Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 207 (2022) 1017–1026

Procedia Computer Science

www.elsevier.com/locate/procedia

26th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2022)

Sustainable Personalisation and Explainability in Dyadic Data Systems

Jorge Paz-Ruza*[a], Carlos Eiras-Franco[a], Bertha Guijarro-Berdiñas[a], Amparo Alonso-Betanzos[a]

[a] *CITIC. Universidade da Coruña. 15071 A Coruña, Spain*

Abstract

Systems that rely on dyadic data, which relate entities of two types together, have become ubiquitously used in fields such as media services, tourism business, e-commerce, and others. However, these systems have had a tendency to be *black-box* systems, despite their objective of influencing people's decisions. There is a lack of research on providing *personalised explanations* to the outputs of systems that make use of such data, that is, integrating the idea of Explainable Artificial Intelligence into the field of dyadic data. Moreover, the existing approaches rely heavily on Deep Learning models for their training, reducing their overall sustainability. In this work, we propose a computationally efficient model which provides personalisation by generating explanations based on user-created images. In the context of a particular dyadic data system, the restaurant review platform *TripAdvisor*, we predict, for any (user,restaurant) pair, the review of the restaurant that is most adequate to present it to the user, based on their personal preferences. This model exploits the usage of efficient Matrix Factorisation techniques combined with feature-rich embeddings of the pre-trained Image Classification models, developing a method capable of providing transparency to dyadic data systems while reducing as much as 80% the carbon emissions of training compared to alternative approaches.

Keywords: Image-based personalisation; Explainable Artificial Intelligence; Dyadic Data; Matrix Factorisation; Image Classification

1. Introduction

Dyadic data models and systems have been used over multiple information filtering contexts to improve user/customer satisfaction. However, they still show a strong tendency to be *black-box* systems, as they may not be transparent when presenting their outputs. Moreover, the use of Deep Learning (DL) models to achieve high precision in these systems not only compromises transparency, but also the environmental efficiency of the models, due to the high carbon footprint of this AI approach [1]. In this work, we will explore the idea of *Explainability* in a particular application of dyadic data, aiming also at using efficient while sustainable approaches.

* Corresponding author. Email: j.ruza@udc.es

10.1016/j.procs.2022.09.157

Among other classical applications exploiting the characteristics of dyadic data, Recommender Systems (RS) have obtained clear success as tools that enhance user engagement and generate business. Netflix, for instance, revealed that *"(...) now 75% of what people watch is from some sort of recommendation. (...)"* [2]. Therefore, embedding *explainability* in our system will enable users to understand the recommendations they receive (answering the question *"Why am I being recommended this item"?*). Moreover, as each user is different and has distinctive preferences, the explanation of an item is typically not the same for every user: integrating explainability in RS can be considered, by all means, a way of adding **personalisation** to it. This is, we aim at knowing ***how to recommend***.

Our proposal will explore the idea of obtaining personalised explanations by means of existing images of reviews in the TripAdvisor platform, searching for the photograph that best complements the recommendation of a restaurant to a user, according to their preferences. This is akin to recreating the review that the user would post if they had visited the recommended restaurant previously. Furthermore, existing approaches that make use of user-created content, such as ELVis [3], rely heavily on training DL models, which carry a sizeable carbon footprint [1]. Consequently, of our main research objectives will be assessing the viability of more computationally efficient methodologies, specifically Matrix Factorization (MF) techniques, when seeking transparency in systems that deal with dyadic data, as a step to reduce the environmental impact of models oriented towards XAI (Explainable Artificial Intelligence).

2. Materials

With the objective of training our model using user-uploaded images, we gathered large amounts of reviews from the *TripAdvisor* platform over six cities of varying sizes and cultures. Table 1 presents basic information of each city's dataset, while Figure 1 shows the amount of users that have posted *n* reviews in the platform for each city.

Table 1. Information of each city's dataset including raw review count, as well the number of *Unique users* and *Unique restaurants*.

Dataset	No. of Reviews (Total)	Unique users	Unique restaurants
Gijón	18,679	5,139	598
Barcelona	150,416	33,537	5,881
Madrid	203,905	43,628	6,810
New York City	231,141	61,019	11,982
Paris	251,636	61,391	11,982
London	479,798	134,816	13,888

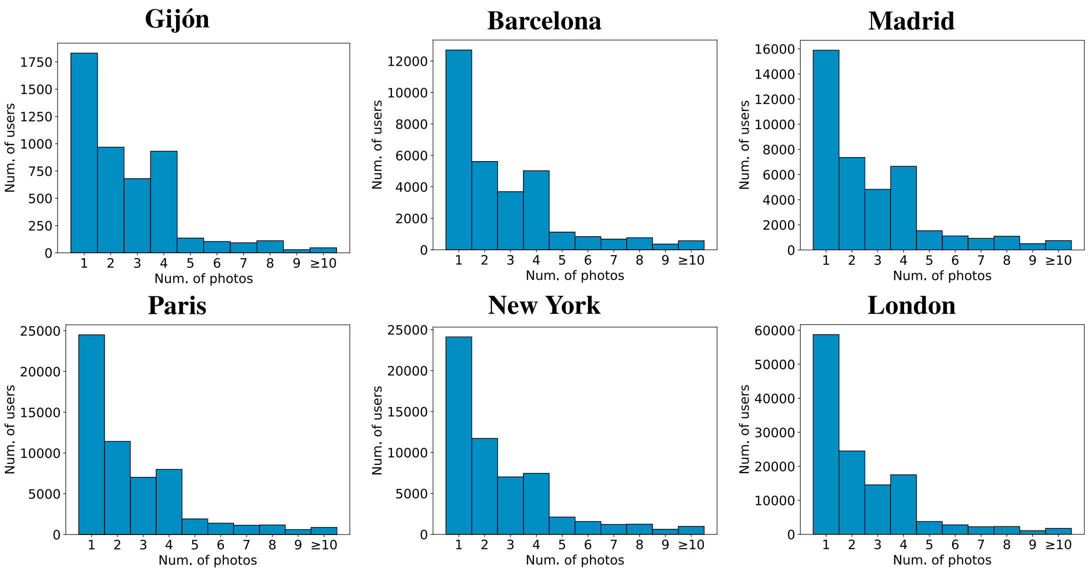

Fig. 1. Distribution showing the number of images per user in each individual city dataset.

Based on this, we may conclude there is a clear problem of ***sparsity*** in our data, caused by the large amount of inactive users in the TripAdvisor platform. As seen in Table 1, on average we will find a new user for each four reviews. This lack of knowledge about a majority of the users has an impact on the training process of Machine Learning algorithms, relatable to the *cold start problem* in the context of RS. Likewise, in figure 1 most users have low review counts, that is, they are rather *inactive* in the *TripAdvisor* platform.

3. Methods

As we intend to use the review images as inputs to our model, we are required to map or *embed* them to a numerical representation from a visual context. In this section, we address how Image Classification and Matrix Factorisation techniques enable us to create vector representations of the user's content and use them in our XAI model, respectively.

3.1. Image Classification

The dataset created and provided in [3] has been chosen for the image embeddings needed to train and test our explainable model. To obtain these low-dimensional image embeddings, the model used was *Inception-ResNet-v2* [4], a Deep Convolutional Neural Network which takes 299 × 299 pixels RGB images as input, trained over one million images from the ImageNet database. This 164-layer-deep network provides as its output a vector of estimated class probabilities. The extensive usage of residual connections, which act as "shortcuts" in the model, allows to mitigate the so-called "degradation problem" and accelerates the training, thus being able to train larger models more efficiently.

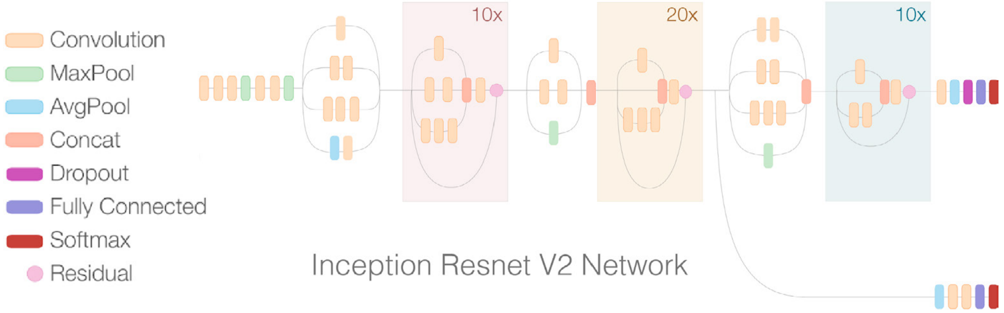

Fig. 2. Schematic diagram of Inception-ResNet-v2, a Deep Convolutional Neural Network used to create image embeddings in this work.

3.2. Matrix Factorisation in Recommender Systems

Matrix Factorisation (MF) is a widely used approach to deal with dyadic data models, and has the advantage of posing a lower computational overhead than more complex, black-box approaches like those based on DL. To obtain personalisation, we will try to exploit MF as a simple but powerful technique in terms of accurate representation of our data entities, as well as an efficient and environmentally sustainable alternative to DL models.

The key aspect of MF is its ability to map both users and items to a low-dimensional latent feature space. In an information system which tracks the data of n users and m items, the full user-item interaction matrix has $n \times m$ entries. Factorising this matrix, we instead obtain two matrices U and V, which in total have $(n+m)d$ entries, where $d \ll m$ and $d \ll n$, and d represents the size of the *latent space*. Alternatively, we can describe d as the amount of *latent features* we are using to describe our user's preferences (U) and the item's characteristics (V). As it can be observed in Figure 3, the lower-dimensional Item and User matrices may be arbitrarily small compared to the original full interaction matrix [5]. Once we have obtained these two matrices U and V, predictions for $(user, item)$ ratings can be efficiently modeled as a simple dot product of vectors of size d, which is a computationally cheap operation.

Opportunely, our numerical item information (the embedded representations of images in reviews) matches the latent feature space of MF techniques, enabling us to use them as direct latent feature representations that together act as the factorised Item matrix. This way, we can make use of **content-based filtering**, utilizing them to draw similarities between users' preferences.

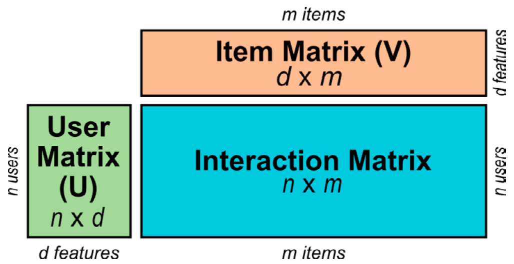

Fig. 3. By using MF techniques, the full (user-item) interactions matrix can be represented with two lower-dimensional matrices: the Item Matrix, which represents the latent features of each item, and the User Matrix, which represents the users' preferences regarding those latent features

4. Proposed Method

As it is typical in systems dependant on dyadic data, we will consider a set of users U and a set of items I. In the context of the TripAdvisor platform, U represents the group of users that have posted reviews in restaurants of a given city, whereas I is the set of those restaurants. However, it is important to remark that our intention is not to provide recommendations of the form $(\boldsymbol{u}, \boldsymbol{it})$, where $\boldsymbol{u} \in U$ and $\boldsymbol{it} \in I$. Instead, following a previous recommendation (or any other type of presentation of a restaurant to a user), our aim is to provide an explanation of it. To achieve this, we will use the existing images (from now on, contents) by other users of the same restaurant. Thus, we can define:

- $C(\boldsymbol{u})$: the set of contents posted by user $\boldsymbol{u} \in U$
- $C(\boldsymbol{it})$: the set of contents posted about the restaurant $\boldsymbol{it} \in I$

and immediately, by extension:

- $C(\boldsymbol{u}, \boldsymbol{it})$: the set of contents posted by user $\boldsymbol{u} \in U$ about restaurant $\boldsymbol{it} \in I$

In this work, our aim is to provide users with images that reflect their personal preferences when presenting the restaurant to them. Our approach to achieve this will be to portray these preferences as a correlation to content *authorship*. This is, our concern will be to create a model able to identify which contents of a given restaurant may have been uploaded or *authored* by a given user, as we understand that these are representative of that user's preferences regarding restaurants of the *TripAdvisor* platform. Therefore, we will establish an Explainer for a RS where the set of users corresponds directly to U, the users of the platform, and the set of items corresponds instead to C, the group of contents of the restaurants of a given city, denoted as $C(\boldsymbol{it})$.

Every element of this matrix corresponds to a labeled pair

$$Pr(\boldsymbol{u}, \boldsymbol{c}) = \begin{cases} 0 & \boldsymbol{c} \notin C(\boldsymbol{u}) \\ 1 & \boldsymbol{c} \in C(\boldsymbol{u}) \end{cases} \tag{1}$$

where again $\boldsymbol{u} \in U$ represents a user, and $\boldsymbol{c} \in C$ is an individual piece of content attached by a user in a review. The label holds the meaning of the *authorship* of said content. Therefore, each element of the matrix $C \times U$ will correspond to 1 if the user is the *author* of that content, and 0 otherwise. Consequently, when personalising a recommendation of an item $\boldsymbol{it}$ to a user $\boldsymbol{u}$, we can provide an explanation by showing to said user the content $\boldsymbol{c}*$ that best represents the user's preferences. To achieve this, we will predict which of the available pieces of content for that item the user is most likely to have authored. Formally, we can define these predictions as $\boldsymbol{c}* = \arg\max_{c \in C(it)} Pr(\boldsymbol{u}, \boldsymbol{c})$, with $Pr(\boldsymbol{u}, \boldsymbol{c})$ denoting the prediction about a user $\boldsymbol{u}$ being the author of content $\boldsymbol{c}$.

4.1. Dataset Preparation

In this subsection, we present the particularities of the method used to prepare the datasets presented in Section 2, in order to be then fed to the system for training or testing.

Prior to performing *train/test* or *oversampling*, we undertake a simple step that carries a large impact on the sustainability of the system: all contents are *embedded* using the Inception-Resnet-V2 model, and saved in an auxiliary array, which is then loaded at the start of each training/test execution. This approach allows for a higher computational efficiency that embedding them in run-time.

4.1.1. Dataset Partitioning

It is relevant noticing that, due to the context of the problem, a straightforward percent splitting of the dataset (e.g. 70% Train, 15% Dev, 15% Test) is not feasible. By the intrinsic rules of the *authorship* prediction method, and the *content-based* RS approaches, it is not possible for the system to know the preferences of users it has not "seen" before. As a majority of users only have uploaded one review to the platform, testing over users the system has not trained with before would be bound to happen. To solve this, we will apply a customised partitioning method akin to the one used in [3] that ensures that all tested users have been previously "seen" in the training phase, even if with minimal information, avoiding the encounter of strict *cold start* situations:

1. For those users that have at least two reviews (each review includes up to four images), one of them is reserved for the Test partition, and the remaining belong to the Train+Dev partition.
2. Using the same procedure, the Train+Dev partition is split between Train and Dev partitions.

It must be pointed out that at this point we still only have *positive samples* in the partitioned dataset, that is, samples $(\boldsymbol{u}, \boldsymbol{c})$ such that $Pr(\boldsymbol{u}, \boldsymbol{c}) = 1$. To generate negative samples, we decided to assume that no user would have authored content that they didn't upload themselves. We chose this as a computationally inexpensive method to generate negative samples, usual in the field of PU Learning [9] Based on this, we propose the following sampling method:

- In **training** sets, for each positive sample $(\boldsymbol{u}, \boldsymbol{c})$, denoting user $\boldsymbol{u}$ uploaded content $\boldsymbol{c}$ of restaurant $\boldsymbol{it}$, we add:
  - 10 negative samples $(\boldsymbol{u}, \boldsymbol{c}')$, where $\boldsymbol{c}'$ is a content of the same restaurant $\boldsymbol{it}$ uploaded by a different user $\boldsymbol{u}'$.
  - 10 negative samples $(\boldsymbol{u}, \boldsymbol{c}'')$, where $\boldsymbol{c}''$ is a content of a different restaurant $\boldsymbol{it}'$ uploaded by a different user $\boldsymbol{u}'$.
  - 19 copies of the positive sample $(\boldsymbol{u}, \boldsymbol{c})$, to compensate the 20 negative samples we have added.
- In **testing** sets, for each positive sample $(\boldsymbol{u}, \boldsymbol{c})$, we will add negative samples $(\boldsymbol{u}, \boldsymbol{c}')$ with all the contents of the same restaurant $\boldsymbol{it}$ that appear in the training set.

4.2. Proposed Model

To achieve the purpose of personalising explanations, we present an XAI model architecture to predict the *authorship* of uploaded content (review images) by a given user. The layers that conform this architecture for *authorship* prediction are summarised in Figure 4, and can be grouped in two separate blocks:

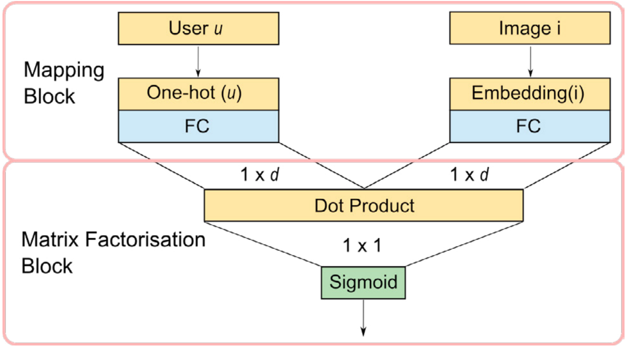

Fig. 4. Overview of the architecture of the proposed *authorship* prediction model. *d* denotes the size of the subspace where the dot product operates (or alternatively, the no. of latent features extracted from contents), with $d = 1024$ in our model. The Embedding block represents image embedding obtained with the Inception network.

- The **Mapping** block has the purpose of codifying the input data (user identifier and content embedding) to the subspace of size d (in our case, $d = 1024$, where we exploit the usage of MF techniques:
  - Each user $\boldsymbol{u}$ is first codified with a *one-hot* codification, and then mapped into a d-dimensional embedding by means of a dense Embedding layer.
  - Each content $\boldsymbol{c}$ is represented by a real-valued vector of size 1536 obtained with the Inception network, and transformed by a Fully Connected (FC) layer to obtain a lower dimensional vector of d elements.

  The results of this mapping block are two vectors of size d which, in terms of MF and RS, will represent the latent vectors of **user preferences** and **item features**, respectively.
- The **Matrix Factorisation** block applies the basic concepts of obtaining content-based recommendations with MF discussed in Section 3.2. As the output of the aforementioned mapping block contains the desired latent item features and user preferences in a low-dimensionality subspace of size d, it is now possible to apply a simple dot product between both vectors. The result of this dot product operation is a single Float value that effectively represents the similarity or *affinity* between the user and the content, and therefore, a joint prediction of the user's *authorship* of said content. Finally, a sigmoid activation function is used to produce a probability output in the [0, 1] range. This probability denotes what we defined as $Pr(\boldsymbol{u}, \boldsymbol{c})$ at the beginning of this section, viz. the probability that user $\boldsymbol{u}$ authored content $\boldsymbol{c}$. The final configuration of the different layer sizes and training parameters of the model was obtained through numerous grid-search processes using each individual city dataset, and an *Adam* optimizer with Binary Cross-Entropy (BCE) as loss function.

5. Evaluation Methods

To evaluate the performance of this model, an *implicit* evaluation method was conceived, obtaining the experimental results relying on the observed user behaviour and analysing it [6]. Here, we will make a reasonable assumption: if the model is able to place the user's own content on top of the ranking when mixed with all the other images, this means it was able to correctly predict its authorship, and therefore **capture the user's preferences**. Consequently, for any (*user*, *restaurant*) pair, the top ranked contents will be representative of these preferences, being comparable to an *explicit* user satisfaction measure. We used two different metrics to evaluate the quality of these rankings:

As a simpler metric we chose to use **Recall at** k; in our case, this directly defines how often our model ranks the user's real uploaded content, namely $\boldsymbol{c}$, among the top k positions of the ranking. This is computed as

$$Recall@k = \frac{|\{Relevant\ Documents\}| \cap |\{Top\ k\ Retrieved\ Documents\}|}{|\{Relevant\ Documents\}|} \tag{2}$$

However, it is important to remind that this metric is heavily biased depending on the size of the rankings, and therefore our main measurement of quality will be an unbiased *percentile* metric, computed as

$$percentile(\boldsymbol{c}, \boldsymbol{R}) = \frac{pos(\boldsymbol{c}, \boldsymbol{R}) - 1}{|\boldsymbol{R}|} \times 100 \tag{3}$$

where $\boldsymbol{c}$ is the content uploaded by user $\boldsymbol{u}$ at restaurant $\boldsymbol{it}$, $\boldsymbol{R}$ is the ranking created by the model for the pair $(\boldsymbol{u}, \boldsymbol{it})$, and $pos(\boldsymbol{c}, \boldsymbol{R})$ is the position of $\boldsymbol{c}$ in said ranking. As the ranking is sorted in descendant $Pr(\boldsymbol{u}, \boldsymbol{c})$ order, a lower *percentile* metric implies a better ranking. Additionally, to test the computational efficiency and sustainability of our model against the existing user content-based approach ELVis [3], which follows a Deep Learning approach, we will compare for each dataset the training times, as well as the CO2 emissions using the *CodeCarbon* package [7].

As an example of the overall functioning of our proposed model and evaluation methods, Figure 5 shows the predicted ranking for a user in a specific restaurant. The photos uploaded by the user (top row) mainly show design plating, drinks and food close-ups. After computing and sorting the ranking of predicted probabilities $Pr(\boldsymbol{u}, \boldsymbol{c})$ for all of the photographs in the restaurant, we observe the ones at the topmost of the ranking are the ones that best match the user's preferences, and vice-versa. Moreover, the user's own photo was placed in the second place of ranking. With this, we can conclude that our model was able to grasp the peculiar preferences of the user, and more relevantly, that our *percentile* metric can quantitatively measure the quality of these rankings.

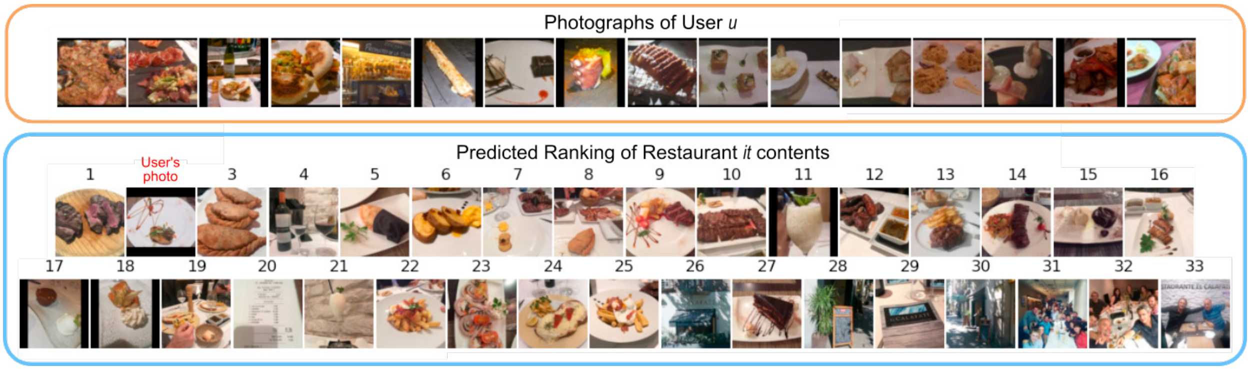

Fig. 5. Ranking of the predicted authorship of a user for the restaurant in the famous Casa Milà, in Barcelona.

6. Experimental Results

This section presents the experimental results obtained with the methods established in Section 5. To analyze the effect of inactive users in the results, we computed the *median percentile* with 100 different "active user" thresholds in each city, accounting only those samples from users with more than n photos in the Train set, for $n \in [0, 100]$. In the case of the *Recall @ k* metric, we consider two cases: all users, and only those with $n > 10$ (active users).

Figure 6 shows the obtained percentile metrics over the Test partition from each of available datasets, where the X axis represents said "activity threshold", comparing it to two baseline methods: a Random (RND) method, where the user's photo is placed randomly in the ranking, and a Centroid (CNT) method, where the closer an image is to the centroid of the user's photos, the higher it will be on the ranking. As expected, our model easily outperforms these *baseline* methods. We can see that for these comparative algorithms, the amount of training information is not really relevant. Our model exhibits good results, and consolidates one of our prime expectations about the problem: its performance increases when applied to users for whom we have more information. This is reflected in the percentile figures: the percentile value decreases (therefore a better result) when we test the model with users with more training images. Moreover, even near a *cold start* situation (left-most side of the graphics), we are still able to produce decently accurate rankings, notably better than the defined baseline methods. Also, as it can be observed, the small size of the Gijon dataset causes the performance to be highly erratic and not representative of any of the algorithms' performance. Conversely, the best scores were obtained in those datasets with larger photograph counts.

Likewise, Table 2 shows for all six cities the percentage of test cases where the user's real photo is ranked in the Top-k positions (*Recall at k*), both taking into account *all* users (upper table), and considering only active users with $n > 10$ (lower table). It also contains the scores obtained by *ELVis* [3]. In all cases, in order to obtain representative results, we have filtered out the data from restaurants which have less than 10 photographs and included at most 100 of the available ones. As we can observe, for every individual method, all cities show relatively similar scores in this *Recall* metric. Regarding the individual methods, the baseline method CNT is the worst in every case; this is a trend also observable when analysing the results derived from the *percentile* metric. We can derive that while the image embeddings provided by the Inception network are good vector representations of the photos, these do not really distill the semantic rules that define each user's preferences; we must learn those through an active learning process.

Regarding our model's performance (MDL), it again exhibits a noticeably good performance when compared to the *baseline* methods. When placing the user's photo among the Top 10 positions of the ranking, our proposal is around 15 percentage points better than *Random* in all cities when taking into account "active" users, and about 10 points better if considering all users. As observed previously, the performance in Gijón is rather volatile due to the small size of the dataset.

It is not trivial to determine from Table 2 whether our model's performance is competitive with the state of the art model ELVis: MDL matches or surpasses ELVis' accuracy in 2 of 6 datasets, while ELVis performs better on the rest. For this reason, we applied the Nemenyi post-hoc test [8] with $\alpha = 0.05$. Results can be seen in Figure 7, showing that there is no statistical significant differences among them.

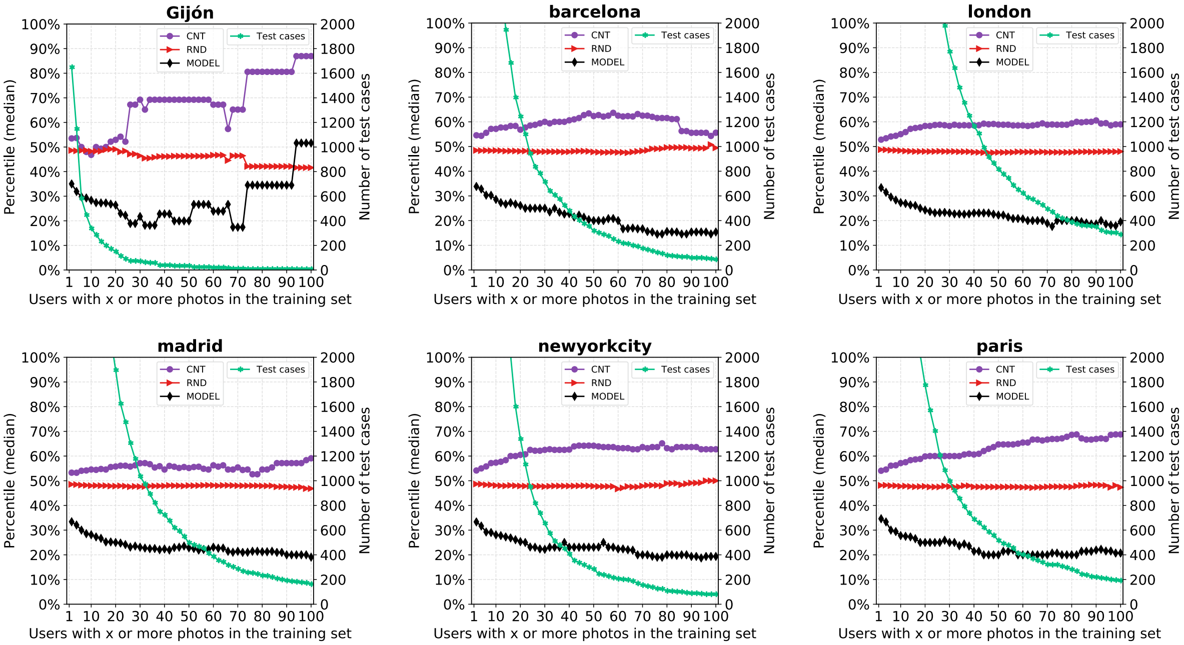

Fig. 6. Median percentiles of the user's real photographs in the predicted rankings for each city dataset. The X axis represent the minimum amount of photos by the user in the training test to be included in the computation. The green curve represents the number of test cases (total no. of rankings) available with each threshold value.

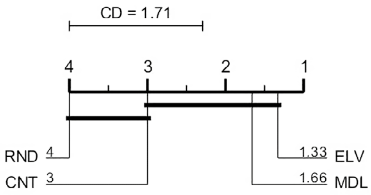

Fig. 7. Nemenyi test with using the *Recall at k* metric, for $k = 10$ using active users (more than 10 images in training set).

6.1. Sustainable Machine Learning

As we have previously mentioned, our aim is to propose a computationally efficient model with a good balance between accuracy and training time (and therefore carbon emissions). Therefore, in this section we contrast the efficiency of our proposed model against the reference agent ELVis [3]models'. In the same task, ELVis utilizes a model architecture with a higher focus in deep-learning, whereas our model puts an emphasis in exploiting the simplicity of Matrix Factorisation techniques to still achieve ambitious results. As seen in Table 2 our results are notably competitive, specially when considering the percentage of photos placed in the very first positions of the ranking, with ELVis being about 3 percentage points better. On the other hand, analyzing the performance for users with low content counts, we can suspect that ELVis' higher complexity allows for a better fit to the data of those users with little to no information. However, as mentioned beforehand, an expected advantage of basing our model in MF techniques was to have an edge regarding the computational cost of the learning process. ELVis' complex architecture implies in the long run a need to train the model for longer (in ELVis' case, 100 epochs). On the other hand, the architectural simplicity of our proposal allowed to obtain our best results with a training length of 15 epochs.

Table 3 contains the comparison of training times and CO2 emissions during training for each city. As it shows, we are able to consistently obtain noticeably lower training times and environmental impact across all cities, which

Table 2. For each dataset, comparison of the *Recall@k* (*Top-k*) metric between RND, CNT, ELVis, and our model (MDL); larger values imply more test cases in the Top-k of the ranking, thus better results. The number of test cases in each city is shown in parenthesis. The upper table considers all test cases, and the lower table only test cases about users with $n > 10$ reviews in the Train set. To obtain relevant results, we only include restaurants with at least 10 reviews, and use at most 100 of them. The results highlighted in bold are the best performing model for each k in each city.

TOP	Gijón (2,005)				Barcelona (15,342)				Madrid (22,384)			
	RND	CNT	ELVis	MDL	RND	CNT	ELVis	MDL	RND	CNT	ELVis	MDL
1	3.0%	1.5%	5.2%	**6.8%**	3.1%	1.5%	**8.3%**	7.1%	2.8%	1.4%	**7.6%**	6.6%
2	5.9%	3.5%	9.6%	**10.7%**	6.3%	3.3%	**14.4%**	12.3%	5.6%	3.1%	**13.4%**	11.5%
3	8.7%	5.0%	13.6%	**14.3%**	9.4%	5.4%	**19.7%**	16.6%	8.4%	5.1%	**18.8%**	15.5%
4	11.4%	7.7%	**18.3%**	17.6%	12.5%	7.9%	**24.6%**	20.9%	11.2%	7.3%	**23.5%**	19.2%
5	14.3%	10.1%	**22.2%**	21.9%	15.7%	10.2%	**29.4%**	24.9%	14.0%	9.8%	**27.7%**	22.7%
6	17.1%	12.7%	**25.9%**	25.4%	18.8%	13.0%	**33.5%**	28.3%	16.7%	12.3%	**31.7%**	26.1%
7	20.0%	15.2%	**29.8%**	28.1%	22.0%	15.9%	**37.5%**	31.7%	19.5%	15.0%	**35.1%**	29.2%
8	23.1%	17.6%	**32.6%**	31.0%	25.2%	18.9%	**41.0%**	34.9%	22.3%	17.8%	**38.3%**	32.1%
9	26.0%	21.0%	**35.2%**	34.2%	28.3%	22.3%	**44.4%**	38.3%	25.1%	20.9%	**41.3%**	34.9%
10	29.0%	24.9%	**37.8%**	36.7%	31.4%	26.1%	**47.1%**	41.5%	27.9%	24.3%	**44.1%**	37.5%
TOP	New York (28,531)				Paris (23,450)				London (53,901)			
	RND	CNT	ELVis	MDL	RND	CNT	ELVis	MDL	RND	CNT	ELVis	MDL
1	2.6%	1.3%	**7.1%**	6.1%	3.8%	1.9%	**10.0%**	8.3%	2.5%	1.3%	5.9%	**6.0%**
2	5.1%	2.8%	**12.6%**	10.5%	7.5%	3.9%	**17.2%**	14.2%	4.9%	2.8%	**10.3%**	**10.3%**
3	7.5%	4.4%	**17.4%**	14.3%	11.3%	6.3%	**23.4%**	19.4%	7.4%	4.4%	**14.7%**	14.3%
4	10.1%	6.3%	**21.7%**	17.8%	15.0%	9.3%	**29.1%**	24.1%	9.9%	6.3%	**19.9%**	17.8%
5	12.6%	8.4%	**25.8%**	21.0%	18.7%	12.3%	**34.7%**	28.6%	12.4%	8.3%	**25.8%**	21.0%
6	15.2%	10.7%	**29.5%**	23.9%	22.4%	15.8%	**39.6%**	32.7%	14.8%	10.7%	**28.0%**	24.3%
7	17.7%	13.2%	**33.2%**	26.6%	26.2%	19.7%	**43.8%**	36.7%	17.3%	13.2%	**31.0%**	27.4%
8	20.2%	15.8%	**36.2%**	29.2%	29.9%	23.4%	**47.8%**	40.0%	19.8%	15.7%	**34.5%**	30.4%
9	22.7%	18.5%	**38.9%**	31.8%	33.5%	27.7%	**51.5%**	43.5%	22.3%	18.4%	**37.2%**	33.1%
10	35.3%	21.5%	**41.4%**	34.2%	37.3%	32.4%	**54.8%**	46.8%	24.8%	21.4%	**41.1%**	35.5%

TOP	Gijón (338)				Barcelona (3,023)				Madrid (4,578)			
	RND	CNT	ELVis	MDL	RND	CNT	ELVis	MDL	RND	CNT	ELVis	MDL
1	4.3%	2.7%	8.9%	**11.2%**	4.0%	1.6%	**11.8%**	10.6%	3.6%	1.6%	**11.9%**	10.0%
2	8.3%	5.9%	**16.3%**	16.0%	7.9%	4.0%	**22.2%**	17.9%	7.5%	3.6%	**20.3%**	17.3%
3	11.8%	7.7%	20.1%	**21.6%**	11.8%	6.1%	**29.3%**	23.4%	11.3%	5.9%	**27.9%**	23.3%
4	15.7%	11.2%	26.6%	**27.2%**	15.8%	9.2%	**34.9%**	30.2%	14.9%	8.8%	**33.5%**	28.6%
5	19.6%	15.7%	29.9%	**33.1%**	19.9%	12.2%	**39.8%**	36.1%	18.7%	11.9%	**38.8%**	33.4%
6	23.7%	18.9%	35.2%	**36.4%**	23.7%	15.9%	**44.9%**	40.3%	22.4%	15.1%	**43.2%**	38.1%
7	27.9%	21.6%	40.2%	**41.1%**	27.7%	20.1%	**49.0%**	43.9%	26.1%	18.5%	**47.0%**	42.4%
8	32.5%	24.0%	42.9%	**43.8%**	31.8%	23.6%	**53.0%**	47.9%	29.9%	22.4%	**50.6%**	45.7%
9	36.9%	27.2%	46.7%	**50.0%**	35.9%	27.9%	**56.3%**	52.0%	33.6%	26.3%	**53.8%**	49.3%
10	40.9%	35.5%	52.1%	**54.7%**	39.9%	32.8%	**59.7%**	55.7%	37.3%	31.1%	**57.2%**	52.8%
TOP	New York (4,230)				Paris (4,625)				London (9,176)			
	RND	CNT	ELVis	MDL	RND	CNT	ELVis	MDL	RND	CNT	ELVis	MDL
1	3.8%	1.6%	**11.6%**	10.3%	4.6%	1.9%	**13.9%**	12.7%	3.4%	1.7%	**11.5%**	9.6%
2	7.4%	3.7%	**20.1%**	18.0%	9.3%	4.3%	**22.5%**	20.6%	6.9%	3.5%	**19.3%**	17.6%
3	11.2%	5.6%	**26.9%**	23.9%	13.8%	6.9%	**29.7%**	27.6%	10.3%	5.4%	**25.5%**	23.6%
4	14.9%	8.0%	**32.4%**	28.9%	18.3%	10.3%	**35.8%**	33.6%	13.7%	7.8%	**30.9%**	28.7%
5	18.6%	11.3%	**36.8%**	33.4%	22.8%	13.5%	**42.0%**	38.4%	17.1%	10.6%	**35.7%**	33.2%
6	22.3%	14.2%	**41.4%**	37.7%	27.3%	17.4%	**47.6%**	43.6%	20.5%	13.9%	**39.9%**	38.2%
7	26.0%	18.3%	**45.3%**	41.4%	31.9%	22.6%	**52.3%**	48.2%	24.0%	17.2%	**43.8%**	42.5%
8	29.7%	22.3%	**49.2%**	45.0%	36.4%	27.4%	**56.5%**	52.5%	27.5%	20.5%	**47.4%**	46.4%
9	33.5%	26.2%	**52.4%**	48.7%	40.8%	33.1%	**60.4%**	56.1%	30.9%	24.3%	**50.1%**	49.8%
10	37.3%	30.1%	**55.3%**	51.6%	45.2%	39.3%	**64.4%**	60.0%	34.2%	28.2%	**53.1%**	**53.1%**

Table 3. Comparison of training times and CO2 emissions between ELVis (DL) and our proposed model, run in the same environment. Times are averaged over 5 executions and standard deviation is included, to account for possible punctual performance issues.

	Total CO2 emissions (g)	
Dataset	**Deep Learning**	**Matrix Factorization**
Gijón	1.5	**0.4**
Barcelona	9.2	**1.8**
Madrid	16.0	**3.5**
New York	42.8	**7.9**
Paris	57.8	**10.0**
London	138.8	**19.1**

	Training Times (s)	
	ELVis	Proposed Model
Gijón	240 ± 4.35	**53.90 ± 1.80**
Madrid	2465 ± 16.1	**696 ± 5.02**
Barcelona	1530 ± 10.4	**436 ± 5.49**
New York	2865 ± 19.5	**746 ± 7.31**
Paris	2940 ± 14.5	**786 ± 6.48**
London	5197 ± 48.9	**1578 ± 17.4**

reflects our expectations about the advantages of using a simpler model. This time and carbon dioxide emissions efficiency is specially remarkable if we take into account the necessary re-training the model needs to undergo if we want to include new users into the system (due to the usage of internal one-hot encoding to codify the user, among other limitations): having a simpler model with noticeably lower training times mitigates the issue of having to train the model from scratch, with the additional advantage of providing a notably lower environmental impact than ELVis.

7. Conclusions and future work

Throughout the conception and development of this research work, we have made several interesting insights into the studied topics, context, methods, tools and experimental results:

- In the context of dyadic data, we encountered the challenge of a widespread absence of pre-labeled datasets, making unfeasible a direct, explicit evaluation of the model. Moreover, the intrinsic scarcity of data, the sparsity of interactions in the context of the *TripAdvisor* platform, added to the absence of direct negative samples, was specially troubling. Nevertheless, by means of reasonable ground truth assumptions, as well as *implicit*, context-specific evaluation methods, we were able to conduct a satisfactory experimentation with our model.
- In relation to the methods used to construct our proposal, we have made an insight into the usefulness of using pre-trained Image Classification models (Inception-ResNet-v2) to efficiently create our proposal, saving significant time and computational costs with *transfer learning* approaches, and therefore providing a more environmentally sustainable methodology for Explainable Artificial Intelligence.
- With respect to MF techniques, we have ascertained its viability in achieving data and computational efficiency in the field of XAI and RS, while at the same time exploiting our image embeddings as representations of latent item features and user preferences.

As future work, we forsee the exploration and integration into our proposal of techniques that may provide an additional improvement both in model accuracy as well as computational efficiency, such as Few-shot learning techniques. We had expected and can appreciate a decline in the performance when little to no information is known about users (which is a major occurrence in *TripAdvisor* and similar platforms), or in cities with low available review data. In order to maximize the usability of our approach, it would be beneficial to consider the usage of models which are able to address our *authorship* prediction problem with few samples. A successful integration of these techniques should also lead to a more efficient learning with a lower environmental impact, which ultimately has been one of the main objectives of this research work.

References

[1] Strubell, E., Ganesh, A. & McCallum, A. Energy and policy considerations for deep learning in NLP. *ArXiv Preprint ArXiv:1906.02243*. (2019)

[2] Blog, N. Netflix Recommendations: Beyond the 5 stars (Part 1). (2012), https://netflixtechblog.com/netflix-recommendations-beyond-the-5-stars-part-1-55838468f429

[3] Diez, J., Perez-Nunez, P., Luaces, O., Remeseiro, B. & Bahamonde, A. Towards explainable personalized recommendations by learning from users' photos. *Inf. Sci.*. **520** pp. 416-430 (2020), https://doi.org/10.1016/j.ins.2020.02.018

[4] Szegedy, C., Ioffe, S., Vanhoucke, V. & Alemi, A. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *Proc. 21st AAAI Conf. on Artif. Intell.* pp. 4278-4284 (2017)

[5] Venkateswaran, S., Senthilkumar, S. & Thandapani, S. Recommendation Systems: Collaborative Filtering using Matrix Factorization — Simplified. (2019), https://medium.com/sfu-cspmp/recommendation-systems-collaborative-filtering-using-matrix-factorization-simplified-2118f4ef2cd3

[6] Herlocker, J., Konstan, J., Terveen, L. & Riedl, J. Evaluating Collaborative Filtering Recommender Systems. *ACM Trans. Inf. Syst.*. **22**, 5-53 (2004,1), https://doi.org/10.1145/963770.963772

[7] Schmidt, V., Goyal, K., Joshi, A., Feld, B., Conell, L., Laskaris, N., Blank, D., Wilson, J., Friedler, S. & Luccioni, S. CodeCarbon: Estimate and Track Carbon Emissions from Machine Learning Computing. (Zenodo,2021), https://github.com/mlco2/codecarbon

[8] Demšar, J. Statistical comparisons of classifiers over multiple data sets. *The Journal Of Machine Learning Research*. **7** pp. 1-30 (2006)

[9] Bekker, J. & Davis, J. Learning from Positive and Unlabeled Data: A Survey. *Mach. Learn.*. **109**, 719-760 (2020,4), https://doi.org/10.1007/s10994-020-05877-5