Profile-Based Retrieval

Universidad Politécnica de Madrid

Federico Paschetta, Cecilia Peccolo, Nicola Maria D'Angelo

federico.paschetta@alumnos.upm.es
cecilia.peccolo@alumnos.upm.es
nicolamaria.dangelo@alumnos.upm.es

Contents

Chapter 1

Introduction

In today's world, we're all swimming (or sometimes drowning) in a sea of information. Users navigating the vast expanses of the internet are often bombarded with more information than they can process, much of which may not even align with their interests or needs. This scenario underscores the critical importance of personalized information retrieval (IR) systems—tools designed not only to navigate the depths of available data but to surface with content precisely tailored to individual users' interests. It is within this context that our project, the development of a Profile-Based Information Retrieval Engine, finds its purpose and potential. The conceptual foundation of our project is straightforward: to create an IR engine that dynamically delivers content to users based on a nuanced understanding of their unique interests. This endeavor is inspired by the real-world observation that users' interests are often multi-faceted and may intersect across seemingly disparate domains—ranging from politics and sports to entertainment and technology. By recognizing and catering to these diverse interests, our IR engine aims to not only enhance user engagement through personalized content delivery but also to streamline the information consumption process, making it more efficient and enjoyable. At the heart of our project is a commitment to precision and adaptability. We envision a system capable of discerning the subtle nuances of user profiles, distinguishing between a casual interest in a topic and a profound passion. This requires a deep dive into the realms of natural language processing (NLP) and machine learning, fields that offer the tools and methodologies necessary to analyze,understand, and categorize vast datasets. Our engine is designed to be a bridge between the endless sea of digital content and the specific, individualized interests of users, delivering not just information, but relevance. Furthermore, this project acknowledges the dynamic nature of interests—they evolve, intersect, and sometimes wane. As such, our IR engine is built to be not just responsive but anticipatory, capable of adapting to changes in user profiles and the ever-shifting landscape of digital content. This adaptability is crucial for maintaining the relevance and effectiveness of the IR system over time, ensuring that it remains a valuable tool for users navigating the complexities of the information age.

Through the pages of this report, we will explore the methodologies underlying our approach, and evaluate the performance of our engine. It is our hope that this project will not only fulfill its immediate objectives but also contribute to the broader discourse on personalized information retrieval and its potential to enrich the digital ecosystem.

Chapter 2

Resources

The project's initial phase revolved around conducting a thorough search for datasets. Given our focus on a profile-based information retrieval engine, our primary task was sourcing documents suitable for our project. While the web is brimming with data, uncovering high-quality collections covering a diverse array of topics proved challenging. Consequently, we opted for newspaper data, which typically spans a wide range of subjects, catering to various end users.

Although we encountered numerous APIs offering news data, the datasets we obtained were often deficient, containing solely news headlines or restricted to a single topic. Eventually, we stumbled upon the BBC-text dataset on Kaggle, which aligned perfectly with our requirements. This dataset comprises over 2000 news articles, each categorized into one of five genres: Sport, Business, Politics, Technology, and Entertainment.

As mentioned, the dataset consists of 2126 entries, with each representing a news article and containing two attributes: *text* containing the full article text, and *category* indicating the article's genre.

While we utilized this dataset for our project, our engine is versatile and can function with alternative datasets. The genre categories served as our foundation, informing our understanding of user interests and guiding our engine's development. We believe these categories encompass the most common topics users typically engage with. However, if necessary, they can be modified without altering our methodologies.

In addition to the dataset, our resources prior to scripting the engine included text processing and performance evaluation codes available on Moodle. We judiciously integrated these resources into our development process.

In the development of our engine, we leveraged several essential libraries that are integral for the execution of the script:

- pandas: This library is indispensable for efficient handling of data structures, particularly data frames. With pandas, tasks such as data manipulation, cleaning, and analysis become streamlined and intuitive. It provides powerful tools for data aggregation, transformation, and visualization, making it a cornerstone for data-centric applications.

- numpy: When it comes to advanced numerical computations, numpy is the go-to library. It offers comprehensive support for mathematical operations on arrays and matrices, enabling

efficient computation of complex algorithms. Numpy's array-oriented computing capabilities enhance performance and productivity, especially in scientific and engineering applications.

- matplotlib: Visualization is crucial for understanding data and conveying insights effectively. Matplotlib is a versatile plotting library that facilitates the creation of various types of graphs and charts. It excels in generating high-quality visualizations, including evaluation graphs, which are essential for assessing the performance of models and algorithms.

- nltk (Natural Language Toolkit): NLP (Natural Language Processing) tasks demand specialized tools, and NLTK is a comprehensive library tailored for this purpose. It provides a wide range of functionalities for working with text data, including tokenization, part-of-speech tagging, syntactic analysis, sentiment analysis, and more. NLTK's extensive collection of corpora and resources further enriches its capabilities, making it indispensable for NLP tasks.

- scikit-learn (sklearn): Machine learning tasks often involve the creation and manipulation of complex vector data structures. Scikit-learn is a powerful machine learning library that offers a rich set of tools for building and deploying machine learning models. From classification and regression to clustering and dimensionality reduction, sklearn provides efficient implementations of various algorithms along with utilities for data preprocessing and model evaluation.

- random: Randomness is frequently required in computational tasks, and Python's built-in random module provides convenient functions for generating random numbers. Whether it's for sampling, shuffling, or seeding random processes, the random module ensures reproducibility and randomness in simulations and experiments.

By integrating these libraries into our engine, we've equipped it with the necessary tools and capabilities to tackle a wide range of computational and analytical challenges, ensuring robustness, efficiency, and versatility in its functionality.

Chapter 3

Text Processing

In this chapter, we delve into the intricacies of text processing and feature engineering, which are vital steps in preparing textual data for analysis and machine learning tasks. Our text processing steps are quite the same provided in *text_processing_workflow.ipynb* file, available on Moodle, we will illustrate each stage of the process.

3.1 Installing Required Packages

Before we begin, it's crucial to ensure that all necessary packages are installed. We achieve this using Python's package manager, pip, to install the required libraries.

3.2 Importing Essential Libraries

Our journey starts with importing essential libraries for text processing and data manipulation. These libraries include pandas, numpy, nltk (Natural Language Toolkit), and matplotlib.

3.3 Dataset Acquisition

Next, we acquire our dataset from a CSV file. For demonstration purposes, we utilize the "bbc-text.csv" dataset, containing news articles categorized into various topics.

3.4 Text Cleaning and Processing Steps

Text preprocessing plays a critical role in standardizing and improving the quality of textual data. We outline several preprocessing steps:

3.4.1 Removing Punctuation

Punctuation marks typically do not contribute to the meaning of text and are often removed during preprocessing.

3.4.2 Tokenization

Tokenization involves splitting the text into individual words or tokens. We accomplish this by tokenizing the cleaned text using NLTK's word_tokenize function.

3.4.3 Removing Small Words

Words with a length less than or equal to three are often considered less informative and can be removed.

3.4.4 Removing Stopwords

Stopwords are common words that do not carry significant meaning in text analysis. We eliminate stopwords using NLTK's corpus stopwords list.

3.4.5 Lemmatization

Lemmatization converts words to their base or dictionary form, considering language grammar and vocabulary.

3.4.6 Part-Of-Speech Tagging

We annotate each word with its Part-Of-Speech (POS) tag, providing insights into grammatical roles.

Text processing and feature engineering are indispensable steps in extracting valuable insights from textual data. By employing techniques such as cleaning, tokenization, stopwords removal, lemmatization, and POS tagging, we transform raw text into structured and analyzable form, laying the groundwork for subsequent analysis and modeling.

Chapter 4

Methodologies

The fundamental idea behind the project is to create new labels for texts and subsequently identify the most relevant documents for the topics that each user engaged with the most. To achieve this objective, two methodologies were employed. The first methodology involves calculating the probability of each document belonging to a topic using topic modeling, specifically Latent Dirichlet Allocation (LDA). The second methodology entails computing the similarity between TF-IDF scores of documents and topics.

4.1 Methodology 1: Latent Dirichlet Allocation (LDA) for Topic Modeling

Latent Dirichlet Allocation (LDA) is a probabilistic generative model used for topic modeling in text corpora. It operates under the assumption that each document is a mixture of multiple topics, and each topic is characterized by a distribution of words. LDA is particularly useful for our project for the following reasons:

- Topic Extraction: LDA helps in extracting latent topics from text corpora without requiring prior knowledge of the topics.
- Dimensionality Reduction: By representing documents as distributions over topics, LDA facilitates the reduction of dimensionality in text data.
- Document Similarity: LDA enables the comparison of documents based on their topic distributions, allowing for the identification of relevant documents for specific topics.

4.1.1 Goal

The objective we aim to accomplish through this methodology is to provide a selection of documents tailored to the interests of each user. This endeavor holds significant potential across various applications, such as recommendation algorithms, more and more important nowadays.

4.2 Methodology 2: TF-IDF Similarity Calculation

TF-IDF (Term Frequency-Inverse Document Frequency) is a statistical measure used to evaluate the importance of a word in a document relative to a corpus. In our project, TF-IDF scores are calculated for both documents and topics. The similarity between documents and topics is then determined based on the cosine similarity between their TF-IDF vectors. This methodology is advantageous due to the following reasons:

- Semantic Understanding: TF-IDF captures word importance by considering their relevance not just within documents but across the entire dataset. This approach enables more accurate matching between documents and user preferences.
- Dimensionality Reduction: TF-IDF converts textual data into numerical vectors, reducing dimensionality and improving computational efficiency while preventing overfitting, resulting in more generalizable models.
- Scalability: TF-IDF similarity calculations are scalable, handling large datasets efficiently to maintain the responsiveness and efficiency of our recommendation system as data volume increases.
- Interpretability: TF-IDF methodology enhances recommendation process transparency and stakeholder trust by quantifying word and topic relevance, offering actionable insights into decision-making.

4.2.1 Goal

The goal we want to achieve throughout this methodology is twofold, in the first part we want to get a result similar to the goal previous mentioned, to provide the user a list of documents, selected as the most interesting to him. Then we will try to flip the engine, identifying users who may be interested in each document.

Chapter 5

Implementation

5.1 Application of the first method

In this section, we delve into the practical implementation of the first methodology, which revolves around the application of Latent Dirichlet Allocation (LDA) for topic modeling. The implementation process encompasses several crucial steps aimed at efficiently identifying the most relevant words for each document and subsequently categorizing documents into distinct topics. Let's explore the details of this implementation journey.

5.1.1 Preprocessing and Filtering:

Our initial step in leveraging LDA involved identifying the most probable words for each document. To enhance the efficiency of this process, we embarked on calculating the TF-IDF scores for all words present in the corpus. Following this, we employed a filtering mechanism whereby words with scores lower than the total average of all scores were removed. This strategic elimination of words with lower scores served to eradicate irrelevant terms from the modeling process, thus refining the dataset. Additionally, the word "said," owing to its frequent occurrence across multiple topics and the potential introduction of bias, was also removed from consideration. With the text duly preprocessed and filtered to retain only the most relevant words, we proceeded to apply LDA.

5.1.2 LDA Modeling:

The crux of our methodology lay in the application of LDA, aimed at identifying distinct topics within the dataset. Setting the number of topics to five, our objective was to differentiate between the predefined categories present in the dataset, namely *entertainment, tech, politics, business, and sport*. By running the LDA model, we sought to discern the most relevant words associated with each of the five categories. The resulting output provided insights into the thematic structure of the documents and the effectiveness of the topic modeling process.

Results of LDA Modeling: The output of our LDA modeling endeavor revealed five distinct topics, each characterized by a set of prominent words. Notably, these topics closely aligned with the predefined categories in the dataset, indicating a successful differentiation. The extracted topics

offered valuable insights into the underlying thematic composition of the documents, laying the groundwork for subsequent analyses. The results are as follows:

- **Topic 1:** *film, game, best, year, award, music, star, number, time, song, actor, band, director, album, movie, play, million, world, oscar, like, chart, prize, sale, british, ...*
- **Topic 2:** *game, player, england, year, time, world, club, play, match, team, second, wale, ireland, good, final, champion, injury, coach, season, minute, france, week, think, rugby, chelsea, victory, ...*
- **Topic 3:** *year, government, company, market, firm, sale, country, price, economy, share, bank, growth, economic, month, 2004, rate, business, deal, world, china, analyst, rise, plan, group, cost, chief, state, expected, profit, ...*
- **Topic 4:** *people, labour, party, service, election, mobile, technology, blair, phone, government, minister, make, tory, year, time, like, brown, told, say, home, music, work, want, digital, plan, network, ...*
- **Topic 5:** *company, security, software, court, firm, virus, case, email, year, program, microsoft, people, trial, user, drug, ukip, told, charge, computer, legal, criminal, kilroysilk, window, police, lawyer, right, ...*

Considering these results we labeled Topic 1 as **"entertainment"**, Topic 2 as **"sport"**, Topic 3 as **"business"**, Topic 4 as **"politics"**, Topic 5 as **"tech"** .

5.1.3 User Definition and Document Retrieval:

Moving forward in our implementation, we proceeded to define random users and assign potential interests to each user. These interests were represented as combinations of topics, reflecting the specific areas of interest for each user. The users have been defined as follows:

- User 1 interest: entertainment
- User 2 interests: tech and business
- User 3 interest: politics
- User 4 interests: business and politics
- User 5 interest: sport

Following user definition, we embarked on retrieving the top 50 documents associated with the highest probabilities for each topic, as calculated by the LDA model, tailored to each user's specified interests.

Example of Retrieved Documents: The retrieved documents, curated based on the specified interests of each user, were meticulously examined. By perusing through the top 50 documents for each user, we gained valuable insights into the relevance and alignment of the retrieved documents with the user's stated interests, as determined by the LDA model. This is a example of output for the first two users:

- **User 1: entertainment**
  - **Document 1 - ID: 1004**
    critic back aviator oscar martin scorsese aviator best film oscar according leading movie critic several
  - **Document 2 - ID: 1263**
    producer scoop stage award producer beaten mary poppins battle blockbuster west musical olivier award
  - **Document 3 - ID: 901**
    eastwood baby scoop oscar clint eastwood million dollar baby beat martin scorsese aviator award oscar
  - **Document 4 - ID: 1499**
    aviator win globe accolade aviator named best film golden globe award star leonardo dicaprio named best actor
- **User 2: tech business**
  - **Document 1 - ID: 1800**
    microsoft make antipiracy move microsoft say clamping people running pirated version window operating system
  - **Document 2 - ID: 1191**
    jones file lawsuit conte marion jones filed lawsuit defamation balco bos victor conte
  - **Document 3 - ID: 808**
    microsoft set sight spyware window user could soon paying microsoft keep free spyware
  - **Document 4 - ID: 1412**
    worldcom director admits lying former chief financial officer telecom firm worldcom admitted

In conclusion, the implementation of the first methodology, centered around the application of LDA for topic modeling, proved to be a robust approach for categorizing documents and retrieving relevant content based on user interests. Through a systematic combination of preprocessing techniques, LDA modeling, and user-defined interest parameters, we successfully extracted meaningful insights from the dataset, paving the way for further analysis and exploration.

5.2 Application of the second method

In this section we will llook at the implementation of the second methodology, which uses TF-IDF and cosine similarity to check for similarities between document and a set of words. We will analyze in detail each step of the process, starting from the dataset already pre-processed as seen in Chapter 2.

5.2.1 Topic describing words selection

The Word Selection process serves as a crucial step in building the foundation for the recommendation system. By identifying key words associated with given topics, the system can better understand user interests and match them with relevant documents. In order to be the most unbiased possible, we asked ChatGPT to provide us a list of the best 50 words to describe each of the topics our documents belong to. We then stored them in a different file, *words.txt*, read them at runtime and stored them in a dictionary, which will be useful later.

Results of Words Selection

- **Tech:** *technology, innovation, gadgets, startups, ai, cybersecurity, blockchain, data, internet, software, hardware, apps, digital, automation, vr, ar, iot, cloud, robotics, machinelearning, cryptocurrency, e-commerce, privacy, siliconvalley, bigdata, ...*
- **Business:** *economy, market, finance, investment, stocks, companies, trade, growth, profit, loss, shares, entrepreneurship, globalization, competition, innovation, startup, revenue, sector, entrepreneur, capital, venture, bank, corporation, mergers, ...*
- **Politics:** *government, elections, policy, democracy, leadership, president, congress, legislation, politics, politician, campaign, voting, party, opposition, power, authority, governance, diplomacy, international, national, state, federal, capitol, political, ...*
- **Entertainment:** *film, music, television, hollywood, celebrities, actor, actress, director, star, entertainment, showbiz, celebrity, performance, premiere, boxoffice, streaming, blockbuster, drama, comedy, action, romance, thriller, ...*
- **Sport:** *sport, athletics, football, soccer, basketball, baseball, tennis, golf, olympics, athlete, team, league, championship, victory, defeat, score, goal, match, game, tournament, season, training, coaching, ...*

5.2.2 TF-IDF Matrix

To convert textual data into numerical vectors, tokenization was performed to extract meaningful text units, or tokens. These tokens were used as column labels in a TF-IDF matrix, capturing the importance of each token across the document corpus. Additionally, TF-IDF vectors representing each document were generated, Setting the stage for similarity computations and the development of recommendation systems.

5.2.3 Similarity DataFrame

A DataFrame is initialized to capture similarity values between users and documents. Document-related data is extracted and assigned to respective columns. Using cosine similarity, similarity values between user and document vectors are calculated and populated into the DataFrame.

5.2.4 User Definition and Document Retrieval

We used users already created in first application, in order to be able to compare the two applications. With the DataFrame we built, we were able to get cosine similarity between document vector and user vectors. In this way we computed the similarity between each document and each user. These were the results obtained.

Best Document for User

Here we got the most appropriate documents for each user interests among the full corpus.

- **User 1: entertainment**
  - **Document 1 - ID: 664**
    world premiere stage musical coproduced canadian theatrical impresario david mirvish take place toronto princess wale theatre
  - **Document 2 - ID: 273**
    critic laud comedy sideways road trip comedy sideways praise heaped critic association adding honour already picked chicago film critic association
  - **Document 3 - ID: 364**
    bangkok film festival battle organiser third bangkok international film festival determined carry year event despite ravage asian tsunami disaster festivity scaled carpet
  - **Document 4 - ID: 1073**
    sundance honour foreign film international film given prominence film next year sundance film festival movie dominated theme independent film festival feature international cinema
- **User 2: tech business**
  - **Document 1 - ID: 1373**
    economy facing major risk manufacturing sector continue face serious challenge next year british chamber commerce said group quarterly survey company
  - **Document 2 - ID: 1254**
    israeli economy picking pace israel economy forecast grow 2004 continues emerge three-year recession main driver fasterthanexpected expansion export tourism
  - **Document 3 - ID: 1649**
    trade ballooned october trade deficit widened expected october hitting record level higher price raised import cost figure shown trade shortfall 555bn
  - **Document 4 - ID: 514**
    regulation still possible blurring boundary internet raise question regulation watchdog ofcom said content internet move closer year

Best User for Document

Here, for each document in the corpus we got the most appropriate user (or, if above a similarity threshold, fine-tuned at 0.025, users). Results for random documents are listed below.

- **Document 1**
  ID: 97
  Text: ireland eclipse refereeing error international rugby board step stop frustrated coach player publicly haranguing referee thing bellyup whole nflstyle video camera field
  Category: Sport
  User Selected: 5 - sport

- **Document 2**
  ID: 674
  Text: pirate profit motive part huge network internet software pirate known drink convicted bailey news investigates network worked motivated
  Category: Tech
  User Selected: 2 - tech business

- **Document 3**
  ID: 1272
  Text: vera drake bafta triumph hope bafta film award saturday night prospect homegrown movie could walk clutch trophy vera drake mike leigh tale 1950s
  Category: Entertainment
  User Selected: 2 - tech business

- **Document 3**
  ID: 51
  Text: strike threat pension plan million public service worker could strike minister scrap final salary pension scheme make work longer warn union leader cabinet office
  Category: Politics
  User Selected: 4 - business politics, 3 - politics

In summary, the implementation involved utilizing ChatGPT to generate topic-specific word lists, which were then used to construct TF-IDF vectors for both users and documents. These vectors were employed to compute similarities, facilitating the identification of the most relevant documents for each user and vice versa.

5.3 Evaluation

The evaluation aimed to assess whether the documents retrieved for each user were genuinely relevant to their interests. We sought to determine if our method of assigning topics to documents effectively classified them for return to users. To achieve this, precision and recall metrics were employed, utilizing the "evaltool" function provided. To obtain meaningful evaluation results, a query vector was created, treating it as a vector of 1s, and a response vector was generated for comparison. The response vector was designated as 1 where the document returned to the user aligned with their interest and 0 where misclassification occurred, resulting in the return of an irrelevant document to the user. We obtained the following results with the dataset we built the engine with.

- **First method**
  Precision: 0.89
  Recall: 1

- **Second method**
  Precision: 0.67
  Recall: 1

Expanding the evaluation in the second method, we aimed to assess its effectiveness in identifying suitable users for document delivery. We employed straightforward metrics, namely accuracy, but implemented two variations: *strict* and *loose*. These metrics provided insights into the methodology's ability to match users with documents based on their preferences, offering a comprehensive understanding of its performance. Below the meaning of the two evaluations functions.

- **Loose**

$$\frac{n^\circ \textit{ documents with most interested user predicted correctly}}{\textit{total } n^\circ \textit{ documents}}$$

  With this "loose accuracy" we compute the number of documents where the first element in the interested users list has among his interests the topic the document belong to and we divide it by the total number of documents. With the dataset we used during all our project we got 85.67% of loose accuracy.

- **Strict**

$$\frac{n^\circ \textit{ documents with all users predicted correctly}}{\textit{total } n^\circ \textit{ documents}}$$

  To get the percentage of the "strict accuracy" we compute the number of documents where all elements in interested users list has among his interests the topic the document belong

to and we divide it by the total number of documents. With the dataset we used during all our project we got 78.02% of loose accuracy. In order to maximize this score we needed to fine-tune the threshold parameter, setting it to 0.025: this parameter must be tuned based on necessities and specific cases.

Then, to make sure that our results weren't caused only by the goodness of the corpus, we repeated the process with another dataset with the same structure, BBC News Train Data and we obtained similar results.

- **First method**
  Precision: 0.89
  Recall: 1

- **Second method**
  Precision: 0.62
  Recall: 1
  Loose Accuracy: 85.63%
  Strict Accuracy: 76.71%

Chapter 6

Conclusion

6.1 Comparison results with group 1 and group 4

During our research, we carefully assessed the results obtained from our analyses. In order to broaden our perspective and ensure the reliability of our methods, we compared our evaluations with those of two additional independent groups. This comparison enabled us to gain a deeper understanding and to compare our findings with those of other researchers actively involved in the field.

- **Group 1**
  MAP: 0.62

  The approach utilized by this group was also intriguing, as they too employed a markedly different strategy from ours: they obtained the most relevant documents for each topic by optimizing the loss function, thereby ensuring retrieval of only the top 10 documents per topic that a user might be interested in. The results yielded are quite satisfactory for some queries, yet markedly inconsistent, with peak precision achieved at the second query and a significant decline observed in the fourth and fifth queries.

  Their method demonstrates a keen focus on precision and relevance, aiming to deliver tailored results aligned with user interests. By optimizing the loss function, they prioritize the retrieval of documents that are most likely to meet user expectations, enhancing the overall user experience.

  While their approach may yield promising results for certain queries, the observed variability underscores the complexity inherent in document retrieval tasks. Further refinement and exploration may be necessary to address the disparities and optimize performance across various query scenarios.

  Overall, their innovative approach offers valuable insights into the nuances of document retrieval and highlights the diverse strategies that can be employed to enhance search effectiveness and user satisfaction.

- **Group 4**
  Precision: 0.47

  The method our colleagues used diverges from the our automatic approaches. Instead, they adopt a different perspective, one that emphasizes the vector space of documents over the semantic analysis of topics and words. Their vectors are notably lengthy, perhaps contributing to the low values observed in calculating the similarity between a document and the vector of relevant topics.

  Rather than delving deeply into semantic nuances, they prioritize the spatial arrangement of documents within a high-dimensional vector space. This approach allows them to discern patterns and relationships based on the geometric proximity of documents, offering a unique lens through which to understand and classify textual data.

  By focusing on the structural representation of documents within the vector space, they aim to capture inherent similarities and differences that may not be immediately apparent through traditional semantic analysis alone. While their method may diverge from the norm, it offers a promising avenue for document classification that warrants further exploration and refinement.

6.2 Conclusion and Consideration

In conclusion, we are very satisfied with the results achieved in this research. Our analysis, has highlighted remarkable performances. In particular, the discussion around recall has raised interesting issues; despite recognizing its importance, we concluded that it should not be the sole criterion for evaluating results. On the contrary, we found that other measures, presented in the evaluations, offer a more balanced and relevant perspective on the effectiveness of our approach. These reflections led us to consider more carefully the metrics that we believe are most significant for our study, suggesting a clear direction for future research. We have demonstrated that, thanks to the robustness of the current approach, it is possible to extend the analysis to new documents belonging to the same collection or type without the need to repeat the entire process. This will be achieved by calculating the posterior probability of the topics for future documents, a step that promises to further optimize our approach, making it even more efficient and scalable. Looking forward, we propose to focus on further development of this methodology, exploring its potential applications in different contexts and refining our ability to predict user interest in specific documents. This will not only strengthen our understanding of the dynamics of the analyzed documents but also provide a personalized service to users, based on an accurate prediction of their interests.
Lastly, as we continue to improve our understanding and techniques, we remain committed to exploring new frontiers in research, always with the goal of enhancing the accuracy and efficiency of recommendation and classification systems.