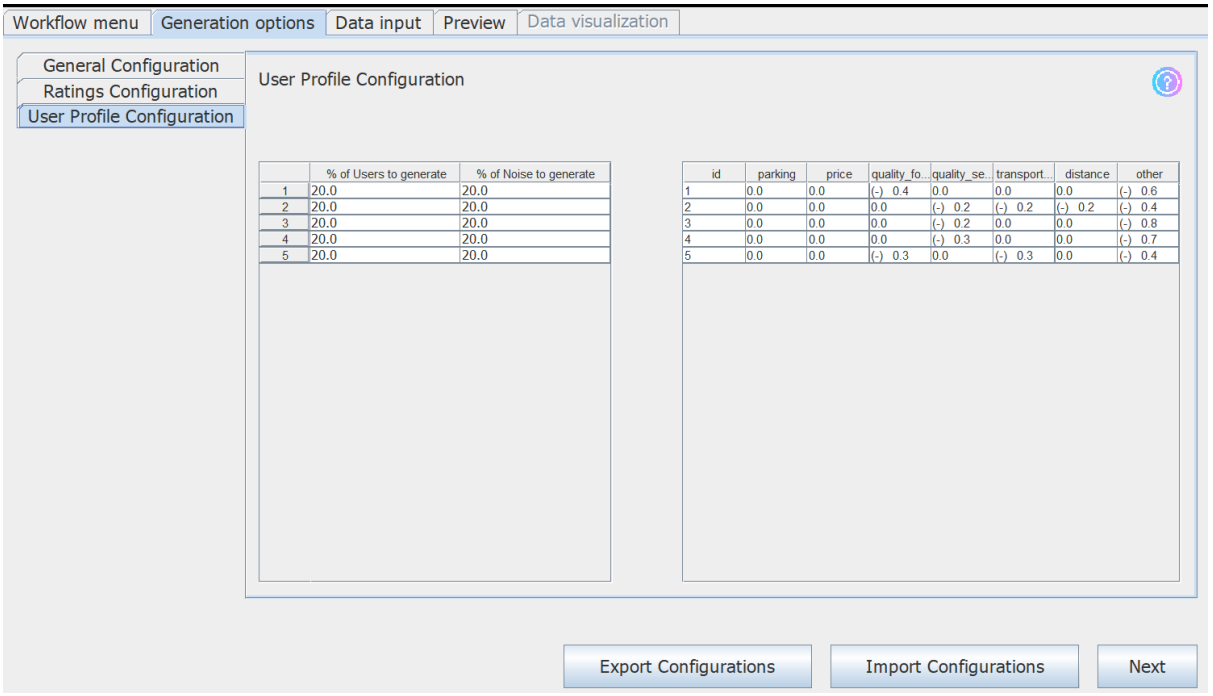

Figure 26: User profile configuration

In the next tab, *"Preexisting Files"*, nothing is needed to run this generation so we can move on to the next one.

In the last tab before execution, *"Preview"*, we observe that the attributes of the schemes are correct and that we are not missing any more data, so we can proceed to run the generation. When finished, we automatically advance to the *"Data visualization"* tab, where we can observe different statistics about the generated dataset, as well as being able to download it in different ways, save it in the dataset or evaluate it.

Figure 27: Data visualization tab

In this case we are going to save it in the workspace and evaluate it.

When opening the evaluation window, we can simply choose different forms of validation, with different recommenders and strategies, and select the metrics to obtain. Now we're going to choose the K-Fold Cross Validation with 10 folds, a generic user-based recommender and the Pearson correlation similarity.

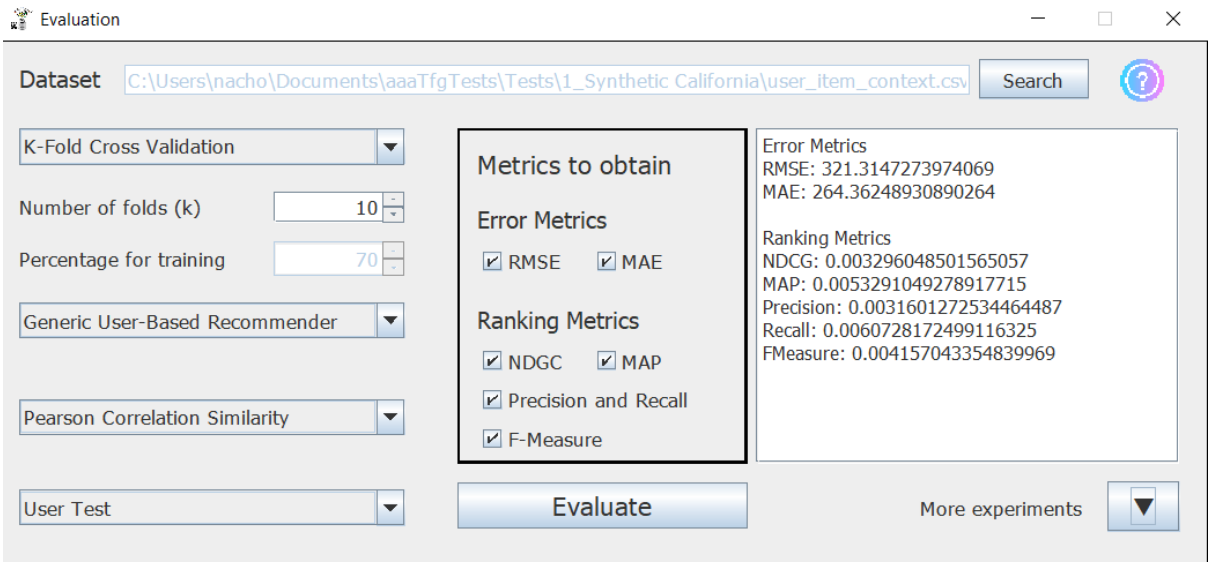

Figure 28: Dataset evaluation

If we click on the *"More experiments"* arrow, the window will expand to be able to carry out more exhaustive experiments with the same options selected. In this case we are going to perform an evaluation test of the dataset, but only with *2,500, 10,000, 20,000, 30,000, 40,000, and 50,000* ratings. After the experiment, we can see the ranking or error metrics in the graphs according to the number of ratings with which it has been evaluated

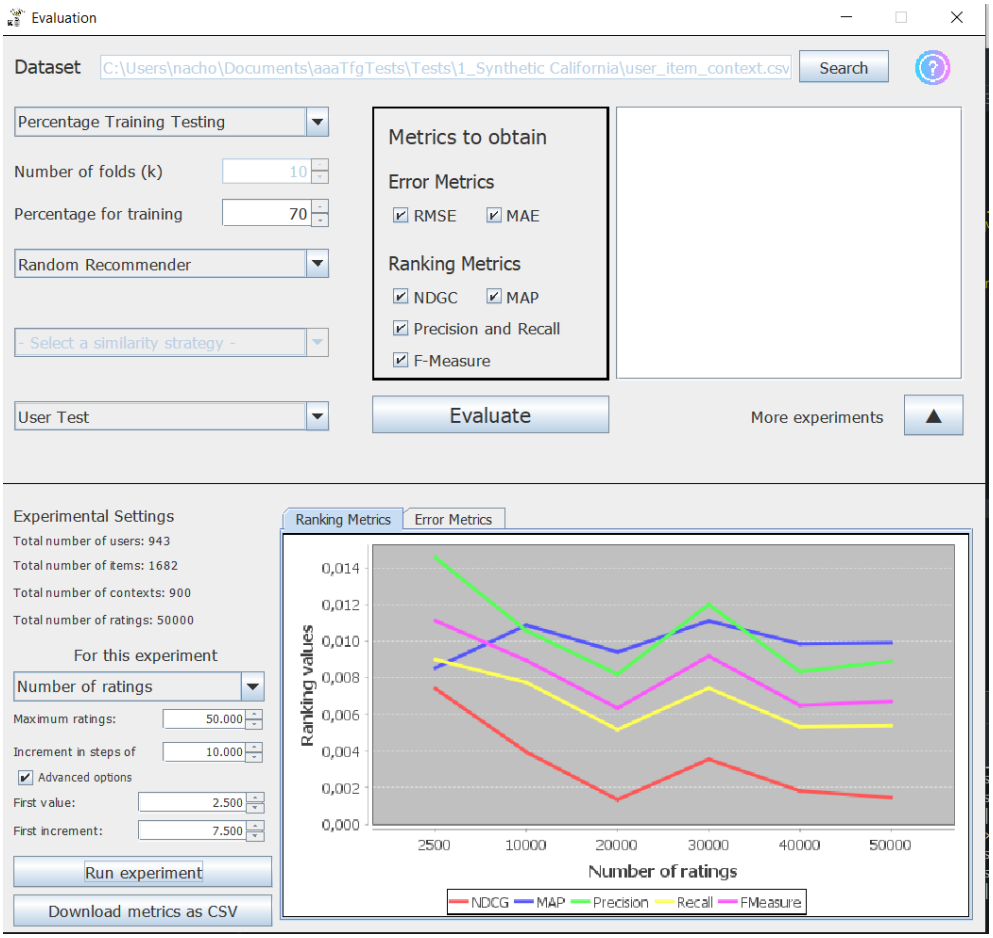

Figure 29: Dataset evaluation

5.2. Using a previously generated dataset

With the previously created dataset saved, we proceed to duplicate the workflow that we have run before by clicking on *Project* > *Duplicate current workflow*. Once duplicated, we look for the newly created workflow in the tree on the left and select it to change the workflow and modify its name and generations to run.

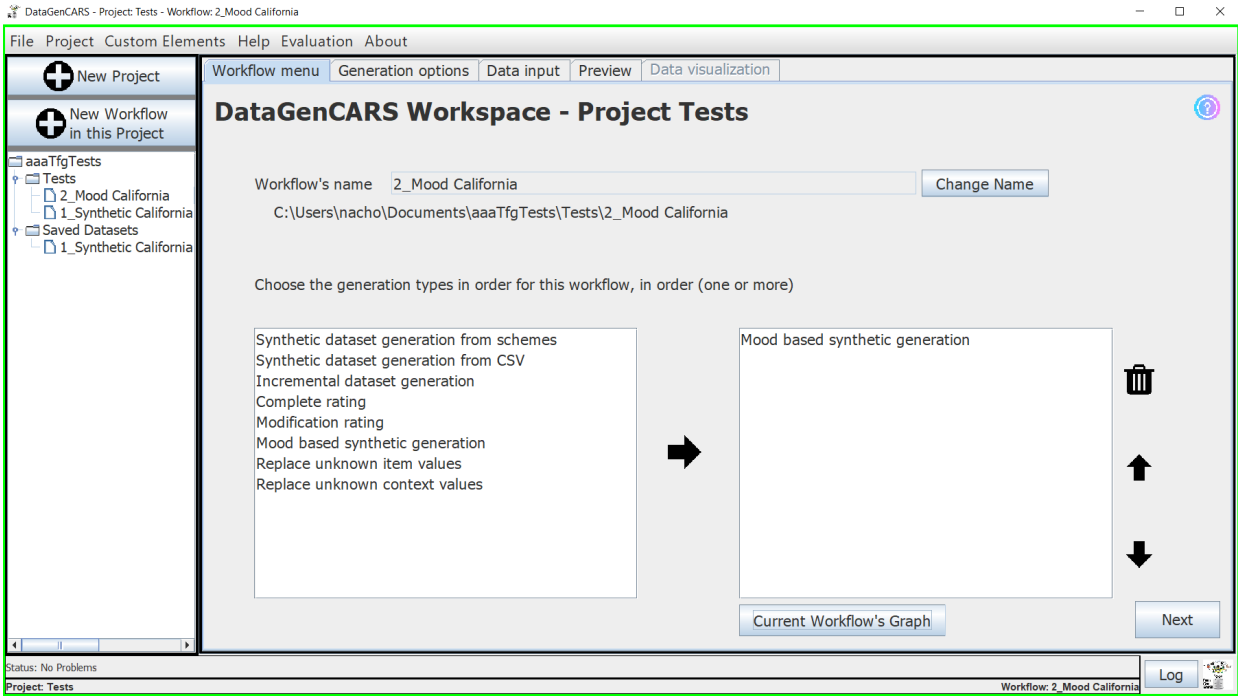

Figure 30: Duplicated workflow

Maybe we don't know what a *Mood based synthetic generation* is or what files it uses, so we press the help button (blue interrogation mark at the top right side) and look for that specific generation.

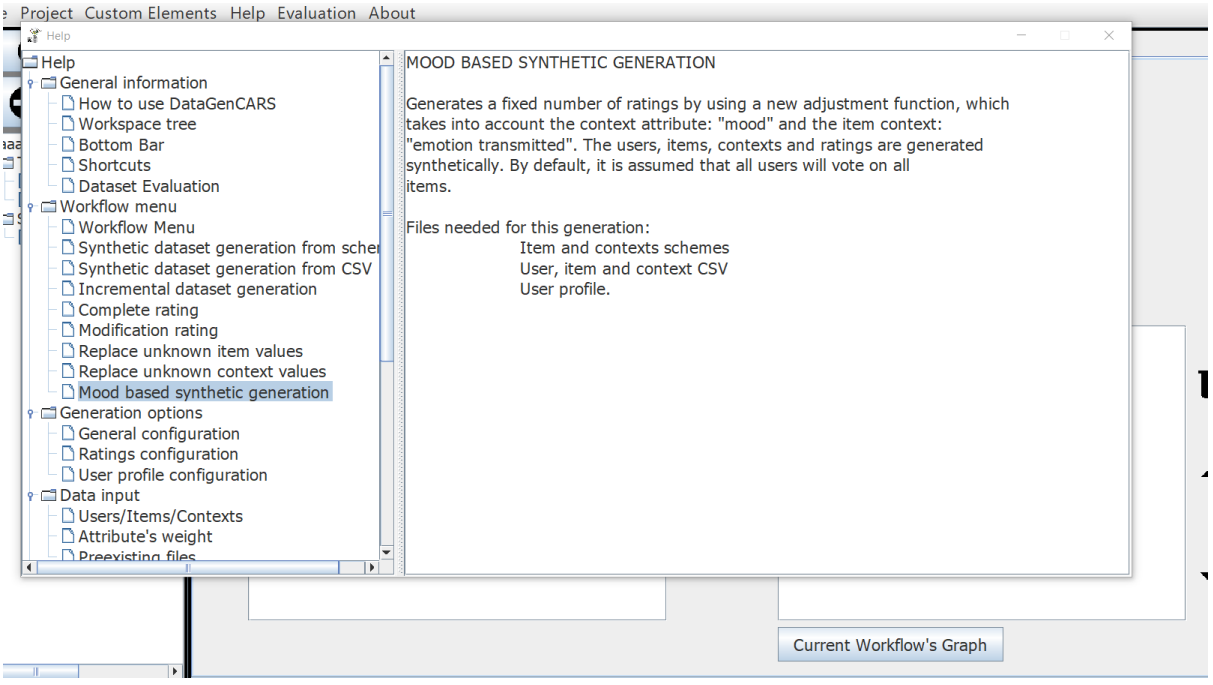

Figure 31: Help menu

Being a duplicate workflow, the restaurant schemes and user contexts are already filled in, as well as the basic configuration.

In the *"Preexisting files"* tab we can use the previously saved dataset to select the CSV file generated in that dataset, searching for said dataset in the saved datasets window (*Custom Elements> Saved Datasets* or by clicking on the button on this tab), selecting the dataset and clicking the necessary icon:

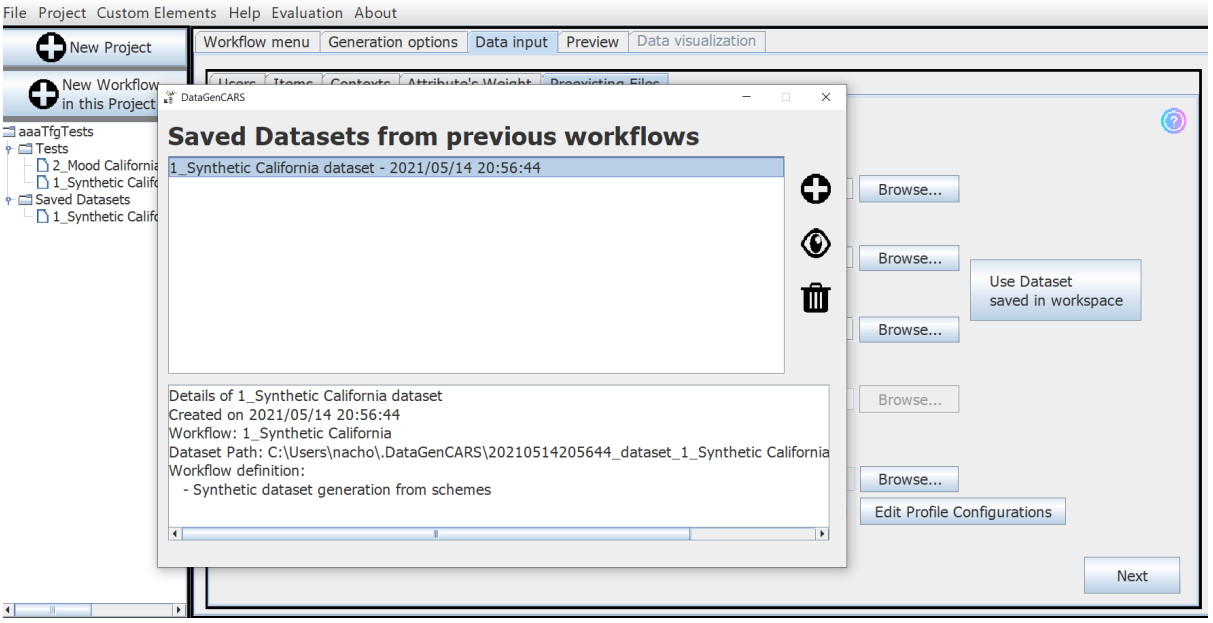

Figure 32: Using a previously saved dataset

Once this is done, the files we need will be automatically filled in:

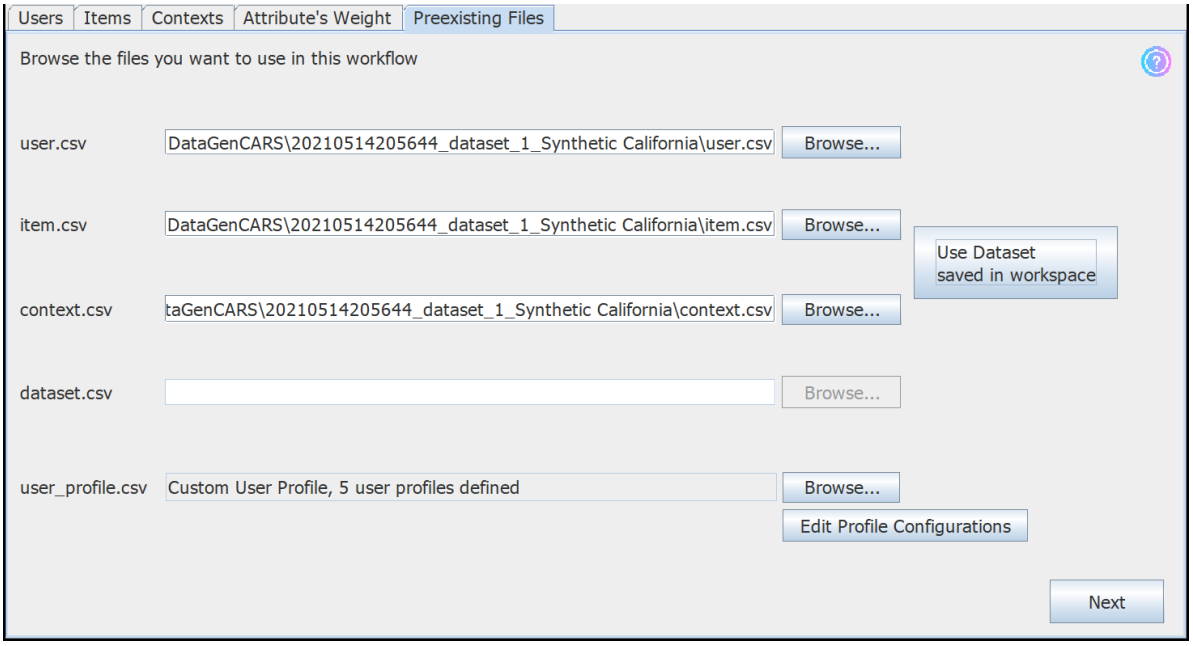

Figure 33: Preexisting files tab

And with this we can run the workflow, giving us a result like the figure 34 shows. We can see that by adding these mood attributes, users have valued these restaurants more positively.

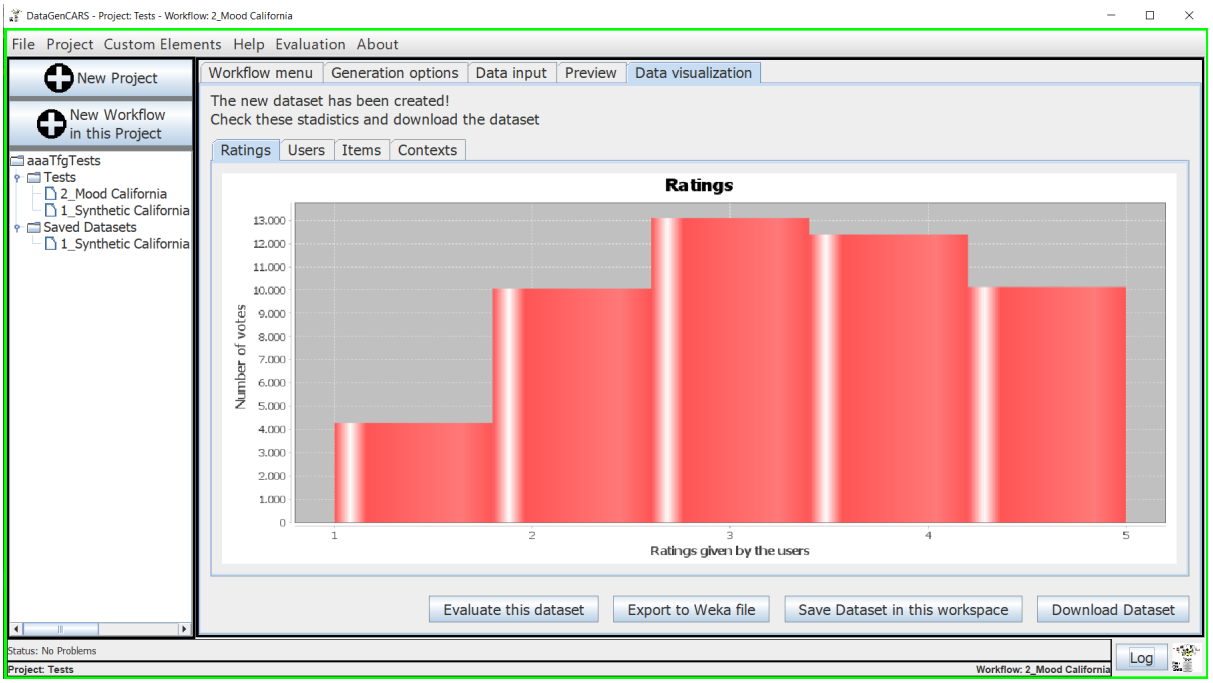

Figure 34: Results of this dataset generation

5.3. Working with a real dataset and run several generations at once

In this example we have a dataset already created, with the corresponding CSV file for users, items (which in this case are movies), contexts and ratings. There is also a problem, and it is that both in the movie file and in the context file there are some values that could not be filled in for whatever reasons, and they have the value NULL.

	A	B	C	D	E	F	G	H	I	J	K	L
1	itemID	director	movieCountry	movieLangua	movieYear	genre1	genre2	genre3	actor1	actor2	actor3	budget
2	1	549	36	9	2010	1	3	13	494	1348	1467	10000000
3	2	776	23	9	2010	10	14	21	491	135	137	1500000
4	3	246	37	9	2010	7	10	18	74	334	1910	30000000
5	4	438	37	9	2006	1	3	10	817	1763	1652	28000000
6	5	775	37	9	1998	7	6	10	1636	1539	1402	90000000
7	6	33	37	9	2008	7	10	18	1373	1510	1691	24000000
8	7	488	37	9	2003	3	10	18	98	385	1827	35000000
9	8	624	37	9	2007	1	8	21	1126	1992	1908	53000000
10	9	571	37	9	1993	1	6	10	1877	931	701	33000000
11	11	346	37	9	2010	7	18	18	862	61	1378	20000000
12	12	161	37	9	2003	10	NULL	NULL	482	1190	393	NULL
13	13	590	37	9	2010	1	14	19	1382	57	1109	60000000
14	14	234	36	9	2007	7	NULL	NULL	1303	1524	1656	9000000
15	15	195	37	9	1971	1	8	21	384	93	715	4000000

Figure 35: Movies.csv

We create a new empty workflow and add three generations in a row and in the order indicated. They can be rearranged using the arrow icons, or deleted with the trash can icon. To see what files we need for execution, and what happens to each file during it, we can press the *"Current Workflow's Graph"* button.

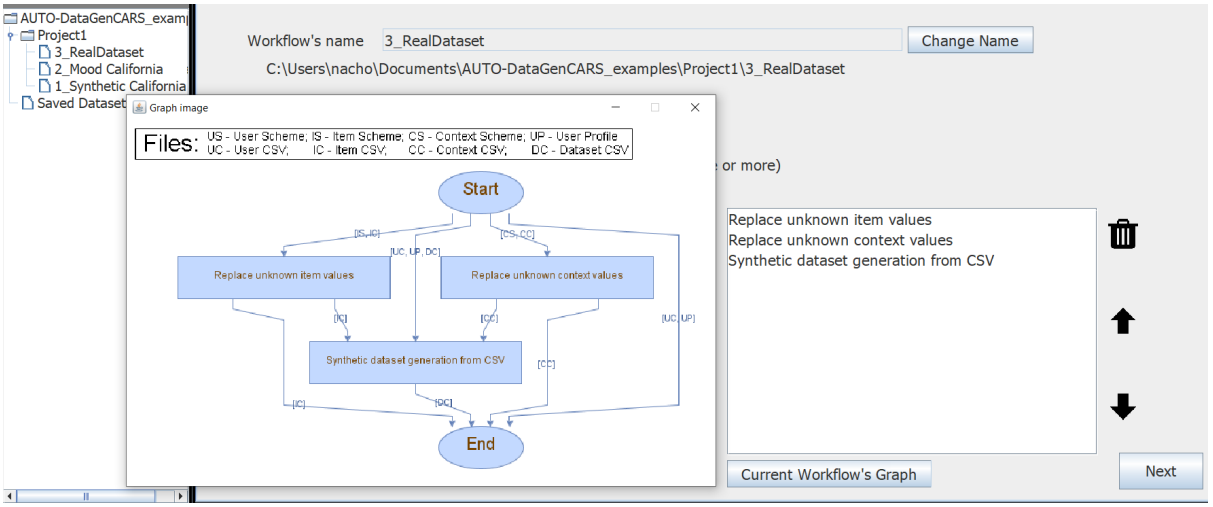

Figure 36: Workflow's graph

We put the same data that *users / items / contexts / ratings* have in the files, although we can change the ratings if we want:

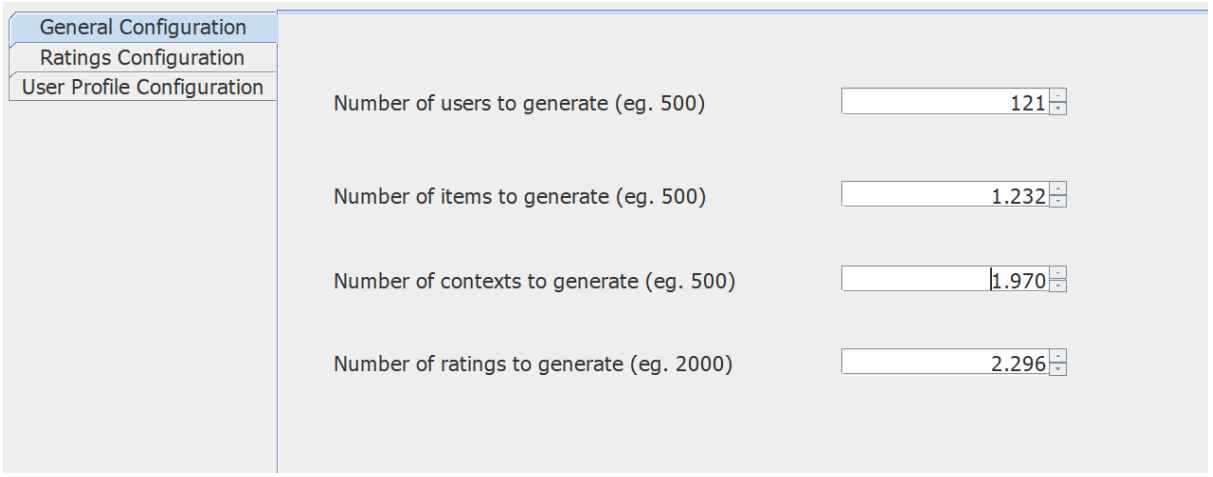

Figure 37: General configuration data

We also have the context scheme in a file already created, so we import it; and a java class that also matches the values of the *item.csv* file of the movies, so we can click the *"Generate scheme with java class"* button to find the class and transform it to scheme.

Next, we find and add each of the necessary CSV files, including the user profiles. In the case that it is not available and since we have the schemes of items and contexts, we can fill it in ourselves.

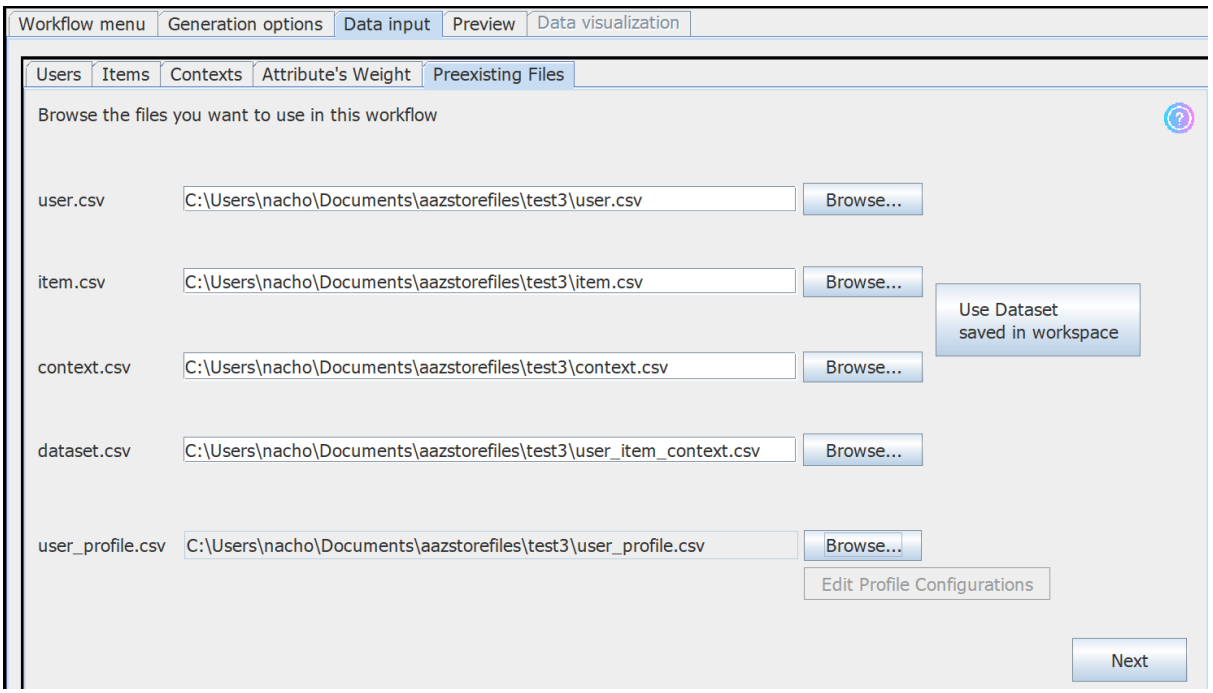

Figure 38: Preexisting files of the dataset

Now we can proceed to run the dataset generation in the *"Preview"* tab, review the results and do what we want with them. Let's proceed to save it in the workspace:

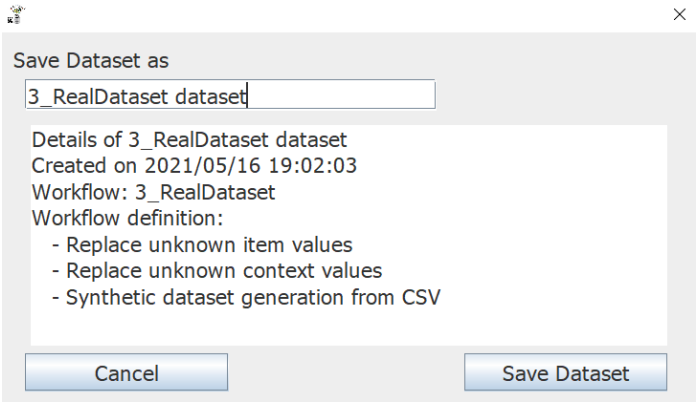

Figure 39: Saving a dataset in the workspace

Once saved, we can go to the saved datasets' menu (now also accessible by clicking on the corresponding saved dataset in the main window's tree) to add its files to the workflow that is currently being worked on, delete them or inspect the dataset itself:

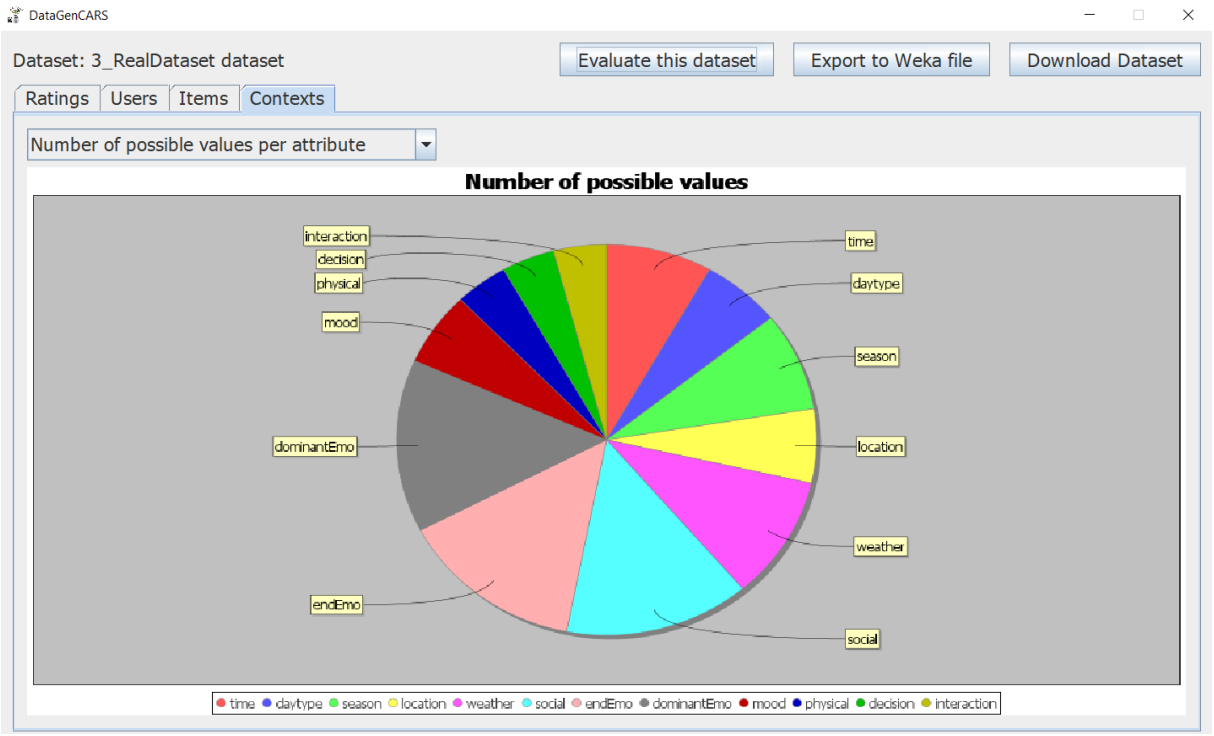

Figure 40: Statistics of a saved dataset

5.4. Enlarge an existing dataset

Now we want this same dataset created in the previous example, but increasing the number of ratings created in the generation.

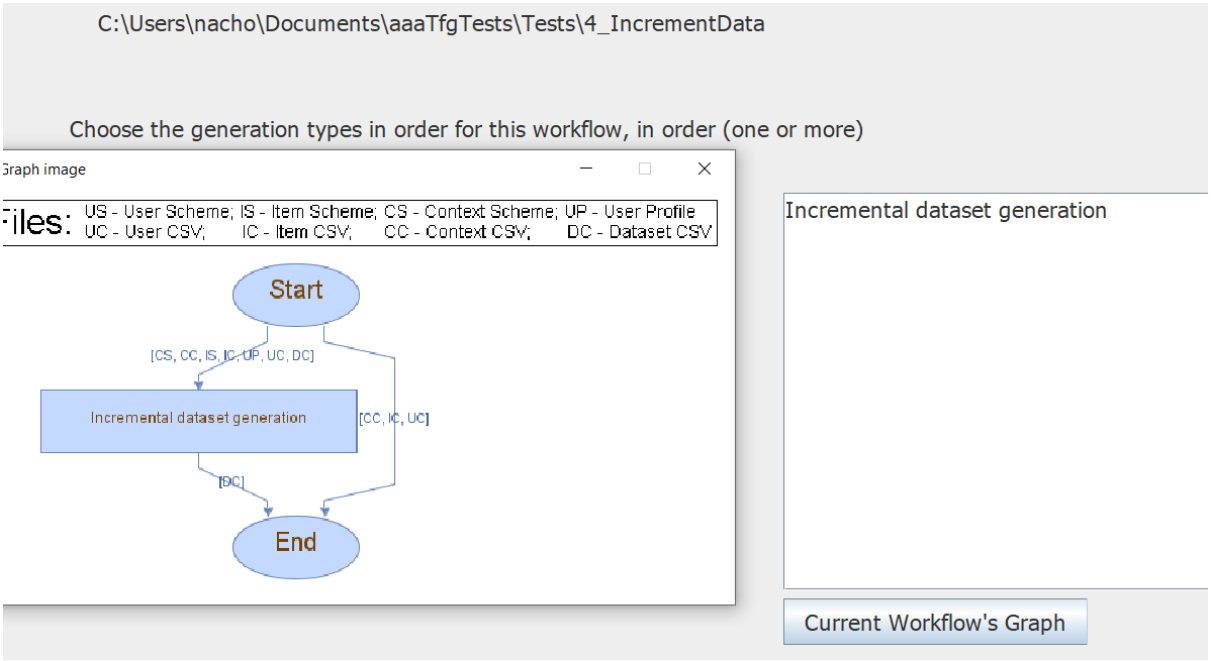

Figure 41: New workflow to increment the previous dataset

We increased the number of ratings to *5000*, instead of the previous *2296*.

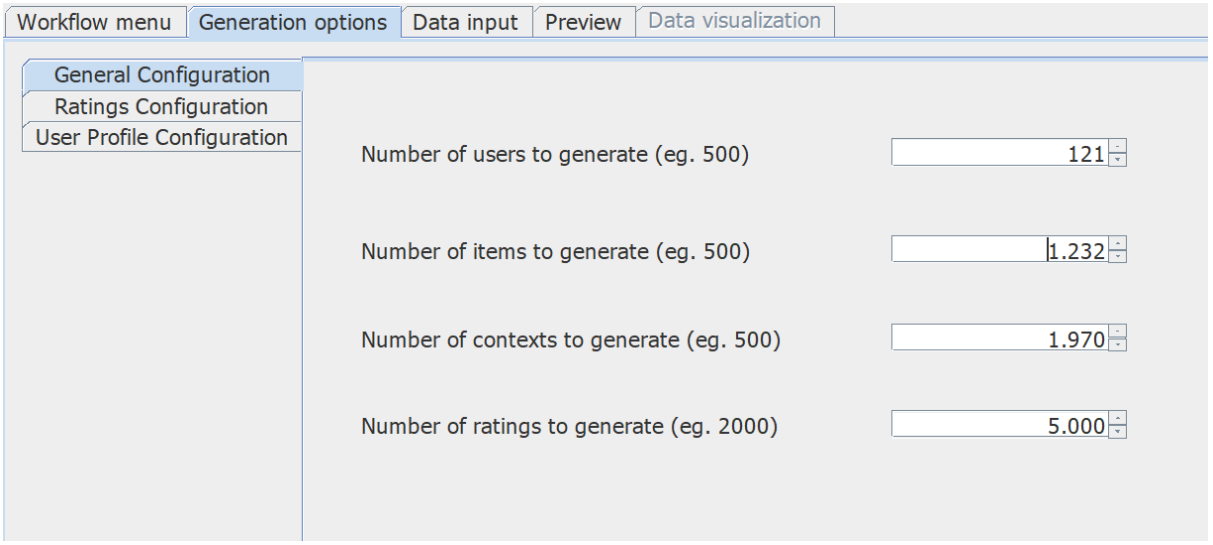

Figure 42: New general configuration

We import the already created schemes from items and contexts, and use the saved data from the dataset created in the previous step. After that, we can run the generation and check the new data.

5.5. Create a synthetic dataset with the same behaviour as an original dataset, but without context information

In this example a synthetic dataset will be generated using another existing dataset, both without context. We start by creating a workflow, giving it a name and selecting both the generation type of "*Generate a dataset's user profile*", which will create user profiles of the original dataset that will store their behavior, and "*Synthetic dataset generation from CSV*", to create the new ratings. Before going to the next tab, it is important to uncheck the "*Create dataset with context*" checkbox so that the context is not taken into account when generating the data, even if the original dataset had context. In the event that you want to work with datasets with context, or add context to a dataset that does not have it, you would need to leave this checkbox marked.

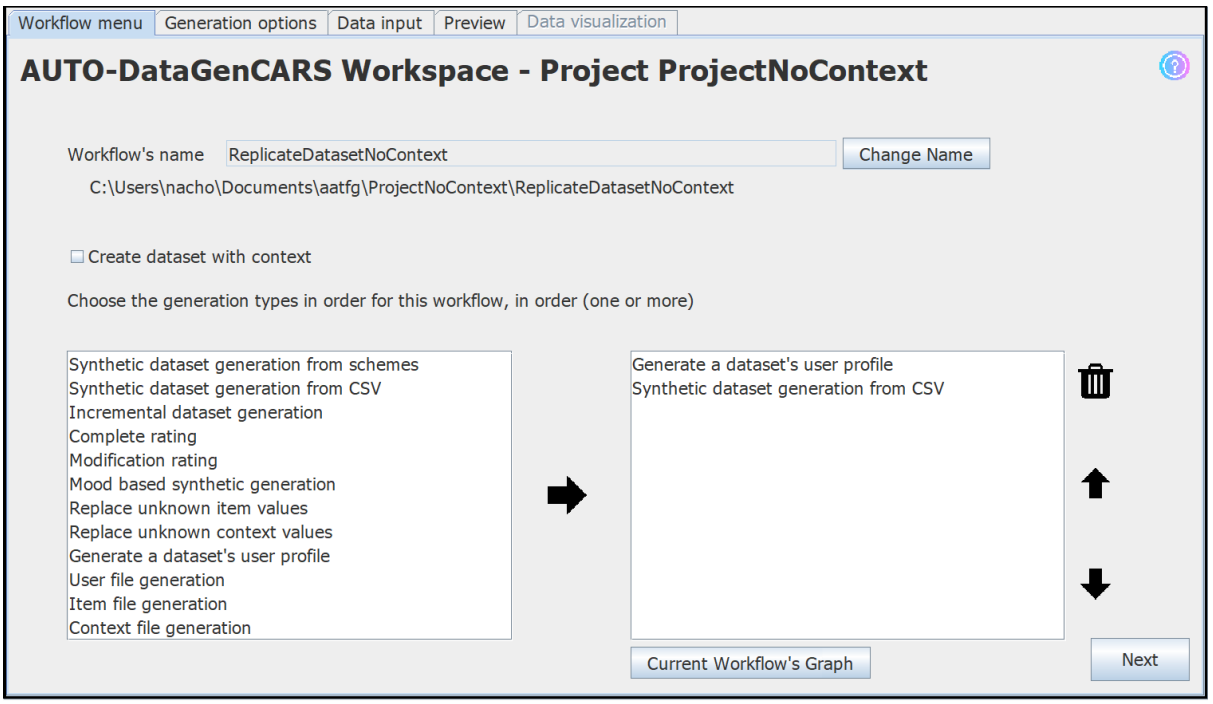

Figure 43: New workflow without context

You can modify the ratings' configurations in the "*Generation options*" tab, such as the minimum and maximum of the ratings, but we will leave it as it is by default for this example.

In the next tab, "*Data input*", we will not need to add anything except in the "*Preexisting files*" tab, where we browse for the user, items and the ratings CSV. Each column of each file must be separated by a semicolon.

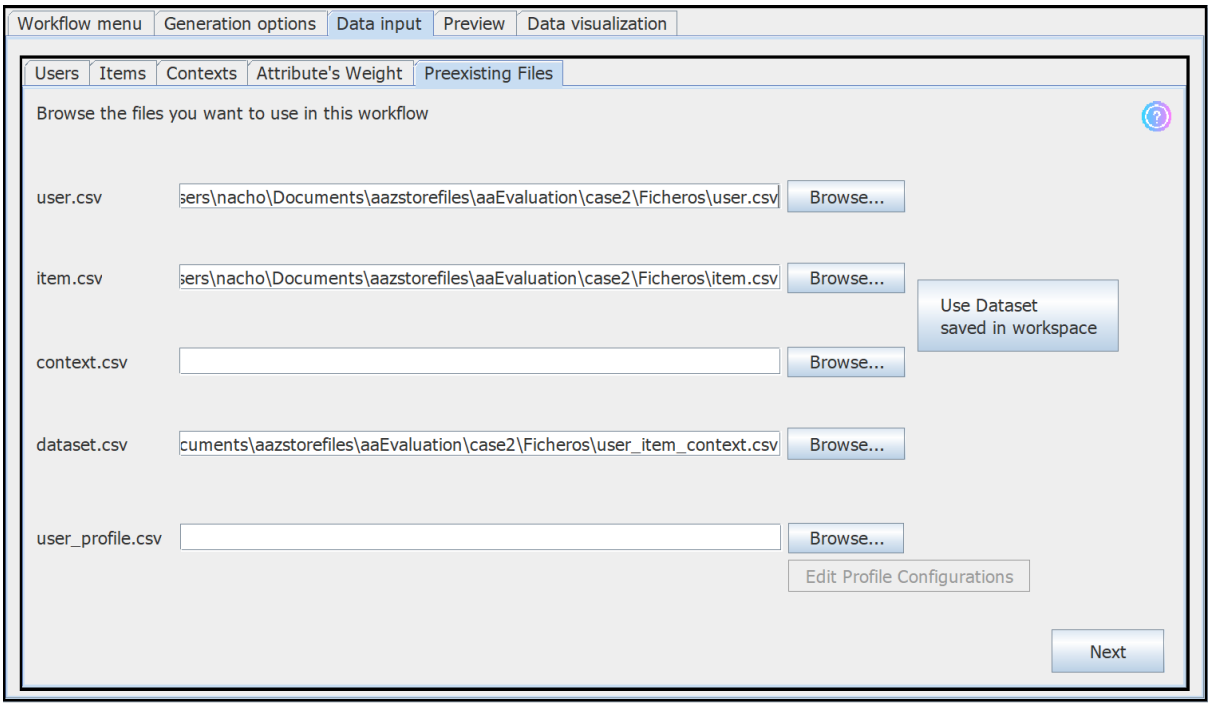

Figure 44: Browsing for the original dataset files

It is not necessary to create or import user profiles that indicate the behavior of the dataset to generate the ratings, since in the indicated generation, "*Generate a dataset's user profile*", *AUTO-DataGenCARS* automatically creates these profiles using the original files, and will use them for the "*Synthetic dataset generation from CSV*" generation. In the event that we want the ratings to be generated with another behavior, we can go to the "*Data input > Attribute's Weight*" tab and observe that the attributes of the browsed file have been added to the list, being able to mark them as important and create different profiles user by pressing the "*Create User Profile*" button; thus being able to give more or less relevance to the selected attributes.

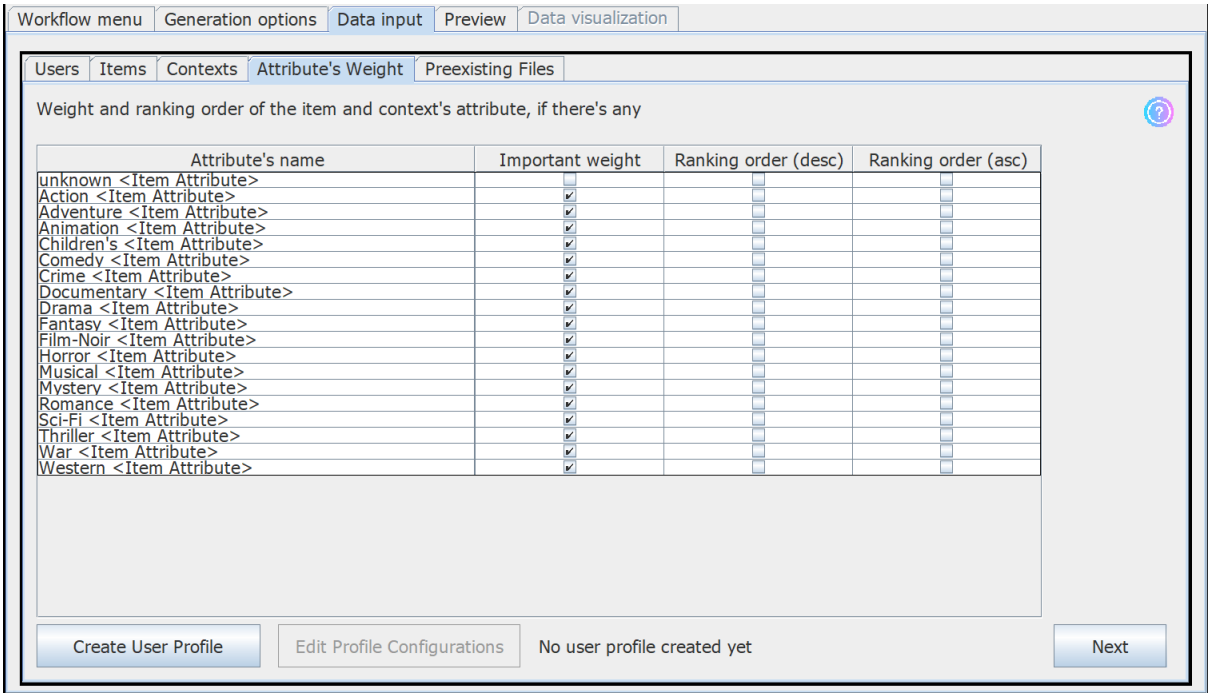

Figure 45: Items CSV attributes added to the list

Once with the original dataset files selected, we can run the generation in the "*Preview*" tab, being able to also see different statistics of these files as well. With the data generated, we can observe the distribution of the ratings (Fig. 46.), and more statistics, through the graphs automatically generated by *AUTO-DataGenCARS* in the "*Data visualization*" tab.

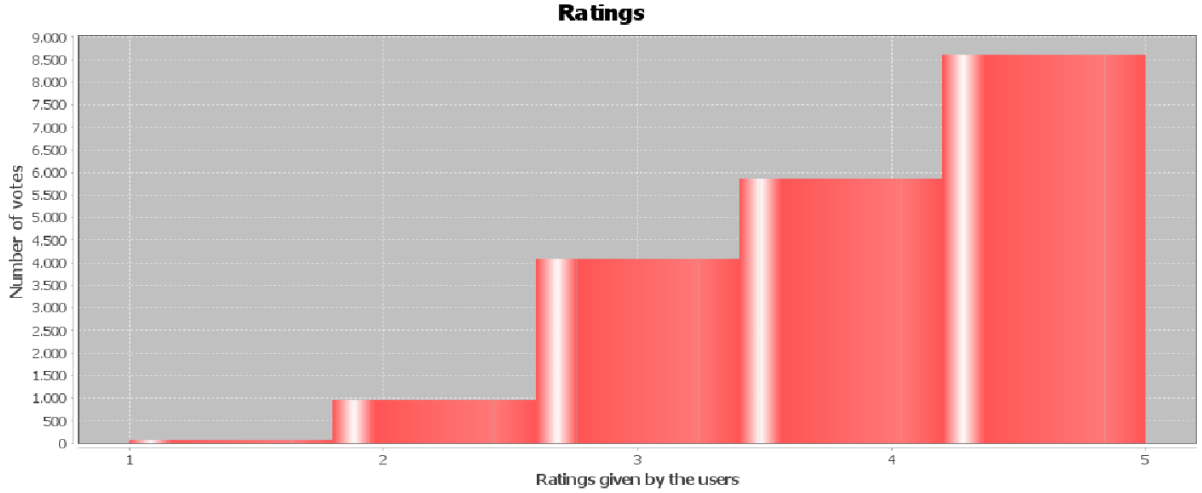

Figure 46: Ratings from the new dataset

The output file of the ratings has no context, as can be seen in Fig. 47.

```
userID;itemID;rating
1;190;5
1;75;4
1;168;3
1;92;5
1;223;5
1;101;5
1;172;5
...
```

Figure 47: Ratings CSV

5.6. Generating and exploiting a synthetic dataset without context

In this last example we are going to create a completely synthetic dataset with no previous data, just as the first example in section 5.1, but without any context. This dataset will focus on a restaurant recommendation scenario for mobile users located in the state of California. The schemas of users and types of items considered are defined as follows:

- **Users**: *age, gender, occupation*.
- **Restaurants**: *web_name, address, province, country, phone, weekday_is_open, hour, type_of_food, card, outside, bar, parking, reservation, price, quality_food, quality_service, quality_price, global_rating*.

We open a new workspace:

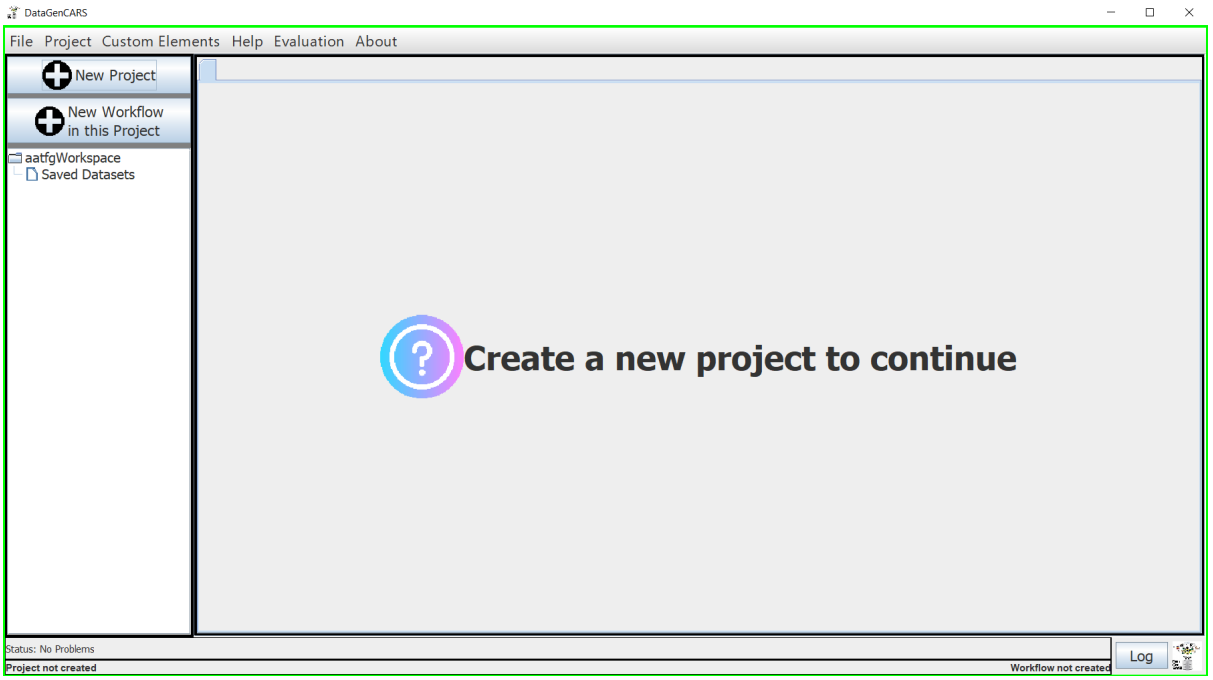

Figure 48: New workspace

To start we create a new project, and once created we create a new workflow in that project. With the workflow created, we name it and indicate that we want to use the *"Synthetic dataset generation from schemes"* generation. Generations can be added to the workflow by clicking the arrow icon pointing to the right. The "*Create dataset with context*" checkbox must be unchecked, so the dataset to be generated does not .

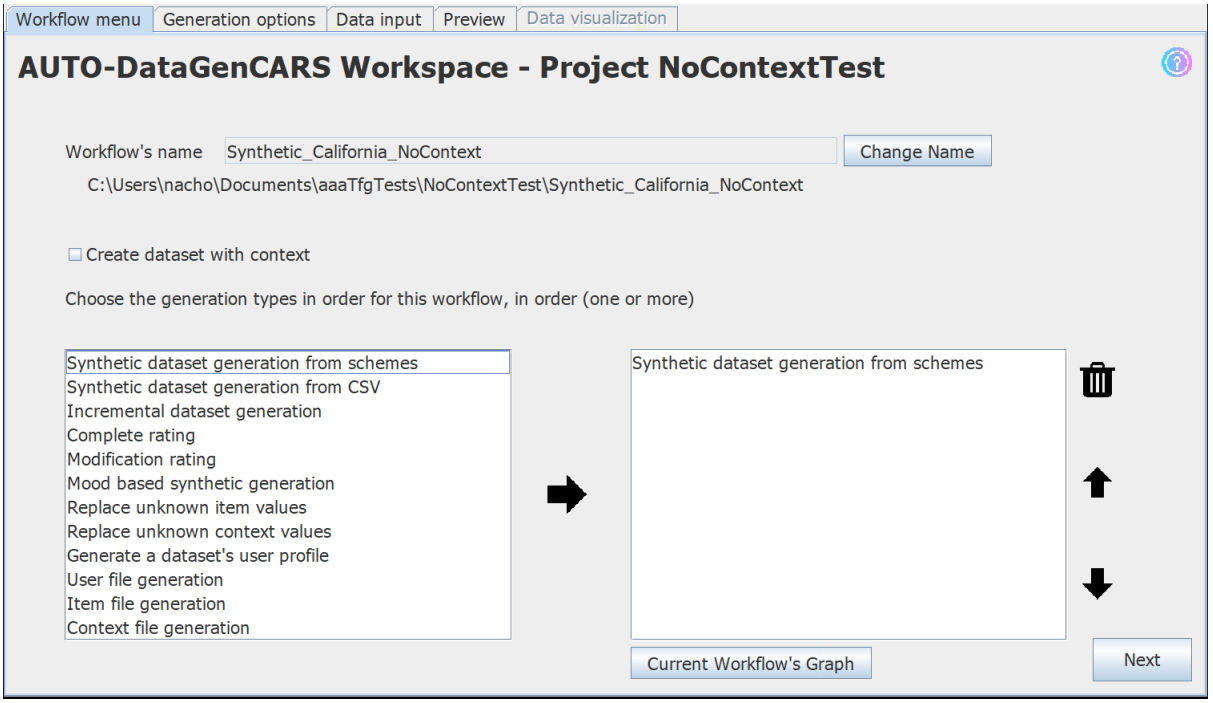

Figure 49: Workflow created

This scenario will consist of *943 users* and *1682 items* (which in this case will be restaurants). We advance, either with the next button or by clicking on the tab names, to the *"Generation options"* tab and enter these figures in the general configuration tab.

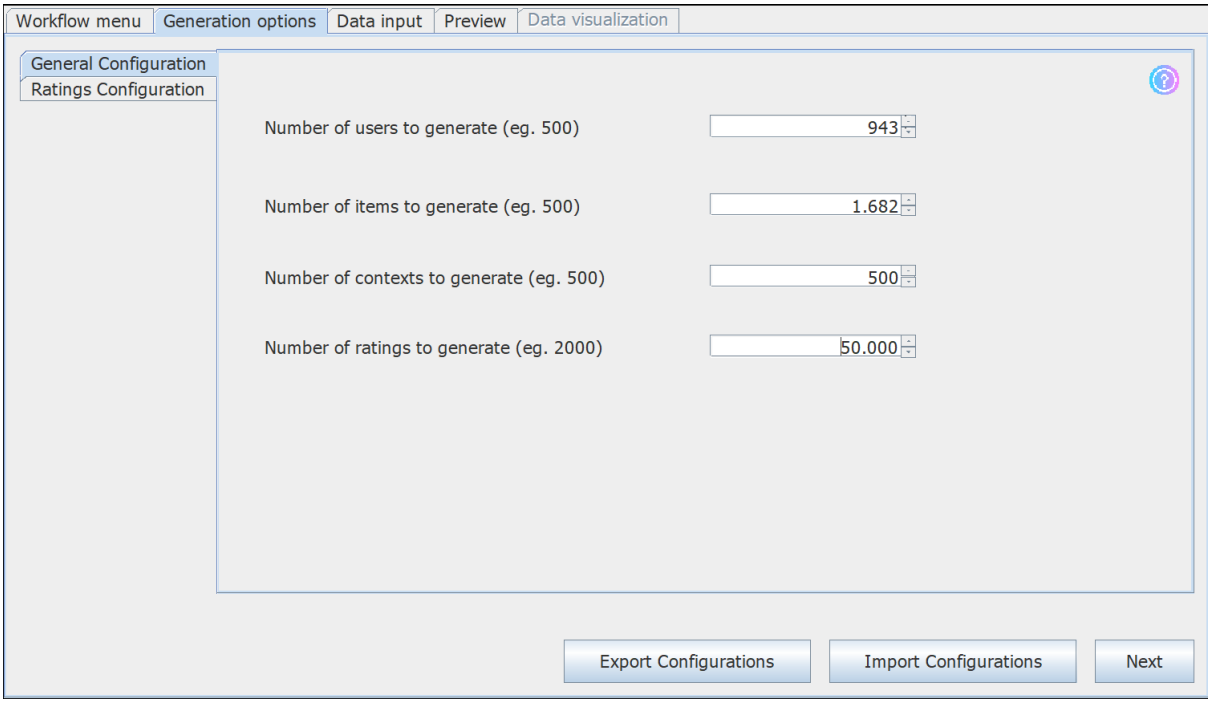

Figure 50: Introducing the general configuration

Specifically, we want to synthetically generate ratings whose values are between 1 and 5, and also labeled with a date, which will be in the range from 1980 to 2020. We enter these data in the ratings configuration tab.

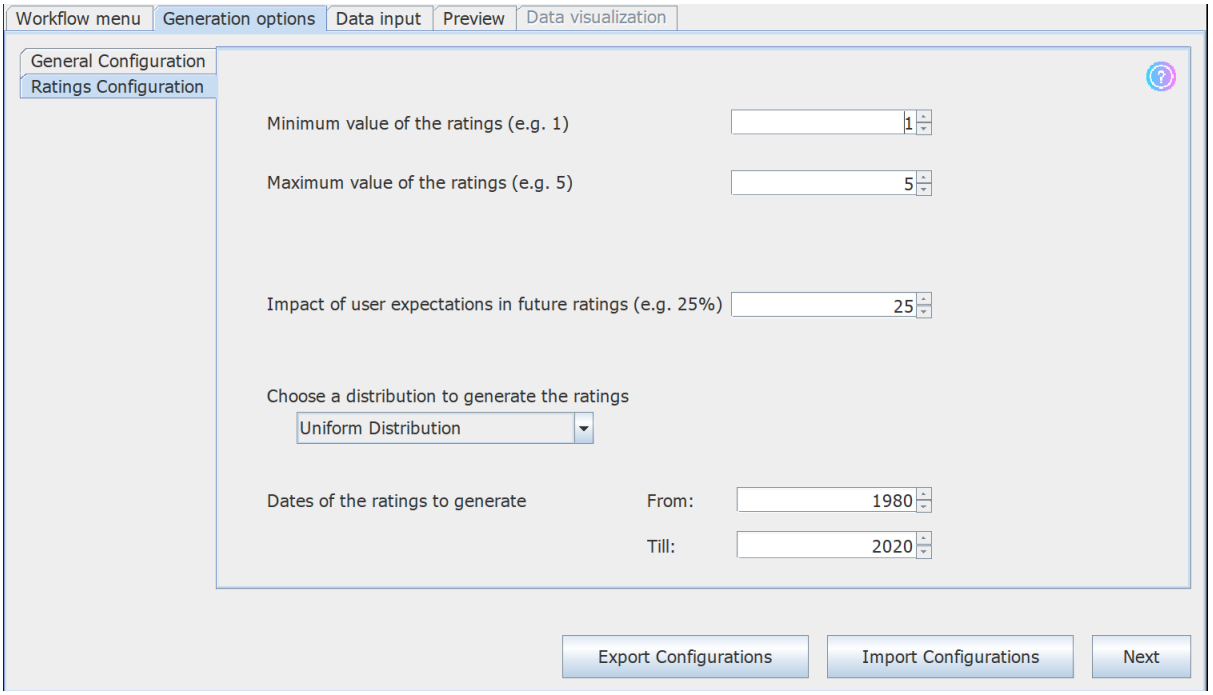

Figure 51: Introducing the ratings configuration

Next it is necessary to create the schemes of the users and the restaurants. To do this, we enter the *"Data input"* tab and begin to create different attributes, such as the following one that represents the age of a user:

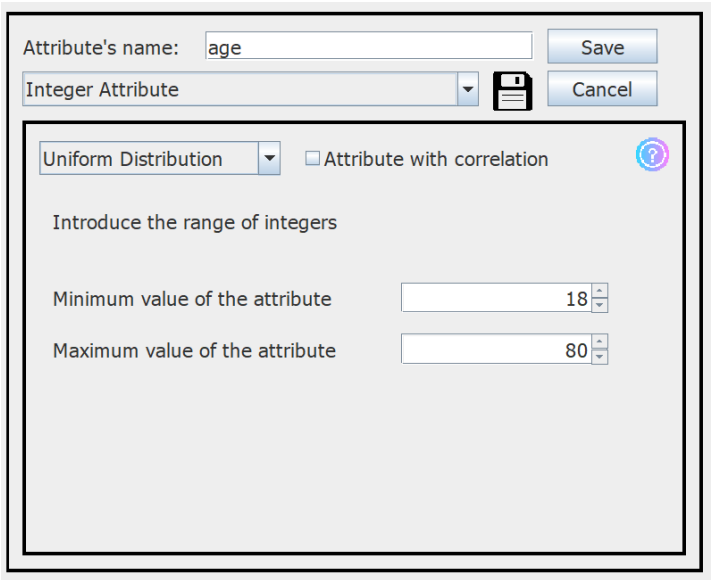

Figure 52: Age attribute

Once the necessary attributes have been created, the users' scheme would look like Fig. 53.

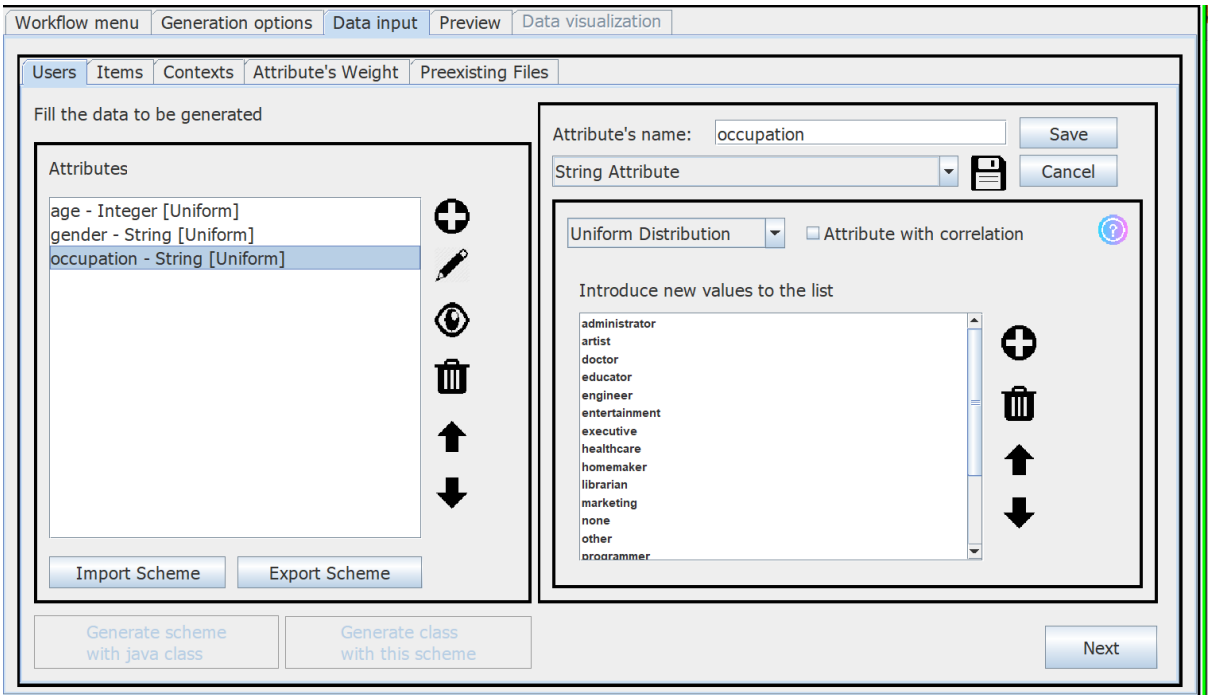

Figure 53: User's scheme

And scheme of the restaurants would look like Fig. 54.

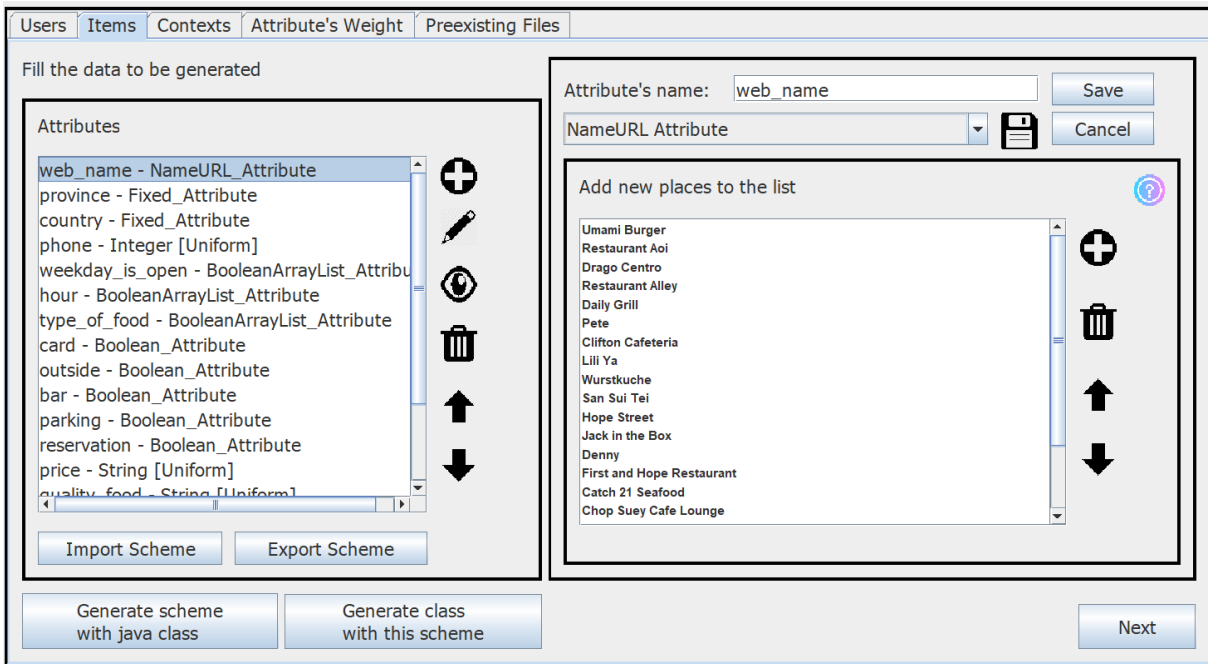

Figure 54: Restaurant's scheme

Next, we move on to the *"Attribute's weight"* tab, in which we select which attributes of the restaurants that are most relevant, and if they have a higher or lower ranking order.

We select the attributes parking, price, quality_food, quality_service for restaurants.

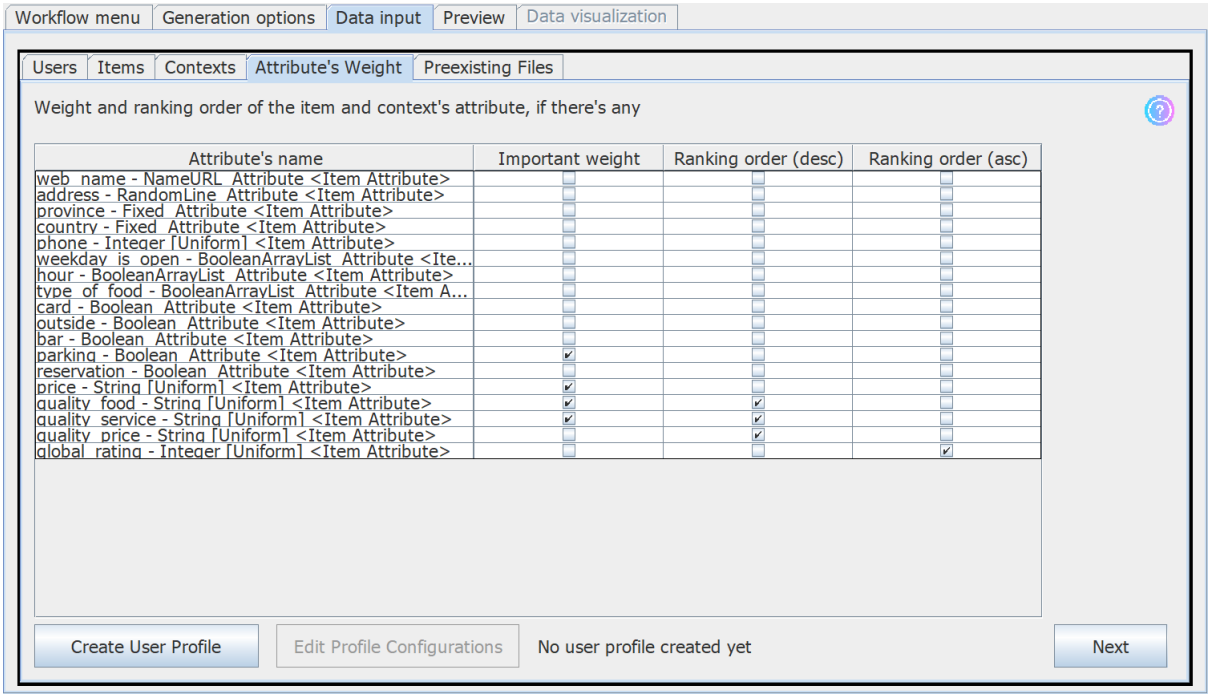

Figure 55: Attribute's weight

With these attributes selected we are going to create 7 different user profiles, and we are going to give different weights to each attribute according to the profile, remembering that the sum of all the weights must equal one (we can press *"Weight readjustment"* so that the weights are readjusted, or we can leave it as we want that they will be readjusted when executing). The created profiles would look like Fig. 56.

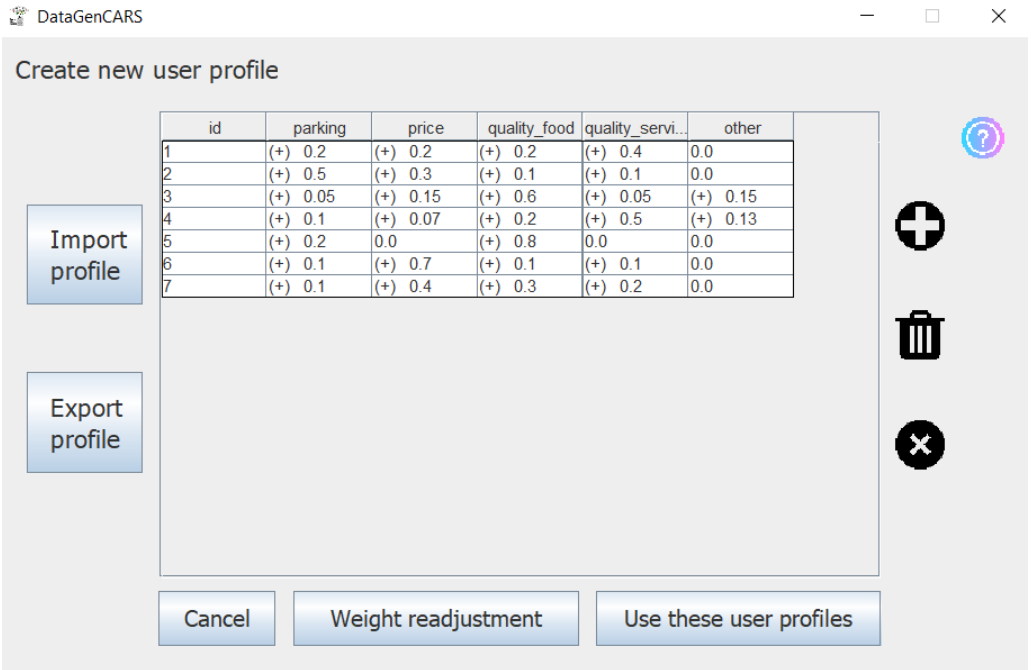

Figure 56: User profiles

In the next tab, *"Preexisting Files"*, nothing is needed to run this generation so we can move on to the next one.

In the last tab before execution, *"Preview"*, we observe that the attributes of the schemes are correct and that we are not missing any more data, so we can proceed to run the generation. When finished, we automatically advance to the *"Data visualization"* tab, where we can observe different statistics about the generated dataset (Fig. 57), as well as being able to download it in different ways, save it in the dataset or evaluate it.

Figure 57: Data visualization tab

The ratings CSV file generated without context looks like Fig. 58.

```
userID;itemID;rating;timestamp;unixTime
1;854;2;1980-01-01 19:00:45.906;315597645
1;109;1;1980-10-03 07:48:00.943;339403680
1;1535;2;1981-07-05 21:35:15.954;363209715
1;1493;4;1982-04-07 10:22:30.96;387015750
1;753;3;1983-01-07 22:09:45.965;410821785
1;976;3;1983-10-10 10:57:00.973;434627820
1;116;3;1984-07-12 00:44:15.98;458433855
1;396;4;1985-04-13 13:31:30.987;482239890
1;701;3;1986-01-14 01:18:45.992;506045925
1;184;3;1986-10-16 14:06:00.996;529851960
1;1346;2;1987-07-19 03:53:15.007;553657995
...
```

Figure 58: Ratings CSV without context

References

[1] Sergio Ilarri, Raquel Trillo-Lado, Ramón Hermoso.

Datasets for Context-Aware Recommender Systems: Current Context and Possible Directions.

First Workshop on Context in Analytics (CiA 2018), in conjunction with the 34th International Conference on Data Engineering (ICDE 2018), Paris (France), IEEE Computer Society, Electronic ISBN 978-1-5386-6306-6, Print on Demand (PoD) ISBN 978-1-5386-6307-3, ISSN 2473-3490, pp. 25-28, April 2018. (DOI: 10.1109/ICDEW.2018.00011)

[2] María del Carmen Rodríguez-Hernández, Sergio Ilarri, Ramón Hermoso, Raquel Trillo-Lado. *DataGenCARS: A Generator of Synthetic Data for the Evaluation of Context-Aware Recommendation Systems*. Pervasive and Mobile Computing, ISSN 1574-1192, volume 38, part 2, pp. 516-541, Elsevier, July 2017. Special Issue on Context-aware Mobile Recommender Systems.

(DOI: 10.1016/j.pmcj.2016.09.020)

Acknowledgments

Work developed at the University of Zaragoza within the COSMOS research group. We thank the support of the project TIN2016-78011-C4-3-R (AEI/FEDER, UE), the Government of Aragon (group reference T64_20R, COSMOS group), and previously the project TIN2013-46238-C4-4-R (AEI/FEDER, UE) and DGA-FSE (group reference T35_17D and T64_20R, COSMOS group).