# XAI User Classifications

Oded Koren, Omer Zilberman, Nir Perel, Michal Koren

## Abstract

Explainable artificial intelligence (XAI) models are artificial intelligence (AI) systems whose learned algorithms and outputs can be trusted and understood by users. In this paper, we suggest an XAI method in the context of topic modeling that allows users to understand how they are classified by an AI system. Our proposed method yields high-accuracy results. This provides a better understanding and awareness for users of how the system classifies their preferences, and therefore users can control and adjust the recommendations they receive from the AI system.

Keywords: XAI; Topic modeling; Classification.

## Introduction

Audience and user targeting has become one of the leading subjects, both for research and marketing purposes.

With the enormous (and increasing) amounts of personal data being collected, an opportunity arises for marketing managers: personalizing advertisements (henceforth ads) and content for each individual user, thus increasing the efficiency of and revenue from these items [1]. In order to personalize ads, there is a need for user-targeting algorithms, which use artificial intelligence (AI), particularly machine learning (ML) [2], in order to classify or cluster subjects.

AI can be defined as a set of capabilities given to a machine to mimic human intelligence in its responses or actions [3]. The harder it is to distinguish between real human actions and the machine’s actions, the better the AI [4].

The learning process in AI is called machine learning (ML), which is generally divided into three main methods: supervised learning, unsupervised learning, and reinforcement learning [5]. Supervised learning requires a labeled input (i.e., input and its corresponding output) that the machine uses for building an optimal mathematical model. In contrast, unsupervised learning does not require any intervention beyond providing the input data (no labels are provided); the machine clusters the data into groups based on a chosen algorithm. Reinforcement learning means that for each prediction or outcome the algorithm generates, it will be “rewarded” or “punished,” depending on the desirable outcome, thus improving its predictions [3].

Nowadays, Big Data offers new opportunities for those who seek to perfect their user-targeted content [6]. Big Data refers to the exponential and ongoing growth of data being created [7]. Big Data differs from regular (or “small”) data by being unstructured and/or too huge to be handled by regular database software [8]. Extracting useful insights or conclusions from the data is referred to as “data mining” [3].

A well-known group of algorithms for dealing with data are classification methods. Classification is a supervised learning problem that requires a classifier, which is basically the protocol by which the machine can classify. A classifier is created by “training” the ML algorithm with labeled data. Then, when a new unlabeled input dataset is presented, the classification algorithm is expected to classify it accurately [9, 10].

Clustering is another data mining technique and is an unsupervised learning method. Clustering means grouping unlabeled data into clusters, where each cluster has its own common ground. There are several types of clustering, such as hierarchical clustering, which generates the next cluster based on previously generated clusters, and partitional clustering, which generates all clusters simultaneously [11].

The capabilities of AI in processing data and clustering it are some of the reasons why it has become a very useful tool for marketing managers to target their users, aiming to increase the efficiency of ads. As a result, various targeting methods have developed, and user-targeted advertisement has become a fast-growing area in the IT industry [12]. There are many platforms on which a “user-targeted feature” can be presented – for example, while playing a mobile game, using a search engine, watching a video, or buying from an e-commerce store. These platforms use AI to target their users efficiently, as advertisers often do not know enough about their customers or cannot extract the insights that AI can regarding them [13].

Despite the benefits from using such a technology, some may be concerned about the privacy of the user (“the target”). ML requires large quantities of data in order to be accurate, and the leakage or misuse of that data can cause major privacy issues for the user [14]. Stolen personal data can be used in many harmful ways – identity theft, account takeover and extortion, to name a few. However, privacy is out of scope of this work.

Besides the obvious privacy issues, another concern is about the transparency of ML’s decision-making process. The results obtained from ML algorithms can be biased, as happened, for example, with Amazon’s[[1]](#footnote-1) AI recruiting tool, which was biased against women [15], and Apple’s[[2]](#footnote-2) credit card system offering different credit limits for men and women [16]. The lack of transparency of the process and its reasoning may challenge the legitimacy of using AI in decision making related to the public [17].

When discussing AI transparency, researchers find it hard to come to a consensus about its definition. In Ref. [18], it is suggested that a system that does not give any justification for its outputs be described as an opaque system. Some suggest that transparency can be regarded differently for different elements of the system: the whole model should be understandable without any prior knowledge, the variables and calculations should be visible and understandable, and the algorithm itself should be understandable [19]. Others might relate to model transparency as the ability of the user to understand and study the mathematical process leading to the outcome [20].

Some researchers choose to address the transparency issue not with explanations but with interpretability measures. As stated in Ref. [21], interpretable ML can be defined as the usage of ML to generate “relevant knowledge” about the data.

Certain steps can be taken in order to increase AI’s transparency, such as creating an explainable AI (XAI), which can clarify the results given for humans to understand [22, 23]; revealing the source code [23]; and using auditing services. However, one should not forget that more transparency does not necessarily mean more accurate results (i.e., less bias) [23].

Explanation is not necessarily a detailed description of the process. In the XAI world, it would typically mean a justification provided for a decision made [22].

As suggested in Ref. [24], XAI methods can encounter difficulties with measuring transparency and rating explanations as there are no agreed-upon measures and definitions. In addition, revealing the source code might encounter difficulties in achieving explanatory goals, either because most users would not have the ability to read the code, let alone to understand it, or because revealing the code may cause a negative reaction – users understand the decision-making process but are unhappy with the decision factors [17]. Moreover, producing an AI that was fully transparent to the public might undermine its purpose, as it would be easier to “trick” the AI. In addition, this approach could influence the programmers to focus more on making the AI legitimate rather than accurate [17,23].

## Related Work

Many of the works mentioned in the Introduction are server-side solutions, meaning that the transparency is achieved following the company’s actions (i.e. reveal the source code, XAI). However, in this paper, we would like to suggest a user-side method to make the process transparent and understandable for the user.

While Refs [22, 24] regard XAI as a method to deploy on a certain AI technology (ML, Deep Learning, etc.), meaning that for each model and algorithm used, a different XAI solution is needed, we feel that the solution should be technology-agnostic. Moreover, many of the works we reviewed offered a solution or an approach only for supervised learning methods.

In Ref. [25], a technique called LIME (Local Interpretable Model-agnostic Explanations) is introduced. LIME’s purpose is to provide explanation for selected model results, and it was found to be a reliable tool to do so, both with expert and non-expert subjects. It offers its explanation by displaying the features and inputs that led to the presented results. However, LIME works with supervised ML and classification models in particular, meaning that it might require data that a client does not have.

Furthermore, a method presented in Ref. [26] suggests an explanation vector as a solution for interpreting the process and the results of a classifying algorithm. Explanation vectors (or local gradients) were found to be efficient in discovering how to manipulate a data record in order to change its predicted label. Although this method’s explanations were found insightful, it is stated that it is yet to be technology-agnostic and is relevant only to some classifying algorithms, ignoring any regression and unsupervised models.

A previous work regarding targeted ads and how are they effected used reverse engineering on profiles; it achieved 79% precision in finding Google’s[[3]](#footnote-3) user interest categories and successfully retrieved 58% of Google Ads profiles [27]. However, this work does not find or rate the ‘most influential’ categories that have an impact on the presented ads.

Another work in this field tried to measure the effect of different forms of targeting. While identifying 43% of location-based and 39% of user-based ads, and showing that time and application are key factors in almost every form of targeting, the researchers stated in the work that it was not its intent to uncover whether profiling is used in targeting ads; thus extracting the ‘influential’ categories is unavailable [28].

In Ref. [28], a model is suggested for unsupervised learning, yet it is illustrated on text classification, stating that the input must be a binary number representing the existence of a “concept”.

In this paper, we propose an XAI method that is based on text mining and topic modeling. The process of text mining enables information to be deduced from unstructured text data. Topic modeling is a common text mining technique for identifying hidden structures and improving text classification based on the statistics of the words. This technique enables huge datasets that include text to be organized and summarized, for example in online social networks [29, 30]. By using topic analysis models, phrase patterns can be created, and detected words are automatically clustered [31, 32]. There are a variety of applications for topic modeling, as well as for document clustering techniques, in fields such as vehicle movements on urban road networks [33], classifications in labor market intelligence [34], and protection of infrastructure and IT systems [35].

Clustering documents and classifying them by topic forms the basis of the analysis. The word2vec model is a two-layer neural net that is useful in various Natural Language Processing (NLP) tasks. The input is a text corpus, and the output model is a set of vectors that represents the document by distributed numerical representations of word features; namely, a vocabulary in which each item has a vector attached to it. The vector created can be used to detect relationships between words [36, 37, 38]. Doc2vec is an extension of word2vec, used for associating arbitrary documents with labels and learning document-level embeddings. The advantage of this model is that it overcomes the weaknesses of bag-of-words models [39, 40].

Latent Dirichlet Analysis (LDA) is a probabilistic approach for topic modeling. The aim of LDA is to identify the topics in a given document, based on the words in the document [41]. LDA works by assigning a probability to each word, according to the corresponding topic. It represents documents as a combination of topics, while a topic is a combination of words.

## Goal

As described in the Introduction, the purpose of XAI is to explain the algorithms, analyses, and outcomes (e.g., recommendations, decisions, etc.) of the AI procedures.

The main goal of our research is to strengthen the user’s trust in the AI process and their understanding of the results obtained from it. To that end, we create a proof-of-concept research model that illustrates these two entities, i.e., the service that deals with the AI process, and the user. ~~can improve the confidence between each other.~~

The main objective of this research is to enlighten and elaborate the AI information, services, content, and recommendations that the user receives, and by that, to validate the AI assurance on the user side. Furthermore, increasing the trust of the user in the service may give rise to customer demand for additional content and services.

## Method

In order to reach the objective of this research, we define and create a proof-of-concept system that can illustrate an XAI model concept and that enables the user, via a recommendation system, to classify the content provided to them by the server.

Figure 1 illustrates a model without the addition of XAI abilities. The model is divided into the server side, in which most of the AI analysis is performed (i.e., algorithms, analysis, and decision making), and the user side, which receives the outcomes (i.e., recommendations, content, services, etc.) sent from the server side.

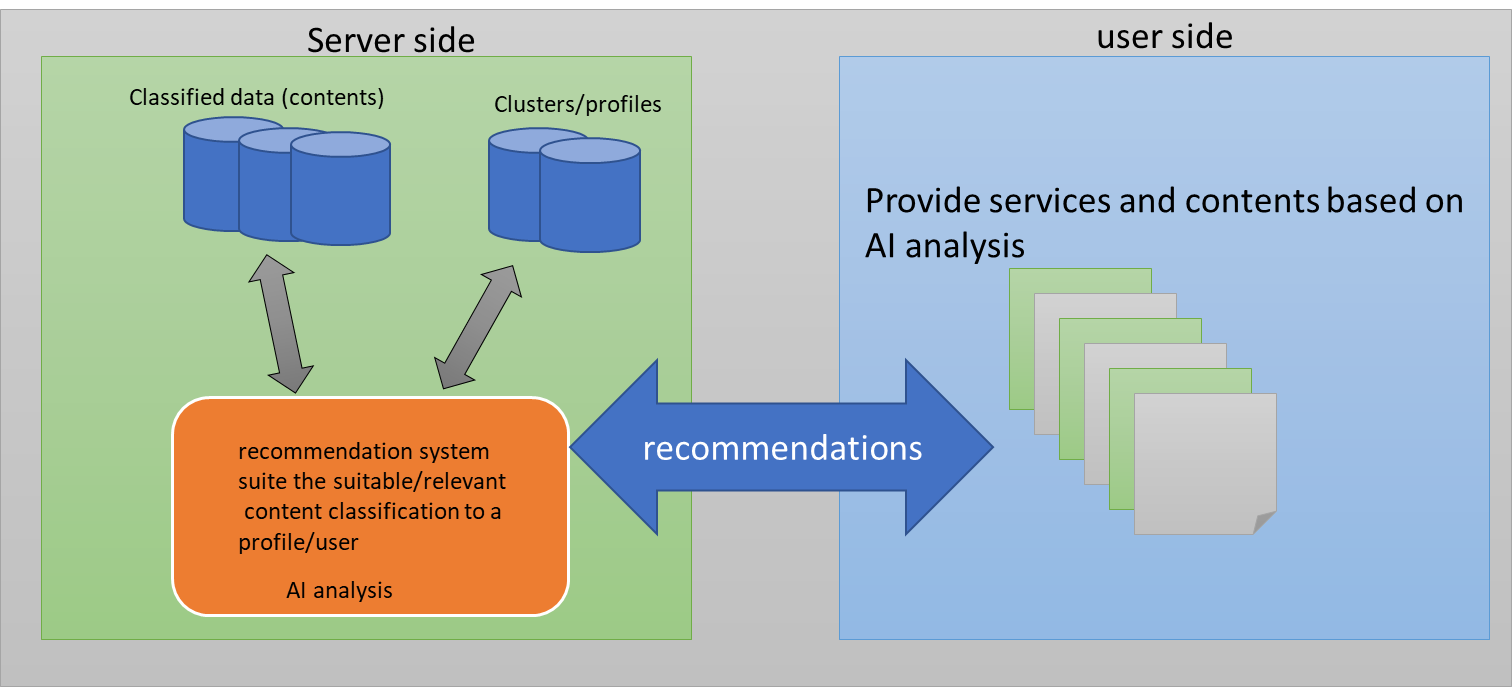
****

Figure : End to end service

It may be that in some cases, users receive content without the ability to understand to which classes this content belongs. That is, users cannot understand why they received that content.

The ability of users to understand and classify the content’s classes increases clarification and explanation to the users. In addition, the users may gain information about how they have been classified by the system, which may improve their trust in the system (one of the XAI goals). Moreover, it is possible that in future phases, users will be able to actively update their current status or classification on the server side (and as a result to expand their services).

Figure 2 illustrates the additional XAI components that improve the current solution and achieve the above goal.

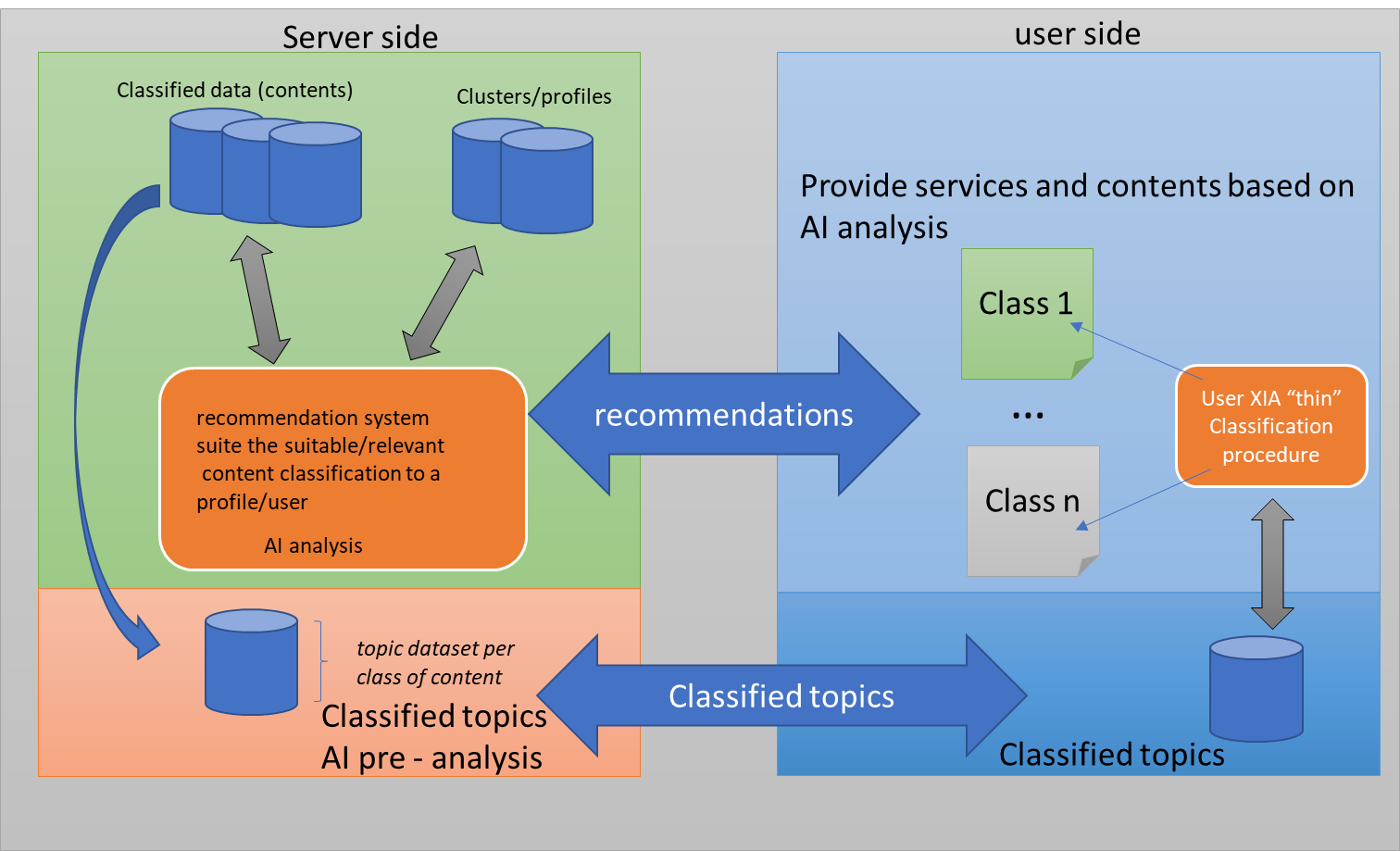


Figure : User XIA classification model

The additional procedures are again divided into the server side and the user side, as follows:

### 4.1 Server side:

Some AI procedure is performed on the classified data (in our illustration, we perform topic analysis on a set of documents, as described in the sequel). The outcome of this AI procedure is a compact dataset (possibly a dataset per class, depending on the implementation) that contains a list of topics and weights (per topic). This procedure runs offline and can be updated and modified according to the content and services, according to the user’s feedback. This compact dataset is transferred to the user side before any analysis is performed there. (The issues of when to transfer, how to transfer, how and when to update the user side, and how and when to receive updates from the user to the server are out of the scope of this research, as are security, privacy, and network issues.) ~~and should be conduct in additional researchers.~~

Note: We assume that there are trust relationships between the users and the server (service), and that the server is willing to share information (such as the classified topics) that is provided to the user.

## 4.2 User side:

The compact dataset generated by the server is now located on the user side. The server detects new and relevant content for the user and sends it to them. Then, some analysis is performed by the user (e.g., topic analysis). The user then utilizes the compact dataset they received from the server to understand how they have been classified on the server side. Specifically, in our illustration, the user gets from the server a list of topics (and their weights), which is compared (by some predefined rules) to the results of the topic analysis performed by the user. This comparison results in a classification of the content this user has received. The user also receives additional information, such as a map of content per classification, predication, true and false positive/negative predications etc. The user can be provided with more knowledge on how they are “mapped” on the server side, how much of the content and classifications they have received, and also what they have not received / have “missed” (this is very important to users that want to expand to new services/content/information and new domains of knowledge).

**Experiments**

In order to run the illustration, we used the BBC Dataset [42], which contains more than 2000 articles, divided into 5 different classes: Business, Entertainment, Politics, and Tech.

## 4.3 Server-side procedure

The procedure performed on the server side is as follows: Each article from a defined class (detailed below) from the BBC Dataset [32] was loaded from the dataset together with the title (class) and the content text.

1. Pre-process procedure:
   1. Per article, run a procedure via parsing.preprocessing [43].
      1. Lower case.
      2. Remove white space.
      3. Remove numbers.
      4. Remove insignificant values, such as punctuation, etc.
   2. Create a dictionary of word mapping via corpora.dictionary [43].
   3. Tuples of words ID and counts using doc2bow via corpora.dictionary [43].
2. Perform Latent Dirichlet Allocation via models.ldamulticore [43] on the outcome of the pre-process (previous stage). The outcome of this process is a list of topics and the weights for each class.
3. Prepare and extract the full topic dataset per class (with all five classes), including data frames of topics, weights, and calculations, such as:
   1. Averages – classified per topic on the user side.
   2. Count – to count the topics on the user side.
   3. Weight minimum – per topic what is the minimum weight?
   4. Weight maximum – per topic what is the maximum weight?

Note that the full outcome (topic data frame) of step 3 is sent to the user.

## 4.4 User-side procedure

The procedure on the user side (on the user’s site; the method runs in “real-time” – that means the process runs per each received item of content on the user side).

1. For each article that is loaded on the user side:
2. Article text is loaded (the title of the article is used only to test and compare the prediction).
3. Process steps 1 (Pre-process) and 2 (LDA) as for the server side. Note: since the experiment is conducted in the same environment (no division between the user machine and server machine), for the implementation use case, we used the same classes as used in the server and passed them to a new process. In real divided separated machines, the classes/code can be divided accordingly.
4. The outcomes are data frames of topics and weights.
5. Compare each set of rules from stage 4 outcomes to the pre-process of the data frames sent by the server. The rules can be: if the weight of each topic within the data frames from the user (per article) is bigger than the minimum or the average + count the number of topics the new article shares with the server datasets topics). Those sets of rules provide a score per title (article) – note: the rules can be defined and adjusted per user case and/or dataset (scenario) – and they illustrate the model XAI analysis and classification concept. The complexity of the rules reduces the confusion (or miss prediction) of the classed prediction.
6. The outcome is a classification report + confusion matrix (based on the score, using the [sklearn.metrics](https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics).classification\_report + [sklearn.metrics](https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics).confusion\_matrix [44]).

Based on the procedures described above, 12 different experiments (divided into two types – see below) were conducted. On the server side, the topic data frames were uploaded and analyzed (see Server-side procedure above), with the following content:

1. 50 articles per class – 5 different classes => total of 250 articles for the final topic data frame
2. 60 articles per class – 5 different classes => total of 300 articles for the final topic data frame
3. 70 articles per class – 5 different classes => total of 350 articles for the final topic data frame
4. 80 articles per class – 5 different classes => total of 400 articles for the final topic data frame
5. 90 articles per class – 5 different classes => total of 450 articles for the final topic data frame
6. 100 articles per class – 5 different classes => total of 500 articles for the final topic data frame

The six different experiments above were conducted in order to examine and analyze if the topic data frame outcomes can be updated and can improve the prediction outcome on the user side in the next two experiment types (more details can be found in Section 5: Outcomes & Analysis).

On each of the six final topic data frames (from above experiments i–vi), we conducted two user-side classification experiments:

1. One class classification – determine/predict based on the received article if the user is classified to class X (one of five classes)
2. Five full class classifications – determine/predict based on the received article if the user is classified to class 1–5 (based on the dataset).

It is important to emphasize that the articles that were analyzed for prediction on the user side (see User-side procedure) are not part of the 50–100 articles (per class) used for the topic data frames procedure on the server side. The user side will use the topic data frames for five different classes (see the Server-side procedure) in order to predict the classification of a new article on the user side (see User-side procedure).

Note that, as described above, the experiment was run in a shared environment (not divided into a user machine and a server machine) to simplify the implementation development. (The subjects of transferring the topics data frame and the classified article; any communication and network issues; and all issues of privacy, permission, and security are out of scope of the current research.)

## Outcomes & Analysis

In this section, we present the outputs and results of the 12 experiments described above.

### 5.1 Procedure outcomes:

For each test (as part of the 12 described experiments), the outcomes were:

1. Classification report

Table 1 is an example of the classification report. It presents the outcome of the 5 classes tested, based on 100 articles per class that built the topic data frame on the server side. The table presents the following measures: precision, recall, f1-score & support (number of test articles: in this case 100 articles were tested); while the columns describe the classes and accuracy, etc.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Business** | **Entertainment** | **Politics** | **Sport** | **Tech** | **accuracy** | **macro avg** | **weighted avg** |
| **precision** | 0.831579 | 0.866667 | 0.766129 | 0.909091 | 0.842105 | 0.836 | 0.843114 | 0.843114 |
| **recall** | 0.79 | 0.78 | 0.95 | 0.7 | 0.96 | 0.836 | 0.836 | 0.836 |
| **f1-score** | 0.810256 | 0.821053 | 0.848214 | 0.79096 | 0.897196 | 0.836 | 0.833536 | 0.833536 |
| **support** | 100 | 100 | 100 | 100 | 100 | 0.836 | 500 | 500 |

Table : Classification report - 5 classes test based on 100 articles per class

1. Confusion matrix

Table 2 is an example of the confusion matrix. It describes the outcomes of the 5 classes tested, based on 100 articles per class that built the topic data frame on the server side. The table contains the true positive and false negatives for each row, divided into the classes being predicted, while the columns describe the true positive & false positives.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Pred\_Business** | **Pred\_Entertainment** | **Pred\_Politics** | **Pred\_Sport** | **Pred\_Tech** |
| **True\_Business** | 0.79 | 0.03 | 0.1 | 0.01 | 0.07 |
| **True\_Entertainment** | 0.04 | 0.78 | 0.08 | 0.03 | 0.07 |
| **True\_Politics** | 0.03 | 0.01 | 0.95 | 0.01 | 0 |
| **True\_Sport** | 0.09 | 0.06 | 0.11 | 0.7 | 0.04 |
| **True\_Tech** | 0 | 0.02 | 0 | 0.02 | 0.96 |

Table : Confusion matrix - 5 classes test based on 100 articles per class

**Note**: The two tables above (classification report & confusion matrix) were calculated and performed for each of 6 tests \* 6 times (1 per class = 5 tables, as part of the one class test analysis + one table for the all classes test analysis) \* 2 tables (classification + confusion) = a total of ***72 tables*** to be analyzed.

### 5.2 Analysis

In order to reduce the complexity (examining and analyzing 72 tables of values) and simplify the analysis process, the authors concentrated on a breakdown of three main outcome values: recall, F1- score, and accuracy. The aim was to undemands the values across the different 72 outcomes (tests), to evaluate whether the model could reach the classification possibility, and to determine if these values were influenced by the number of articles that build/contract the topic data frame values. Note that the issue of finding the efficient number of articles to improve performance is a topic for further research (and is out of scope of this work).

#### 5.2.1 One class test

We examine, for each of the five classes (Business, Entertainment, Politics, Sports, and Tech), whether the user (i.e., the process performed on the user side) can predict the class of a new article, by utilizing the compact data frame that came from the server.

1. Sport class

Figure 3 describes the recall of the sport class. As presented in the figure, the recall values improve from 33% with 70 articles per class in the topic data frame test to 70% with 100 articles per class.

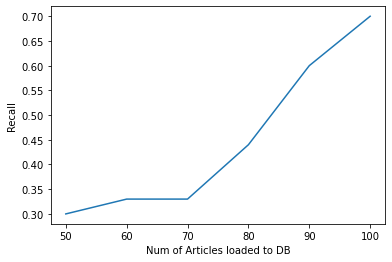


Figure : Recall - one class (sport) analysis

Figure 4 describes the F1-score of the sport class. As presented in the figure, the score values improve from 49% with 70 articles per class in the topic data frame test to 82% with 100 articles per class.

#### C:\Users\Oded.Koren\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\C39CDC0E.tmp

Figure : F1-score - one class (sport) analysis

Figure 5 describes the accuracy of the sport class. As presented in the figure, the score values improve from 33% with 70 articles per class in the topic data frame test to 70% with 100 articles per class.

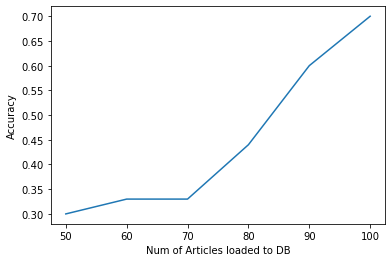


Figure 5: Accuracy - one class (sport) analysis

1. Entertainment class

Figure 5 describes the recall of the entertainment class. As presented in the figure, the recall values are inconsistent and seem to run between 63% (in the 90 articles test) and 78% (in the 100 articles test).

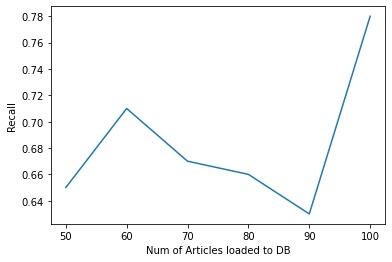


Figure : Recall - one class (Entertainment) analysis

Figure 6 describes the F1-scopre of the entertainment class. As presented in the figure, the F1-scope values are inconsistent and seem to run between 77% (in the 90 articles test) and 87% (in the 100 articles test).

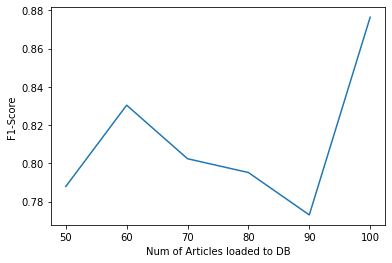


Figure : F1 Score -one class (Entertainment) analysis

Figure 7 describes the accuracy of the entertainment class. As presented in the figure, the accuracy values are inconsistent and seem to run between 63% (in the 90 articles test) and 87% (in the 100 articles test).

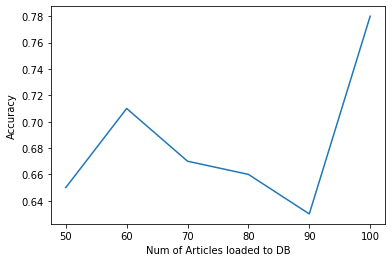


Figure : Accuracy -one class (Entertainment) analysis

1. Business class

Figure 8 describes the recall of the business class. As presented in the figure, the recall values are not fully consistent: from 74% with 60 articles per class in the topic data frame test to 82% with 90 articles per class.

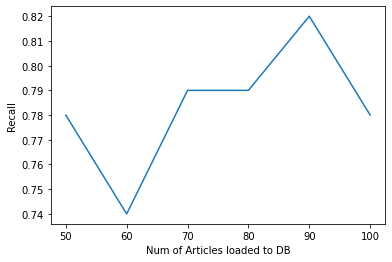


Figure : Recall - one class (Business) analysis

Figure 9 describes the F1-score of the business class. As presented in the figure, the recall values are not fully consistent: from 85% with 60 articles per class in the topic data frame test to 90% with 90 articles per class.

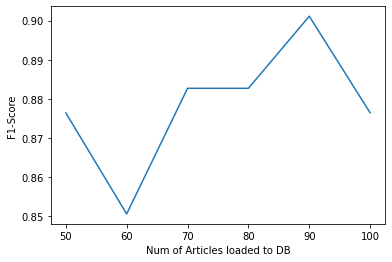


Figure : F1-score one class (Business) analysis

Figure 10 describes the accuracy of the business class. As presented in the figure, the recall values are not fully consistent: from 74% with 60 articles per class in the topic data frame test to 82% with 90 articles per class.

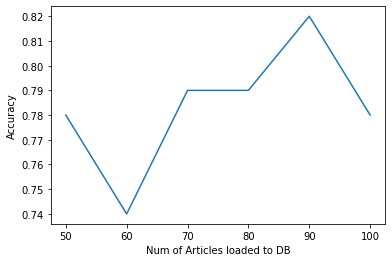


Figure : Accuracy one class (Business) analysis

1. Tech class

Figure 11 describes the recall of the tech class. As presented in the figure, the recall values are not consistent, rising up to the 90 test (then reducing in the 100 test), from 86% with 50 articles per class in the topic data frame test to 95% with 90 articles per class.

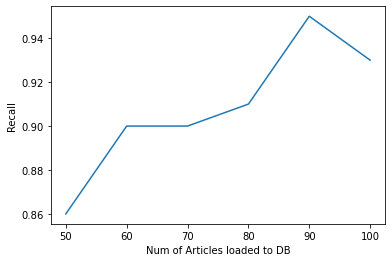


Figure : Recall - one class (Tech) analysis

Figure 12 describes the F1-score of the tech class. As presented in the figure, the F1-score values are not consistent (the same as described in the previous figure), from 92% with 50 articles per class in the topic data frame test to 97% with 90 articles per class.

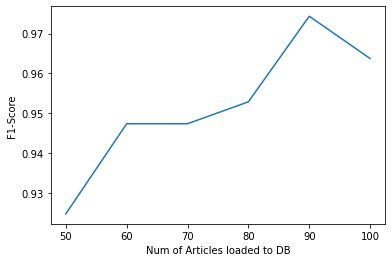


Figure : F1-score one class (Tech) analysis

Figure 13 describes the accuracy of the tech class. As presented in the previous tech figures, the values are not consistent: from 86% with 50 articles per class in the topic data frame test to 95% with 90 articles per class.

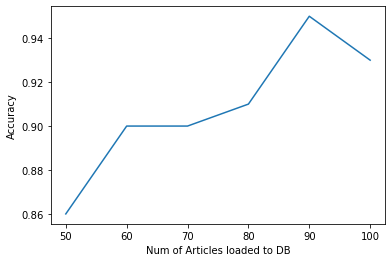


Figure : Accuracy one class (Tech) analysis

1. Politics class

Figure 14 describes the recall of the Politics class. As presented in the figure, the recall values seem more consistent (compared to other classes such as tech or business) and rise up to the 90 test (then reduce slightly in the 100 test), from 87% with 60 articles per class in the topic data frame test to 94% with 90 articles per class.

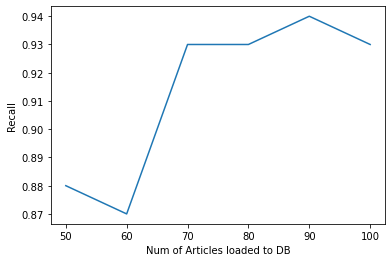
****

Figure : Recall - one class (Politics) analysis

Figure 15 describes the F1-Score of the Politics class. As presented in the figure, the F1-score values seem consistent with the previous graph, from 87% with 60 articles per class in the topic data frame test to 94% with 90 articles per class.

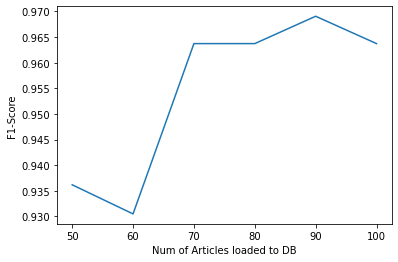
****

Figure : F1-score one class (Politics) analysis

Figure 16 describes the accuracy of the Politics class. As presented in the figure, the accuracy values seem consistent with the previous graphs, from 87% with 60 articles per class in the topic data frame test to 94% with 90 articles per class.

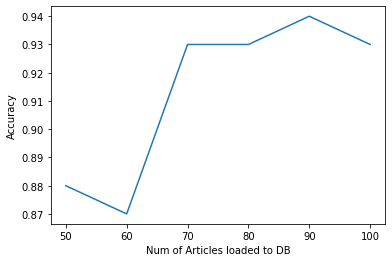
****

Figure : Accuracy one class (Politics) analysis

#### 5.2.2 Full classes test

In this test, we examine whether the user can predict (based on the topic data frames procedure – see User-side procedure) if a new article[s] should be classified to class 1–5 as desired, or if not, we also identify the true/false positive/negative prediction values. In this use case, we used all five classes for predictions – based on the topic data frame. It is also important to emphasize that we examine 100 different articles (which were not included in the topic data frame procedure – see Server-side procedure) in order to test our model, and we present the accuracy of the prediction below.

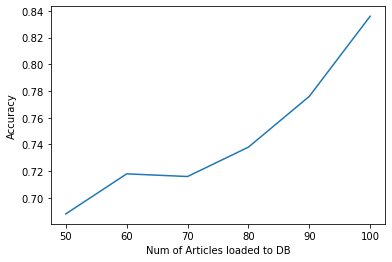
****

Figure : Full classes test - Accuracy analysis

As described in Figure 17, the accuracy rises from 68% with 50 articles in the topic data frame (per class) test to 83% with 100 articles in the topic data frame (per class).

## Conclusions

This paper presents an implementation of an XAI model that offers users the ability to identify and explain content sent to them by an AI recommendation system (and/or any other AI service/procedure).

The offered model, which can enrich user XAI functionality and capabilities, allows different decisions made by the AI system regarding the user to be revealed and explained. Currently, it may be that a user receives content/information/services, and that a system (AI system) identifies the user (via a cluster/profile or other segmentation procedure) with [a] proper/relevant class[es] of information.

The model demonstrated here allows (in some cases) classifications of the content to be identified to the user. This can allow the user to understand the rationale for the decisions made and hence find logic and trust in the system/service that analyzed their preferences and provided content (or services) based thereon.

The research consisted of twelve main experiments with different scenarios that demonstrate the ability of a user (based on a topic data frame that the service provides to the user) to identify (with a minor procedure on the user side) different content (in our demonstration, we used content based on classifications of an articles dataset) and predict the classification of the content.

The approach achieved up to 83% accuracy (prediction of five different classes) in the user prediction procedure when testing all classifications on the user side, and even better accuracy in the case of detecting only one class (prediction of one class out of five, as elaborated in the paper).

This research initiates a novel new domain of work, which needs further elaboration and study in different aspects, such as: What is the most efficient dataset/content that the user should use for effective XAI? What should the relationship between the service/content/AI system and the user be? How does a user progress to the next steps, such as going beyond awareness and explanation of the AI system’s decisions and outcomes, to also maybe actively updating, changing, and reacting to these?

## References (MLA)

1. Pandey, Sandeep, et al. “Learning to target: what works for behavioral targeting.” *Proceedings of the 20th ACM international conference on Information and knowledge management*. 2011.‏
2. Shah, Neil, et al. “Research Trends on the Usage of Machine Learning and Artificial Intelligence in Advertising.” *Augmented Human Research* 5.1 (2020): 1-15
3. Qiu, Junfei, et al. “A survey of machine learning for big data processing.” *EURASIP Journal on Advances in Signal Processing* 2016.1 (2016): 1-16.
4. Hernández-Orallo, José. “Evaluation in artificial intelligence: from task-oriented to ability-oriented measurement.” *Artificial Intelligence Review* 48.3 (2017): 397-447. ‏
5. Adam, Bernard, and Ian F. Smith. “Reinforcement learning for structural control.” *Journal of Computing in Civil Engineering* 22.2 (2008): 133-139.
6. Etzion, Dror, and J. Alberto Aragon-Correa. “Big data, management, and sustainability: Strategic opportunities ahead.” (2016): 147-155.
7. James, Manyika, et al. “The next frontier for innovation, competition, and productivity.” Big data (2011).‏
8. Fan, Wei, and Albert Bifet. “Mining big data: current status, and forecast to the future.” *ACM SIGKDD explorations newsletter* 14.2 (2013): 1-5. ‏
9. Wahbeh, Abdullah H., et al. “A comparison study between data mining tools over some classification methods.” *International Journal of Advanced Computer Science and Applications* 8.2 (2011): 18-26.
10. Kotsiantis, Sotiris B., I. Zaharakis, and P. Pintelas. “Supervised machine learning: A review of classification techniques.” Emerging artificial intelligence applications in computer engineering 160.1 (2007): 3-24.
11. Madhulatha, T. Soni. “An overview on clustering methods.” *arXiv preprint arXiv:1205.1117* (2012).
12. Wang, Jun, Weinan Zhang, and Shuai Yuan. “Display advertising with real-time bidding (RTB) and behavioural targeting.” *arXiv preprint arXiv:1610.03013* (2016).
13. Popov, Artem, and Daria Iakovleva. “Adaptive look-alike targeting in social networks advertising.” *Procedia computer science* 136 (2018): 255-264.
14. Jinghua, Jiang, et al. “Towards Privacy-Preserving User Targeting.” (2016): 22-32.
15. Dastin, Jeffery. “Amazon scraps secret AI tool that showed bias against women.” Reuters, 11 October 2018,

<https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G>.

1. BBC News. “Apple’s ‘sexist’ credit card investigated by US regulator.” BBC, 11 November 2019.

<https://www.bbc.com/news/business-50365609>

1. de Fine Licht, Karl, and Jenny de Fine Licht. “Artificial intelligence, transparency, and public decision-making.” *AI & society* 35.4 (2020): 917-926.
2. Doran, Derek, Sarah Schulz, and Tarek R. Besold. “What does explainable AI really mean? A new conceptualization of perspectives.” *arXiv preprint arXiv:1710.00794* (2017).
3. Lipton, Zachary C. “The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery.” *Queue* 16.3 (2018): 31-57.
4. Murdoch, W. James, et al. “Interpretable machine learning: definitions, methods, and applications.” *arXiv preprint arXiv:1901.04592* (2019).‏
5. Adadi, Amina, and Mohammed Berrada. “Peeking inside the black-box: a survey on explainable artificial intelligence (XAI).” *IEEE access* 6 (2018): 52138-52160.
6. Yu, Ronald, and Gabriele Spina Alì. “What’s inside the Black Box? AI Challenges for Lawyers and Researchers.” *Legal Information Management* 19.1 (2019): 2-13.
7. Arrieta, Alejandro Barredo, et al. “Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI.” *Information Fusion* 58 (2020): 82-115.
8. Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. ““ Why should I trust you?” Explaining the predictions of any classifier.” *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016.‏‏
9. Baehrens, David, et al. “How to explain individual classification decisions.” *The Journal of Machine Learning Research* 11 (2010): 1803-1831
10. Castelluccia, Claude, Mohamed-Ali Kaafar, and Minh-Dung Tran. “Betrayed by your ads!.” *International Symposium on Privacy Enhancing Technologies Symposium*. Springer, Berlin, Heidelberg, 2012.
11. Book, Theodore, and Dan S. Wallach. “An empirical study of mobile ad targeting.” *arXiv preprint arXiv:1502.06577* (2015).‏
12. Bouchacourt, Diane, and Ludovic Denoyer. “Educe: Explaining model decisions through unsupervised concepts extraction.” *arXiv preprint arXiv:1905.11852* (2019).
13. Curiskis, Stephan A., et al. “An evaluation of document clustering and topic modelling in two online social networks: Twitter and Reddit.” *Information Processing & Management* 57.2 (2020): 102034.‏
14. Negara, Edi Surya, Dendi Triadi, and Ria Andryani. “Topic Modelling Twitter Data with Latent Dirichlet Allocation Method.” *2019 International Conference on Electrical Engineering and Computer Science (ICECOS)*. IEEE, 2019.‏
15. Blei, David M. “Probabilistic topic models.” *Communications of the ACM* 55.4 (2012): 77-84.‏
16. Pavlinek, Miha, and Vili Podgorelec. “Text classification method based on self-training and LDA topic models.” *Expert Systems with Applications* 80 (2017): 83-93
17. Liu, Kang, Song Gao, and Feng Lu. “Identifying spatial interaction patterns of vehicle movements on urban road networks by topic modelling.” *Computers, Environment and Urban Systems* 74 (2019): 50-61.‏
18. Colace, Francesco, et al. “Towards labour market intelligence through topic modelling.” *Proceedings of the 52nd Hawaii International Conference on System Sciences*. 2019.‏
19. Kolini, Farzan, and Lech Janczewski. “Clustering and topic modelling: A new approach for analysis of national cyber security strategies.” (2017).‏
20. Mikolov, Tomas, et al. “Distributed representations of words and phrases and their compositionality.” *Advances in neural information processing systems*. 2013.‏
21. Mikolov, Tomas, et al. “Advances in pre-training distributed word representations.” *arXiv preprint arXiv:1712.09405* (2017).‏
22. Goldberg, Yoav, and Omer Levy. “word2vec Explained: deriving Mikolov et al.’s negative-sampling word-embedding method.” *arXiv preprint arXiv:1402.3722* (2014).‏
23. Le, Quoc, and Tomas Mikolov. “Distributed representations of sentences and documents.” *International conference on machine learning*. PMLR, 2014.‏
24. Lau, Jey Han, and Timothy Baldwin. “An empirical evaluation of doc2vec with practical insights into document embedding generation.” *arXiv preprint arXiv:1607.05368* (2016).‏
25. Blei, David M., Andrew Y. Ng, and Michael I. Jordan. “Latent dirichlet allocation.” *the Journal of machine Learning research* 3 (2003): 993-1022.
26. Greene, Derek, and Pádraig Cunningham. “Practical solutions to the problem of diagonal dominance in kernel document clustering.” *Proceedings of the 23rd international conference on Machine learning*. 2006.
27. Rehurek, Radim, and Petr Sojka. “Software framework for topic modelling with large corpora.” *In Proceedings of the LREC 2010 workshop on new challenges for NLP frameworks*. 2010.
28. Pedregosa, Fabian, et al. “Scikit-learn: Machine learning in Python.” *the Journal of machine Learning research* 12 (2011): 2825-2830.

1. www.amazon.com [↑](#footnote-ref-1)
2. www.apple.com [↑](#footnote-ref-2)
3. www.google.com [↑](#footnote-ref-3)