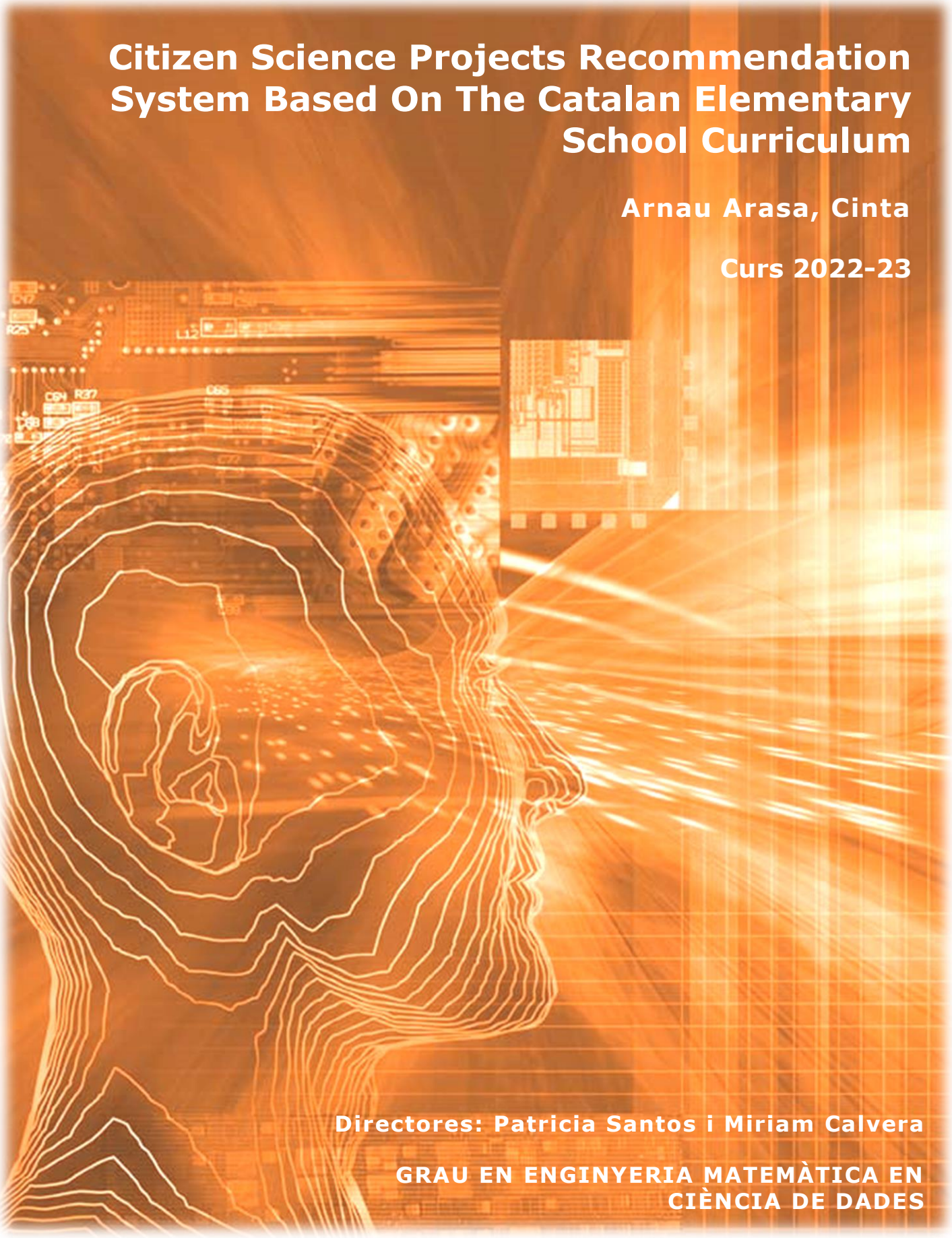
Citizen Science Projects Recommendation System Based On The Catalan Elementary School Curriculum

Arnau Arasa, Cinta

Curs 2022-23



Directores: Patricia Santos i Miriam Calvera

GRAU EN ENGINYERIA MATEMÀTICA EN CIÈNCIA DE DADES

upf. Universitat Pompeu Fabra Barcelona
Escola d'Enginyeria

Treball de Fi de Grau

[Pàgina en blanc]

[S'han marcat els apartats del treball que haurien d'anar en pàgina senar o dreta de la publicació (situació de pàgines que ajuda a estructurar i a donar prioritat formal als diferents apartats).

Hi ha apartats on no s'indica res. En aquests cas l'estudiant pot triar la situació, en pàgina parell o senar, segons li convingui per al compaginat final.

Es recomana que les pàgines blanques no portin número de foli (es compten, però el número no s'imprimeix)]

Si s'imprimeix a simple cara no s'han de deixar pàgines en blanc

To my younger sister Núria
who I love the most

[Pàgina en blanc]

Acknowledgements

Text dels agraïments [12 punts]

[Pàgina en blanc]

Abstract

Resum del treball en la llengua que es presenti [12 punts]
Extensió màxima recomanada: 150 paraules

Resumen [en una 2a llengua. Ex. Resumen]

Resum del treball en una llengua diferent a la utilitzada [12 punts]
Extensió màxima recomanada: 150 paraules

Resum [en una 3a llengua. Ex. Abstract]

Resum del treball en una llengua diferent a la utilitzada [12 punts]
Extensió màxima recomanada: 150 paraules

*[Aquest apartat, **sempre** hauria de començar en pàgina senar, pàgina dreta de la publicació]*

[Pàgina en blanc]

TABLE OF CONTENTS

[Aquest apartat també ha de començar en pàgina senar]

Table of figures [opcional]

1. INTRODUCTION

1.1 Context

"Citizen Science refers to the general public engagement in scientific research activities when citizens actively contribute to science either with their intellectual effort or surrounding knowledge or with their tools and resources" [9].

1.2 Motivation

(…)

1.3 Objectives

The main objective of the project is to analyze the citizen science projects in Barcelona and Spain, as well as the Catalan elementary school curriculum, in order to create a recommendation system that can be used as an educational tool to propose CS projects that can be used in schools to achieve the key competences stated in the curriculum. In a more systematic way, the objectives of this project are:

- Studying the Barcelonian and Spanish Citizen Science platforms.
- Designing a tool to extract information from the Barcelonian platform "Oficina de la Ciència Ciutadana"[1] and the Spanish platform "Observatorio de la Ciencia Ciudadana en España"[2] by programming a crawler and creating the corresponding databases.
- Analysing the Citizen Science projects extracted from both the Barcelonian and the Spanish platforms.
- Analysing the Catalan elementary school curriculum and the key competences stated in it.
- Designing a recommendation system that recommends CS projects based on the key competences of the curriculum.
- Extracting meaningful conclusions regarding the use of citizen science as an educational tool.

[Cada capítol ha de començar en pàgina senar]

[1] "Observatorio de la Ciencia Ciudadana en España" (2023, May 20) https://ciencia-ciudadana.es/.
[2] "Oficina de la Ciència Ciutadana" (2023, May 20) https://www.barcelona.cat/barcelonaciencia/es/.

2. STATE OF THE ART

2.1 Citizen Science

2.1.1) Citizen Science definition and history

2.1.2) Citizen Science projects

2.1.3) Citizen Science as an educational tool

2.2 Catalan elementary school curriculum

"Elementary school education must prepare students to provide innovative responses in a constantly changing and evolving society. The principles of equity, quality, and excellence determine and condition educational action since teaching and learning processes need to be personalized to the maximum extent and take into account the diversity of all students within an inclusive system" [7].

2.2.1) Analysis of the Catalan elementary school curriculum

The Catalan elementary school curriculum[3] describes the objectives, contents and evaluation criteria of each area and subject. It contains information about the key competences and the competency indicators at the end of stage, which constitute the profile upon completion.

"The curriculum must enable the development of personal and professional life projects for everyone based on educational success, counting on the support, involvement, and participation of families. It should also facilitate access to subsequent educational processes and lifelong learning" [8]. The goal of the curriculum is to achieve the key competences that are developed throughout the knowledge areas and subjects.

2.2.2) Key Competences

"The key competences are the achievements that are considered essential for the students to progress successfully in their educational journey and to face the main global and local challenges and demands" [8]. The full list of the key competences stated in the Catalan elementary school curriculum can be checked in the Appendix B.

2.3 Natural Language Processing (NLP)

2.3.1) Term Frequency-Inverse Document Frequency (TF-IDF)

2.3.2) Semantic similarity

[3] DECRET 175/2022, de 27 de setembre, d'ordenació dels ensenyaments de l'educació bàsica.

2.4 Recommendation System

"A recommendation system (or recommender system) is a class of machine learning that uses data to help predict, narrow down, and find what people are looking for among an exponentially growing number of options" [10].

Recommender systems have many benefits, such as improving retention, increasing sales, helping to form customer habits and trends, increasing user satisfaction, speeding up the pace of work and boosting cart value. The main reason for which a recommender system is the chosen tool for this projects is to provide the most adequate citizen science projects based on the key competences stated in the Catalan elementary school curriculum (see Appendix B), given that its use will speed up the pace of work since the process of having to look through a dense amount of CS projects to find the best one will be erased and substituted by the process of only having to introduce the objective to the system.

There are many companies that use recommender systems in order to achieve the previously mentioned benefits. Such companies are Amazon.com, Netflix, Spotify and LinkedIn, among many others.

- Amazon.com uses an item-to-item collaborative filtering recommender system in order to mainly boost the number of items purchased and hence the cart value and increase sales. "According to McKinsey, 35% of Amazon purchases are thanks to recommendation systems" [11].
- Netflix uses a collaborative filtering recommender system, where it shows some category and movie/show suggestions based on the user's preferences and other shows previously watched by the user. "The same McKinsey study we mentioned above highlights that 75% of Netflix viewing is driven by recommendations" [11].
- The Spotify music engine uses three different techniques: collaborative filtering, natural language processing (NLP) and audio file analysis [11].

[Cada capítol ha de començar en pàgina senar]

3. DESIGN AND IMPLEMENTATION

In this section are shown the different tools and platforms containing citizen science projects used in the making of this project. The platforms used are the Spanish platform "Observatorio de la Ciencia Ciudadana en España"[4] and the Barcelonian platform "Oficina de la Ciència Ciutadana"[5]. There is an explanation of the data structure of each of these platforms and the extraction process of all the data related to the citizen science projects.

3.1 Environment set-up and selection of tools

3.1.1) Tools used

In order to extract the data from the two platforms of Citizen Science projects and create the recommendation system various tools have been used throughout the entirety of the thesis process. The following descriptions of these tools aim to clarify why they are necessary and demonstrate their application within the project.

3.1.1.1) Python

Python[6] is a high-level programming language that is widely used for general-purpose programming. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python's versatility and ease of use have made it a go-to language for a wide range of applications, including web development, data analysis, machine learning, artificial intelligence, scientific computing, and automation [1]. For this reason, and given the fact that Python has been the most used programming language throughout the degree, it has been the chosen language for the Citizen Science projects' extraction and the building of the recommendation system.

The libraries that have been used in the scripts (both the project extraction and the recommendation system) are the following:

- Pandas[7]: "It is a powerful open-source Python library widely used for data manipulation, analysis, and exploration". It provides highly efficient data structures and data analysis tools, making it an essential tool for working with structured data, creating DataFrame objects to manipulate data with integrated indexing, and much more functionalities to work with data sets [2].
- Request[8]: "It is a popular tool used for making HTTP requests and interacting with web services. It provides a convenient and user-friendly interface to send HTTP requests, handle responses, and perform various operations related to web communication" [3].
- BeautifulSoup[9]: "Python library used for web scraping and parsing HTML or XML documents" [4].
- NLTK[10]: "The Natural Language Toolkit (NLTK) is a popular Python library widely used for natural language processing (NLP) tasks. offers a wide range of

[4] "Observatorio de la Ciencia Ciudadana en España" (2023, May 20) https://ciencia-ciudadana.es/.
[5] "Oficina de la Ciència Ciutadana" (2023, May 20) https://www.barcelona.cat/barcelonaciencia/es/.
[6] Python Website (2023, May 20) https://www.python.org/.
[7] Pandas Documentation (2023, May 20) https://pandas.pydata.org/docs/.
[8] Request Documentation (2023, May 20) https://docs.python-requests.org/en/latest/.
[9] BeautifulSoup Documentation (2023, May 20) https://www.crummy.com/software/BeautifulSoup.
[10] NLTK Documentation (2023, May 31) https://www.nltk.org/.

functionalities for tasks such as tokenization, stemming, lemmatization, part-of-speech tagging, parsing, semantic reasoning, and more" [13].

- Re[11]: "The re (regular expression) library is a built-in Python library that provides support for regular expressions, which are powerful tools for pattern matching and text manipulation" [14].
- TfidfVectorizer[12]: "Class provided by the scikit-learn library. It is specifically designed for text feature extraction and is based on the term frequency-inverse document frequency (TF-IDF) weighting scheme" [15].
- Cosine_similarity[13]: "Method provided by the scikit-learn library. It is used to compute the cosine similarity between pairs of vectors or matrixes, typically representing feature vectors in a high-dimensional space" [16].
- WordCloud[14]: "Python library used for generating word clouds, which are visual representations of text data where the size of each word corresponds to its frequency or importance within the text" [17].
- Matplotlib[15]: "Python library for creating static, animated, and interactive visualizations. It provides a wide range of plots, charts and graphs" [18].

3.1.1.2) Jupyter Notebook

"Jupyter Notebook[16] is an open-source web-based interactive computing environment widely used for data analysis, visualization, and prototyping" [5]. It supports many programming languages (including Python) and allows for interactive data analysis and visualization, enhancing the exploratory data analysis process. Since Jupyter Notebook has been the most used framework throughout the degree, it has been the chosen one for the creation

3.1.1.3) BeautifulSoup

"BeautifulSoup is a popular Python library used for web scraping and parsing HTML or XML documents" [4]. It provides a convenient way to extract and navigate data from web pages by simplifying the process of locating and manipulating elements within the document structure. BeautifulSoup can be combined with other Python libraries, such as requests, to fetch web page content and then parse and extract the desired information.

3.1.1.4) GitHub

"GitHub[17] is a web-based platform for version control and collaborative software development". It provides a centralized hub where developers can store, manage, and collaborate on their code repositories [6]. A GitHub repository[18] has been created to store and manage all the scripts with the corresponding code of this thesis.

3.1.2) Jupyter Notebook scripts

In order to extract all the information about the Spanish and Catalan Citizen Science projects, the script "projects_extraction.ipynb" has been generated. All the information

[11] Python Documantation. re (2023, May 31) https://docs.python.org/3/library/re.html.
[12] Scikit-learn Documentation. TfidfVectorizer (2023, May 31) https://scikit-learn.org/TfidfVectorizer.
[13] Scikit-learn Documentation. cosine_similarity (2023, May 31) https://scikit-learn.org/cosine_similarity.
[14] Wordcloud Documentation (2023, May 31) https://amueller.github.io/word_cloud/.
[15] matplotlib Documentation. (2023, May 31) https://matplotlib.org/.
[16] Jupyter Notebook Website (2023, May 20) https://jupyter.org/.
[17] GitHub Website (2023, May 20) https://docs.github.com/en.
[18] GitHub repository https://github.com/Cintaa1223/TFG.

regarding the CS projects is extracted using the Python library BeautifulSoup as it will be later explained.

The script named “recommendation_system.ipynb” joins the information extracted from both the Spanish and the Barcelonian platform and the recommendation system to recommend CS projects based on the key competences stated in Catalan elementary school curriculum is created. The process to create the system is explained in section 4.2.

These Jupyter notebook scripts can be found in the Github repository so that anyone interested in Citizen Science or any teacher wanting to use the recommendation system can consult them (see Appendix A to consult the locations of the scripts).

3.2 Spanish platform: “Observatorio de la Ciencia Ciudadana en España”

3.2.1) Platform data structure

The platform distinguishes between initiatives and resources. Initiatives can be citizen science projects, persons, or institutions. The initiatives can be found by field of knowledge, type (citizen science projects, persons, or institutions), or text/keyword. All three types share the same structure.

The initiatives all follow the same structure, no matter their type. The fields each initiative has are the following: project ID, project name, main organisation, subtitle, type of project, keywords, start date, end date, public, province, participants, URL, aim, project description, responsible entity, founding team, more entities, how to participate, results, results link, impact, impact examples, and motivation (why use citizen science).

3.2.1) Extracting the data from the Spanish platform

All the citizen science projects contained in the Spanish platform have been extracted directly from the website after having made sure that it allowed the automatic extraction through the use of robots.
In order to extract all the information from the Spanish platform, it is needed to create a crawler. The chosen pyhton library to do the web scraping is BeautifulSoup, which will make it possible to get the desired data throughout the examination of the website elements. The process to do so in a Jupyter Notebook environment, as it is done in the “projects_extraction.ipynb” script, is the following:

1. Install and import the BeautifulSoup library along with the other needed libraries pandas and request.
2. Sending an HTTP request to the web page by using the ‘requests.get()’ function to send a GET request to the URL of the Spanish website. The response from the server can be used to extract the HTML content of the page.

```
url = "https://ciencia-ciudadana.es/proyecto-cc/"
web_name = 'Observatorio de la Ciencia Ciudadana en España'

# Adding headers to the rquest
headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) '
                         'AppleWebKit/537.36 (KHTML, like Gecko) '
                         'Chrome/58.0.3029.110 Safari/537.36'}

response = requests.get(url, headers=headers)
```

Figure 1. Code to send HTTP request.

3. Parsing the HTML content. The HTML content obtained from the web page is passed to the BeautifulSoup constructor to create a BeautifulSoup object, which allows users to navigate and search through the HTML structure of the page.

```
soup = BeautifulSoup(response.content, "html.parser")
```

Figure 2. Code to parse HTML content.

4. Extracting data. This involves finding specific HTML elements such as tags, classes, or IDs, and accessing their attributes or text content. In order to do so, methods like 'find()' or 'find_all()' are used to locate and extract the desired data.

   To see the HTML structure of the webpage: "Ajustes" → "Más herramientas" → "Herramientas para desarrolladores"[19].

   - First of all, the home page of the Spanish platform contains all the projects which have to be accessed in order to extract the needed information. To do so, we first find where the URL to each project is found in the HTML structure and create a list that contains all the links to the citizen science projects.

```
# Find all elements with class name "underline"
underline_elements = soup.find_all('img', {'decoding': 'async'})

links = []
# Extract the links from the parent elements
for element in underline_elements:
  parent_a_tag = element.find_parent('a')
  if parent_a_tag and 'href' in parent_a_tag.attrs:
    link = parent_a_tag['href']
    links.append(link)
```

Figure 3. Code to extract projects' URLs.

   - Now we access each of the projects' URLs stored in the links array the same way as described in steps 2 and 3. The array can be iterated to get all the necessary information of each project. This needed data includes the following fields: *'Project Name', 'Project Link', 'Project Scope', 'Project Goal', 'Project Description', 'Project Entity/Scientist', 'How To Join', 'Necessary Equipment', 'Initial Date', 'Final Date', 'Public Type', 'Location (Province)', 'Number of Participants', 'Results', 'Link to Results', 'Project Impact', 'Why Using CC?', 'Citizen Science Web Name', 'Citizen Science Web Link'*.
     Given that each information to be extracted follows the same HTML structure, the function 'get_complete_section(project_soup, dtbf)' has been created to make it simple.

```
def get_complete_section(proj_soup, dtbf):
  proj_seg = proj_soup.find_all('div', {'class': 'tb-field', 'data-toolset-blocks-field': dtbf})
  return ''.join([seg.text for seg in proj_seg])
```

Figure 4. Function get_complete_section(proj_soup, dtbf).

[19] It can also be done by either right-clicking on any part of the webpage and selecting "Inspect" or by clicking 'fn'+'f12'.

```python
def get_project_info1(project_link, main_url, main_name):
  proj_info = ['', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '']

  response = requests.get(project_link, headers=headers)
  proj_soup = BeautifulSoup(response.content, "html.parser")
  proj_title = proj_soup.find('h1', {'class': 'entry-title'}).text
  proj_scope = get_complete_section(proj_soup, "28515d8ce1fa37a6527af15754983e83")
  proj_goal = get_complete_section(proj_soup, "ba74407e8b20cac888e283e8576140f9")
  proj_desc = get_complete_section(proj_soup, "7147dde37d9b86b4d4a2dc89b9c12945")
  proj_entity = get_complete_section(proj_soup, "486908e9509d0a139e234cbfb7f8f47d")
  proj_join = get_complete_section(proj_soup, "687686b4b9d6f02068b541e6bc6f2812")
  proj_equip = get_complete_section(proj_soup, "cbcd66c16990127ea3c06cebc87c17f0")
  proj_ini = get_complete_section(proj_soup, "3d66c4d00ab960c97a58ac752fe406f4")
  proj_end = get_complete_section(proj_soup, "9a00b649618c1cd0c0cedd6cfc386f09")
  proj_public = get_complete_section(proj_soup, "e9e0eef27b43baa290bdc058d0ec6cee")
  proj_loc = get_complete_section(proj_soup, "05399f19f4997ca4dc2a1f1770db2d80")
  proj_amt_part = get_complete_section(proj_soup, "5ee97fadc3c34313fff123d0cba66a1a")
  proj_results = get_complete_section(proj_soup, "b064a4637177bd64d367dc70864be32c")

  proj_link_res = proj_soup.find_all('div', {'class': 'tb-field', 'data-toolset-blocks-field': "c5d8b6598ffd2835065c71792c189772"})
  proj_link_res = [seg.text for seg in proj_link_res][0]

  proj_impact = get_complete_section(proj_soup, "4297f88ef1601b65151fcf8fc6fccadd")
  proj_useCC = get_complete_section(proj_soup, "974e65ee4de5d532951d366fa43dc400")
```

Figure 5. Code to extract all the necessary fields of a project.

```python
for link in links:
  get_project_info1(link, url, web_name)
```

Figure 6. Code to iterate through the links array.

5. Storing the extracted data in a pandas DataFrame. In the same function 'get_project_info1()' as in *Figure 5*, all the fields extracted are stored in a dictionary so that it can be added as a new row to the created DataFrame 'df1'.

```python
# Create an empty DataFrame with specified columns
df1 = pd.DataFrame(columns=['Project Name', 'Project Link', 'Project Scope', 'Project Goal', 'Project Description',
                            'Project Entity/Scientist', 'How To Join', 'Necessary Equipment', 'Initial Date',
                            'Final Date', 'Public Type', 'Location (Province)', 'Number of Participants', 'Results',
                            'Link to Results', 'Project Impact', 'Why Using CC?', 'Citizen Science Web Name',
                            'Citizen Science Web Link'])
```

Figure 7. Creation of the DataFrame to store the Spanish platform projects.

3.3 Barcelonian platform: "Oficina de la Ciència Ciutadana"

3.2.1) Platform data structure

3.2.1) Extracting the data from the Barcelonian platform

[Cada capítol ha de començar en pàgina senar]

4. RECOMMENDATION SYSTEM

Recommender systems are trained to understand the preferences, previous decisions and characteristics of users and products based on the data about their interactions such as impressions, clicks, likes, and purchases [10]. In general, the idea of a recommendation system is that, given data from user interests, including profiles, browsing behavior, item interaction behavior, ratings about various items, it leverages such data to make recommendations to users about further interesting items.

4.1 Types of recommendation systems

The recommendation systems can be classified in these three broad categories: collaborative filtering, content-based filtering and context filtering.

4.1.1) Collaborative filtering

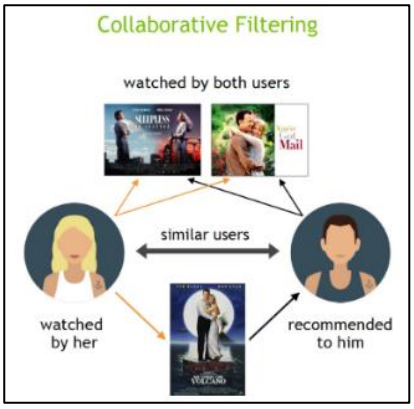


Figure 8. Collaborative filtering.

"Collaborative filtering algorithms recommend items (this is the filtering part) based on preference information from many users (this is the collaborative part). This approach uses similarity of user preference behavior, given previous interactions between users and items" [10]. This technique filters out items that may be liked by a user based on the items liked by similar users. Therefore, this algorithm constantly finds the relationships between the users and in result, it makes the recommendations.

There are two types of collaborative filtering recommendation systems:

- User-based: measures the similarity between target users and other users.
- Item-based: measures the similarity between the items that target users rate or interact with and other items.

4.1.2) Content-based filtering

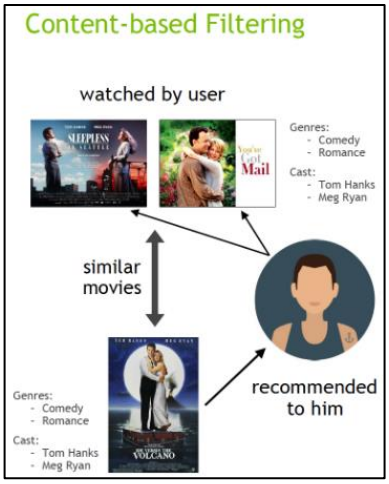


Figure 9. Content-based filtering.

"Content filtering, by contrast, uses the attributes or features of an item (this is the content part) to recommend other items similar to the user's preferences. This approach is based on similarity of item and user features" [10]. In content-based recommendations, users and items are associated with features, which are matched to infer interest. Some of the uses of this recommendation system are recommending other movies with the same director, age, genre, as viewed ones and recommending other products in the same category, brand, color, as purchased ones. This algorithm is able to recommend to users with very particular tastes, recommend new and obscure items, and provide explanations that are easily understandable.

4.1.3) Context filtering

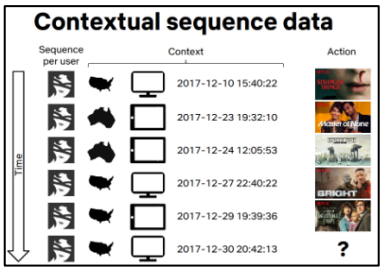


Figure 10. Context filtering.

"**Context filtering** includes users' contextual information in the recommendation process. This approach uses a sequence of contextual user actions, plus the current context, to predict the probability of the next action. In the Netflix example, given one sequence for each user—the country, device, date, and time when they watched a movie—they trained a model to predict what to watch next" [10].

4.2 Design of the Recommendation System

In order to recommend the CS projects obtained from the Barcelonian and Spanish platforms based on the key competences of the Catalan elementary school curriculum, a content-based recommender system has been built using text-similarity.

Previous to the creation of the recommender system, the database with the extracted CS projects had to be created. To do so, the dataframes that stored the projects were download as comma-separated values (CSV) files, which were then imported in the recommender system script. A new column 'Project Full Description' was created by joining the 'Project Description' and the 'Project Goal' columns, which contained the two main parts of the description of the projects. Then, it was decided to eliminate most of the columns of both dataframes, leaving only the following: 'Project Name', 'Project Link', 'Project Scope', 'Project Description', 'Project Goal', 'Project Full Description', 'Citizen Science Web Name', 'Citizen Science Web Link'. This was done to merge both dataframes in one, which would be the final dataframe containing all the CS projects.

4.2.1) Preprocessing of the data

The first step in building a recommender system is preprocessing the data. The process of data preprocessing consists on cleaning and transforming the text data. This includes converting all text to lowercase, removing stop words, removing punctuation and other non-wanted symbols and applying stemming.

To preprocess all the data regarding project descriptions and the input key competence, the function 'build_terms()' has been created. This function receives a line of text as input and returns a list of the words contained in it after having removed the stopwords and all non-wanted symbols, transforming the text to lowercase, tokenizing, and stemming.

```
def build_terms(line):
    stemmer = PorterStemmer()
    stop_words = set(stopwords.words("spanish"))
    line = line.lower()  #Convert to lowercase
    line = line.split()  # Tokenize the text to get a list of terms
    line = [x for x in line if x not in stop_words]  # eliminate the stopwords
    line = [x for x in line if x.startswith(("@", "https://", "$", '#')) != True]
    line = [re.sub('[^a-záéíóúäëïöü]+', '', x) for x in line] # since it's in spar
    line = [stemmer.stem(word) for word in line] # perform stemming
    return line
```

Figure 13. Code to preprocess text data.

All the project descriptions as well as the input key competence are preprocessed by using the mentioned function:

```
KC = input("Please enter the key competence: ")
```

Figure 14. Code to input the key competence.

```
projectsCS_clean['Project Full Description'].apply(build_terms)
KC = build_terms(KC)
```

Figure 15. Preprocessing project descriptions and key competence.

4.2.2) Text embeddings

"Word embedding is the collective name for a set of language modelling and feature learning techniques in natural language processing (NLP) where words or phrases from the vocabulary are mapped to vectors of real numbers. Words that are similar in a semantic sense have a smaller distance between them than words that have no semantic relationship" [12].

The second step is creating text embeddings. This process consists on converting the preprocessed text data into numerical representations that can be used for similarity calculations. The resulting word embeddings can then be combined to represent a document (in this case the project description) as a dense vector.

With the use of the Python library sklearn to import 'TfidfVectorizer', collection of project descriptions can be converted to a matrix of TF-IDF features. This created matrix is a representation where each row corresponds to a project description and each column corresponds to a TF-IDF score of a specific word. This is applied to both the project descriptions and the input key competence:

```
vectorizer = TfidfVectorizer()
text_embeddings = vectorizer.fit_transform(projectsCS_clean['Project Full Description'])
input_embedding = vectorizer.transform(KC)
```

Figure 16. Code to create the text embeddings.

4.2.3) Cosine similarity

The third step is calculating the similarity. For this recommender system, it has been decided to use the cosine similarity as the similarity measure. The cosine similarity calculates the similarity between two document vectors, which in this case would be the key competence and the project description. The similarity measure quantifies how similar the two documents are based on their text embeddings.

Cosine similarity is a metric that measures the similarity between two non-zero vectors. It calculates the cosine of the angle between the vectors, indicating the degree of similarity or relatedness between them [16]. The cosine_similarity function measures the similarity between documents based on their vector representations, such as TF-IDF vectors or word embeddings. It takes as input two arrays representing the vectors or matrices for which the cosine similarity needs to be calculated, and it returns a matrix of similarity scores. In this case, the cosine_similarity function is applied as follows:

```
similarities = cosine_similarity(input_embedding, text_embeddings)
```

Figure 16. Code to calculate the cosine similarity.

Once these three steps are completed, the projects are ranked based on their similarity to the input sentence, which in this case corresponds to the key competence introduced to the recommendation system. When the ranking has been completed, the top 5 projects that are most similar to the input key competence are displayed as recommendations to the user.

4.3 Key Competences

The goal of the recommendation system is to recommend the user the most adequate citizen science projects based on the input key competence. The full list of key competences stated in the Catalan elementary school curriculum can be found in Appendix B.

It has been observed how when a long text input is introduced, such as the whole sentence describing the key competence, the output recommended projects are not related at all with what was introduced to the recommendation system. This is due to two main factors:

- TF-IDF representation. The TF-IDF representation is based on the frequency of words within a document. The fact that the input text is long implies that it contains a higher number of words, from which some of them might occur more frequently, hence dominating the TF-IDF scores. Therefore, the similarity scores between the input text and the project descriptions may be biased toward those few dominant words.
- Cosine similarity. The cosine similarity computes the similarity between two vectors by considering the angle between them. If some words in the input text have higher TF-IDF scores, they can influence the similarity score resulting in redundant recommendations.

An additional reason for this to happen is that most of the project descriptions are relatively short, up to the point that in some cases the input text is longer than the descriptions, which also influences the results of the recommended projects.

There are many approaches to solve this problem, such as limiting the length of the input text, normalizing the TF-IDF scores, applying additional text processing techniques or using other similarity measures. From these solutions, given that the project descriptions are not too extended, the selected approach has been to limit the length of the input text.

To do so, the main idea and goal of each key competence has been analysed and a two-word equivalent has been proposed. The shortened key competences are the following, where each number corresponds to the respective number assigned to the key competence in the Appendix B:

1. **Environmental awareness** or just **environment**.
2. **Responsible consumption of local products** or just **local products**.
3. **Healthy lifestyle**.
4. **Inequality and exclusion** or **empathy** or **inclusion**.
5. **Human rights**.
6. **Gender equality** or **discrimination**.
7. **Society opportunities** or **digital culture**.
8. **Cultural diversity** or **languages and culture**.

9. **Creativity**.
10. **Collective** or **collective engagement**.*
11. **Lifelong learning**.

* Initially, for the key competence number 10, the short version proposed was **collective project**. However, since all projects contain the word ‘project’ in their description, some non-related projects were recommended. Therefore, the final proposed short version is just the word **collective** or **collective engagement**.

After using the shortened version of the key competences as text input, the output recommended systems are more related to the competences introduced and, therefore, more accurate.

4.4 WordCloud

“Word clouds or tag clouds are graphical representations of word frequency that give greater prominence to words that appear more frequently in a source text. The larger the word in the visual the more common the word was in the document(s)” [19].

Wordclouds help analyze the text faster and easier. Therefore, when used to analyze the resulting recommended CS projects, it is less challenging to check whether the output projects are adequate and related to the input key competence or not.

A wordcloud can be easily created using the Python library wordcloud, which provides a simple and straightforward API to create word clouds from textual data.

5. CONCLUSIONS

Limitations to talk about:

- Scarcity of project scopes. Mostly related to science and environment.
- Short project descriptions. Not to explicit.
- Small database.

Recommended projects:

- Usually only top 2 are related to input text.
- With the shortened version of the KCs, there are better results.
- For those inputs that no similar projects are found, the same results are shown for all cases. Why?
- With the visual aid of WordCloud it can be checked how accurate the results are.

6. FUTURE WORK

Some possible future work that can be done:

- Adding more citizen science projects to the database. (…)
- Having a more varied scope of projects. (…)
- Obtaining more information about the description of the projects. (…)
- Recommending CS projects based on more specified competences, such as the specific competences of each of the subjects. (…)
- Expanding the list of key competences to take into account those from other school phases such as Compulsory Secondary Education and High School. (…)

7. BIBLIOGRAPHY

1. Python Software Foundation. (n.d.). Python. Retrieved May 20, 2023, from https://www.python.org/.
2. Pandas Documentation. (n.d.). Retrieved May 20, 2023, from https://pandas.pydata.org/docs/.
3. Requests Documentation. (n.d.). Retrieved May 20, 2023, from https://docs.python-requests.org/en/latest/.
4. BeautifulSoup Documentation. (n.d.). Retrieved May 20, 2023, from https://www.crummy.com/software/BeautifulSoup/bs4/doc/.
5. Project Jupyter. (n.d.). Jupyter Notebook. Retrieved May 20, 2023, from https://jupyter.org/.
6. GitHub. (n.d.). About GitHub. Retrieved May 20, 2023, from https://github.com/about.
7. Educació Primària (n.d.) XTEC. Retrieved May 25, 2023, from https://xtec.gencat.cat/ca/curriculum/primaria.
8. El Decret d'educació Bàsica (September 27, 2022) – El nou currículum. Retreieved May 25, 2023, from https://projectes.xtec.cat/nou-curriculum/educacio-basica/decret-educacio-basica/.
9. UAB - Universitat Autònoma de Barcelona (n.d.). What is citizen science? What is Citizen Science? - Universitat Autònoma de Barcelona - UAB Barcelona. Retrieved May 28, 2023, from https://www.uab.cat/web/research/responsible-research-and-innovation/citizen-science/what-is-citizen-science-1345869944451.html.
10. What is a recommendation system? (n.d.) NVIDIA Data Science Glossary. Retrieved May 30, 2023, from https://www.nvidia.com/en-us/glossary/data-science/recommendation-system/.
11. Recommendation systems: Applications and examples in 2023 (January 17, 2023) AIMultiple. Retrieved May 30, 2023, from https://research.aimultiple.com/recommendation-system/.
12. Bujokas, E. (2022) Text classification using word embeddings and deep learning in python-classifying tweets from Twitter. Retrieved May 31, 2023, from https://towardsdatascience.com/text-classification-using-word-embeddings-and-deep-learning-in-python-classifying-tweets-from-6fe644fcfc81.
13. NLTK Documentation. (n.d.). Retrieved May 31, 2023, from https://www.nltk.org/.
14. Python Documentation. (n.d.). re — Regular expression operations. Retrieved May 31, 2023, from https://docs.python.org/3/library/re.html.
15. Scikit-learn Documentation. (n.d.). TfidfVectorizer. Retrieved May 31, 2023, from https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.
16. Scikit-learn Documentation. (n.d.). cosine_similarity. Retrieved May 31, 2023, from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.html.
17. Wordcloud Documentation. (n.d.). Retrieved May 31, 2023, from https://amueller.github.io/word_cloud/.
18. matplotlib Documentation. (n.d.). Retrieved May 31, 2023, from https://matplotlib.org/.
19. Word cloud (n.d.) Better Evaluation. Retrieved May 31, 2023, from https://www.betterevaluation.org/methods-approaches/methods/word-cloud.

Text de la cita bibliogràfica [12 punts]

[Aquest apartat ha de començar en pàgina senar]

8. APPENDICES

8.1 APPENDIX A

(Location of the scripts).

8.2 APPENDIX B

List of the key competences stated by the Catalan elementary school curriculum:

1. To develop a responsible attitude based on the awareness of environmental degradation, based on the understanding of the causes that contribute to it, worsen it, or improve it, from a systemic perspective, both locally and globally.

2. To identify the different aspects related to responsible consumption and local products, assessing their repercussions on individual and common good, critically judging the needs and excesses.

3. To develop healthy lifestyle habits based on the understanding of how the body functions and the critical consideration of the internal and external factors that influence it, taking personal responsibility for promoting public health, including the knowledge of a positive, respectful, and egalitarian sexuality.

4. To exercise the sensibility to detect situations of inequality and exclusion from the comprehension of the complex causes behind them to develop feelings of empathy.

5. To develop an active commitment to gender equality, equal treatment, and non-discrimination, knowing the historical journey towards achieving human rights for all individuals and groups.

6. To understand conflicts as inherent elements of life in society that need to be resolved peacefully and rejecting any expression of misogynistic, LGBTQ+-phobic, racist violence, motivated by any type of personal or socioeconomic circumstances.

7. To analyze critically and take advantage of all types of opportunities offered by today's society, particularly those related to digital culture, assessing their benefits and risks, and making an ethical and responsible use of them that contributes to the improvement of both personal and collective life quality.

8. To accept uncertainty as an opportunity to generate more creative responses, learning to manage the anxiety it may bring.

9. To cooperate and coexist in open and evolving societies, valuing personal and cultural diversity as a source of enrichment and promoting the interest in other languages and cultures.

10. To feel part of a collective project, both locally and globally, developing empathy and generosity.

11. To develop the skills that allow lifelong learning, based on the confidence in knowledge as a driving force for development and the critical evaluation of the risks and benefits of this knowledge.

8.2 APPENDIX C