CET 1: Recommenders and clustering

Introduction

In this evaluation test we will study how to apply recommendation and grouping techniques using a dataset of videogame ratings.

In this statement the following general masters competences are tackled:

- Ability to project, calculate and design products, processes and facilities in all areas of computer engineering.
- Ability for mathematical modeling, calculation and simulation in technology centers and business engineering, particularly in research, development and innovation tasks in all areas related to computer engineering.
- Ability to apply the knowledge acquired and solve problems in new or unfamiliar environments within broader and multidisciplinary contexts, being able to integrate this knowledge.
- Possess skills for continuous, self-directed and autonomous learning.
- Ability to model, design, define architecture, implement, manage, operate, manage and maintain applications, networks, systems, services and computer content.
- The specific competences of this subject that are worked on in this test are:
- Understand what machine learning is in the context of artificial intelligence
- Distinguish between different types and methods of learning
- Apply the techniques studied to a real case

Goals

In this CET the concepts of the syllabus related to recommendation and grouping will be practiced, in a clearly practical aspect, framed in the use of Python libraries.

CET Description

Data

A data file called DATASET.CSV is attached to this assignment. Each line in this file contains a rating of videogame made by a user. In particular, each line contains the following three comma-separated elements:

- The user who made the rating. The user is expressed by the character U followed by a three-digit number.
- The video game being rated. The video game is expressed with the character V followed by a three-digit number.
- The rating made by the user to the video game. The rating is an integer between zero (the user did not like the videogame at all) and 5 (the user really enjoyed the videogame).

This dataset is synthetic and has been constructed randomly as follows. First, a uniform random rating from -5 to +5 was generated for all possible combinations of users and videogames. Secondly, all combinations with a negative rating were removed, thus simulating user-videogame combinations that have no rating.

This random generation procedure is irrelevant unless you are explicitly asked to take it into account. If you are not explicitly asked about the synthetic data generation process, you should work as if DATASET.CSV is data created by actual users rating videogames.

Exercises

To solve this CET you need to program in Python and use libraries such as NumPy or Surprise. Therefore, you will need to check their documentation. You can also use and modify the sample programs provided with the course material.

In all activities you must clearly explain your answer. In addition, every program you are asked to create must be in a .py (Python) file that can be directly executed regardless of the other solutions. If this means that you have functions, methods, ... repeated between files, there is no problem.

In order for these programs access DATASET.CSV, you must assume that it is in the same folder as the program.

The name of each Python file you submit must necessarily be 'solxy.py', where x is the activity number and y is the specific section. For example, sol1a.py for Activity 1-a.

All these files must be submitted in a ZIP file together with the PDF containing the answers and explanations for each section of the CET.

Exercise 1

In this activity you will need to do some programs that will help you understand the data. Specifically, you are asked the following:

a) Create a program that determines how many users have rated each video game. With this program, determine which video game has received the lowest amount of ratings (and how many), which video game has received the highest amount of ratings (and how many) and what is the average number of ratings that the video games have received. Note that this question refers to the average number of ratings that video games received, not the average rating.

b) Create a program that determines how many video games each user has rated. With this program determine who has rated the least video games (and how many), who is the user who has rated the most video games (and how many) and what is the average amount of video games that users have rated.

c) Based on these results, do you think it would be necessary to pre-process the data in some way before building a video game recommender?

d) What do you think would be the main differences between a real video game rating dataset and the synthetic data you are using? Do you think these differences would affect a video game recommender?

Exercise 2

In this activity you will work on the concept of similarity. Regardless of your response to sections c) and d) of Activity 1, here you must use the raw data in DATASET.CSV.

a) Program a function (or method, ...) that computes the Euclidean similarity between two users. You only have to consider video games that have been rated by both users, discarding those that have only been rated by one and those that have not been rated by any of them. Make a program that, using the mentioned function, determines which two users are the most similar and what is the similarity between them.

b) Program a function (or method, ...) that computes the Euclidean similarity between two users. Unlike the previous function, in this case you should consider video games rated by at least one of the two users and discard only those that have not been rated by any of them. If a video game has only been rated by one user, assume that the rating of the user who has not voted is a neutral rating of 2.5. Make a program that, using the mentioned function, determines which two users are the most similar and what is the similarity between them.

c) Repeat section a) but now using the Pearson similarity instead of the Euclidean one.

d) Repeat section b) but now using the Pearson similarity instead of the Euclidean one.

e) Analyse the results obtained in the previous sections. Which of the four options do you think is the best one?

Exercise 3

In this section you will work with the Surprise library.

a) Make a program that uses all the data in DATASET.CSV to train the Surprise algorithms SVD, KNNBasic, KNNWithMeans and NormalPredictor. Once trained, use each of the trained algorithms to predict the score that the user U098 would give to the video game V077. Compare that prediction to the actual rating. Which algorithm is closest to the actual rating? What do you think is the reason for that?

b) Make a program that, for the same algorithms in the previous section, does cross-validation using 5 folds and computes the RMSE. Which algorithm gives the best results? Does it match the algorithm that worked best in the previous section? Explain clearly your answer.

c) Analyse the results obtained in the previous sections of this activity. Do you think that the distribution of the input data and how the synthetic data set is constructed has any visible effect on the results? What kind of results do you think would appear if real video game ratings data was used instead?

Exercise 4

Check out the Surprise documentation and answer the following questions. You must be clear and concise in your answers.

a) Briefly explain the KNNBasic, KNNWithMeans, KNNWithZScore and KNNBaseline algorithms, emphasizing their differences, and reasoning about the kind of data for which each of them seems to be best suited.

b) Each of the above algorithms allows to define the measure of similarity to use. Similarity measures available are cosine, msd, pearson and pearson_baseline. Briefly explain how each of these similarities is computed and with which kind of data should be used.

Resources

To carry out this CET the following resources are required

- **Basic**: The attached data file.
- **Complementary**: Course material and example programs.

Assessment criteria

Te valuation of each exercise is as follows:

- Exercise 1: 3 points
- Exercise 2: 3 points
- Exercise 3: 3 points
- Exercise 4: 1 points

Within an exercise, all sections score equally.

The source code of your solutions must be included in the delivery.

In addition, each exercise must be answered by reasoning the actions taken to the code and why, and the reflections that we extract from the results of operations.

Format and deadline

The CET will be delivered within the deadline. The solution should consist of a zip file containing a report in pdf format and the files in python format (* .py). The naming convention for the Python files to deliver was discussed in the Exercises section.

Note: **Intellectual property**

It is often inevitable, when producing a multimedia work, to make use of resources created by third parties. It is therefore understandable to do so within the framework of the practice of the studies of the Multimedia Degree, always and this is clearly documented and does not suppose plagiarism in practice.

Therefore, when presenting a practice that makes use of external resources, a document detailing all of them will be presented along with it, specifying the name of each resource, its author, the place where it was obtained and its legal status. : if the work is protected by copyright or is covered by another user license (Creative Commons, GNU, GPL ...). The

student must ensure that the license does not specifically prevent its use within the framework of the practice. If you can not find the corresponding information, you must assume that the work is protected by copyright.

They must also attach the original files when the works used are digital, and their source code if applicable.

Another point to consider is that any practice that makes use of resources protected by copyright can in no case be published in Mosaic, the magazine of the Degree in Multimedia at the UOC, unless the owners of the intellectual rights give their explicit authorization.