ContentWise Impressions: An industrial dataset with impressions included

Fernando B. Pérez Maurera
fernandobenjamin.perez@polimi.it
fernando.perez@contentwise.com
Politecnico di Milano, ContentWise
Milan, Italy

Maurizio Ferrari Dacrema
maurizio.ferrari@polimi.it
Politecnico di Milano
Milan, Italy

Lorenzo Saule*
lorenzo.saule@gmail.com
Politecnico di Milano, ContentWise
Milan, Italy

Mario Scriminaci
mario.scriminaci@contentwise.com
ContentWise
Milan, Italy

Paolo Cremonesi
paolo.cremonesi@polimi.it
Politecnico di Milano
Milan, Italy

ABSTRACT

In this article, we introduce the ContentWise Impressions dataset, a collection of implicit interactions and *impressions* of movies and TV series from an Over-The-Top media service, which delivers its media contents over the Internet. The dataset is distinguished from other already available multimedia recommendation datasets by the availability of impressions, i.e., the recommendations shown to the user, its size, and by being open-source. We describe the data collection process, the preprocessing applied, its characteristics, and statistics when compared to other commonly used datasets. We also highlight several possible use cases and research questions that can benefit from the availability of user impressions in an open-source dataset. Furthermore, we release software tools to load and split the data, as well as examples of how to use both user interactions and impressions in several common recommendation algorithms.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; **Data provenance**; **Information integration**.

KEYWORDS

Implicit Feedback, Impressions, Dataset, Collaborative Filtering, Open Source

ACM Reference Format:
Fernando B. Pérez Maurera, Maurizio Ferrari Dacrema, Lorenzo Saule, Mario Scriminaci, and Paolo Cremonesi. 2020. ContentWise Impressions: An industrial dataset with impressions included. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3340531.3412774

*Intern at ContentWise and Ms.C. Student at Politecnico di Milano during the development of this work

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3412774

1 INTRODUCTION

Recommender Systems are, in this era of information, an ubiquitous technology that can be frequently found in the online services we use. The development of new algorithms and techniques has been fueled by the availability of public datasets to the community, collected by both researchers and industry.

The need to develop ever-better solutions is always present, driven by the evolution of business models and the availability of new data sources. An example of this is the RecSys Challenge[1] held every year since 2010. Each year an industry releases a dataset, challenging the participants on a problem that is relevant to their business model, e.g., job recommendation, accommodation recommendation, playlist continuation. Other examples of these competitions are the KDD Cup[2], WSDM CUP[3], and the several IJCAI[4] competitions. A recent emerging trend is to provide the *impressions*, i.e., what was recommended to the user alongside the user interactions. Recent articles, also from industry, propose algorithms that leverage impressions showing they can improve the recommendation quality [7, 14, 20].

Despite this growing research and industrial interest, as well as indications that impressions can be a useful information source, the research community is constrained by the lack of publicly available impression datasets. Most industrial datasets containing impressions have been released during challenges under a non-redistribute clause, or have been privately collected and mentioned in articles but never shared.

In order to address this limitation, in this work, we propose *ContentWise Impressions*, a new dataset of media recommendations containing impressions, that we release under a CC BY-NC-SA 4.0 license. We describe the data gathering process and provide statistics for the dataset comparing it to other commonly used datasets. We also provide further documentation and open-source tools to read the data, split it, and run several commonly used recommendation models.

[1] https://recsys.acm.org/challenges/
[2] https://www.kdd.org/kdd2020/kdd-cup
[3] http://www.wsdm-conference.org/2020/wsdm-cup-2020.php
[4] https://www.ijcai20.org/competitions.html

The rest of this work is organized as follows. In Section 2, we provide an overview of other datasets with impressions. In Section 3, we describe ContentWise Impressions. In Section 4, we explain the data gathering, preprocessing, and anonymization. In Section 5, we analyze the dataset and compare it with other datasets with impressions. In Section 6, we describe the experiments we performed and our observations from the evaluation procedures. Lastly, in Section 7, we provide final remarks and provide future lines of work.

2 IMPRESSIONS DATASETS

Impression datasets have been used by several articles [1, 2, 7, 13, 14, 20]. They can be classified into two categories: *private* datasets, collected by the authors of the article but, to the best of our knowledge, not made accessible to the community, and *non-redistributable* datasets, made accessible only to the participants of a challenge under a non-redistribute clause. In both cases, only a few researchers will have access to the dataset and will be able to use it. To the best of our knowledge, no open-source dataset with impressions exists.

2.1 Private datasets

Examples of private datasets are *LinkedIn PYMK Impressions* and *LinkedIn Skill Endorsement Impressions*. Both were used to model impressions discounting on large-scale Recommender System (RS)[14] and contain users registered on the LinkedIn[5] platform. More specifically, LinkedIn PYMK Impressions was used to recommend possible new user connections, and Linkedin Skill Endorsement was used to recommend skill endorsement of known users. Impressions in these datasets were present as a list of users, and a list of user-skill tuples, respectively [14].

Another example of a private dataset is the mobile apps impressions used in [7]. This dataset was gathered in order to develop a recommendation model for mobile applications on the Google Play Store in a low-latency scenario. In this dataset, impressions consist of mobile applications, application pages, historical statistics of the application, and more.

2.2 Non-redistributable datasets

The impression datasets that have been made available to challenge participants under a non-redistribute clause in recent years are several. Examples are those provided during the RecSys Challenges: 2016 by Xing [1, 16], 2017 by Xing [2], and 2019 by Trivago[2]. Those datasets, however, were only accessible to participants of the challenge and have not been made available to the wider research community.

Xing[6] is a social network for businesses, where users register to find jobs and recruiters register to find candidates. Users receive job recommendations. In Xing RecSys 2016, the impressions consist of the list of job recommendations provided to the user. In Xing RecSys 2017, the impressions were provided not as the recommendation list but rather as a boolean field to indicate if an item was shown to the user.

[5]https://www.linkedin.com/
[6]https://www.xing.com/

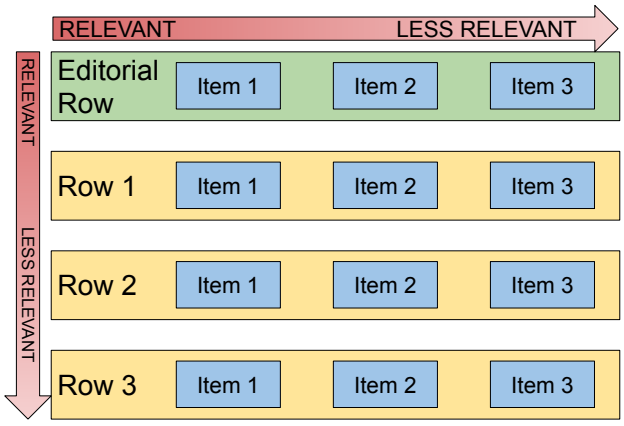

Figure 1: Example of the final user's screen layout. The *editorial* recommendations are generated by the OTT, and are not included in this dataset. In this example, ContentWise Impressions contains the impressions displayed in rows 1, 2, and 3. The number of rows and list lengths can vary. Most relevant rows were situated at the top of the screen. Most relevant items were situated to the left of the row.

Trivago[7] is a hotel search platform operating in several countries. In the Trivago RecSys 2019 dataset, users are provided with accommodation recommendations.

An older non-redistributable dataset is Tencent SearchAds Impressions [14, 20], which was available during the KDD Cup 2012 Track 2[8]. The dataset is a collection of interactions and impressions between users and the Tencent Search engine. The items in this dataset are represented by advertised results. The impressions comprise information about the user, the session, the query, the ads shown to the user, and their position on the screen.

3 DATA DESCRIPTION

In this section, we provide information about the data source and its content. The dataset is publicly available on Github[9].

3.1 Source

The data of ContentWise Impressions comes from an Over-The-Top Media service (OTT). This type of service offers media content to users via an Internet connection. In our case, the service offered content related to television and cinema. We collected the data for over four months between 2018 and 2019[10].

In Figure 1, we report the screen layout shown to the users in which each row of the grid represents a recommendation list. The recommendations contained in a row are all generated by the same algorithm, but different rows can be generated by different algorithms, including non-personalized ones. Moreover, further rows could have been added by the service provider in between the

[7]http://trivago.com/
[8]https://www.kaggle.com/c/kddcup2012-track2
[9]https://github.com/ContentWise/contentwise-impressions
[10]Due to technical difficulties, there are certain days where no data is present

rows we report in this dataset. Note that ContentWise Impressions only contains the impressions that were provided by ContentWise while it does not contain those directly provided by the OTT.

ContentWise Impressions is licensed under a CC BY-NC-SA 4.0 license[11]. Moreover, we emphasize that it is explicitly forbidden to de-anonymize this dataset in order to link any of the identifiers to the original data source.

3.2 Users

Users represent registered accounts for the OTT service. Due to the nature of these types of services, several physical users (e.g., family members, friends) and devices can use the same account to interact with the service. Each user is represented by an anonymized numerical identifier.

3.3 Items

Items represent the multimedia content that the service provided to the users and are represented by an anonymized numerical identifier. As we mentioned before, all items are inside the media domain. More specifically, they refer to television and cinema products.

Items belong to four mutually exclusive categories: *movies*, *movies and clips in series*, *TV movies or shows*, and *episodes of TV series*. These categories are encoded in the dataset with the values 0, 1, 2, and 3, respectively. All items are associated with a *series* identifier, which is used to group items belonging to the same series (i.e., TV series, movie series). Alongside this identifier, we also provide the *episode number* and *series length* for each item.

3.4 Interactions

The interactions represent the actions performed by users on items in the service and are associated with the timestamp[12] when it occurred. Interactions contain the identifier of the impressions, except in those cases where the recommendations came from a row added by the service provider. In Table 1, we provide a description of the interaction data. We categorized the interaction in four different types: *views*, *detail*, *ratings*, and *purchases*. These types are encoded in the dataset with the values 0, 1, 2, and 3, respectively.

3.4.1 Views. Views interactions indicate that the user watched an item of the service, and are represented with the interaction type *zero*. We also provide, in the *view factor* field, the point where the user stopped watching the item. The view factor is a real value that ranges from 0 to 1. If the user stopped watching near the end of the item, its view factor would be close to 1. On the other hand, if the user only viewed the beginning of the item, its view factor would be close to 0.

3.4.2 Details. Detail interactions indicate that the user accessed the item's detail page and are represented with the interaction type *one*.

3.4.3 Purchases. Some items need to be purchased before the user can watch them. Purchase interactions indicate that the user purchased an item of the catalog. We highlight that the catalog varied depending on the user's account subscription. Due to this, some users had to purchase items while others did not. The dataset does not contain any information about the user's account subscription.

3.4.4 Rating. Ratings are the only explicit feedback that the dataset contains, representing the rating value that a user gave to an item. Its values are in the range of 1-5 with a step of 0.5.

3.5 Impressions

The impressions refer to the recommended items that were presented to the user and are identified by their *series*. Impressions consist of a numerical identifier, the list position on the screen, the length of the recommendation list, and an ordered list of recommended series identifiers, where the most relevant item is in the first position. We provide two types of impressions:

3.5.1 Impressions with a direct link to interactions. The user interacted with at least one item in the recommendation list. We identify these impressions with a numerical identifier. In Table 2, we describe the content of these impressions.

3.5.2 Impressions without a direct link to interactions. The user did not interact with any of the items in the recommendation list at the time the list was provided. Note that the user may have interacted with any of those items by other means, e.g., by successive recommendations, or search. We identify these impressions with the identifier of the user who received the recommendations. In Table 3, we describe the content of these impressions.

To summarize, ContentWise Impressions is comprised of three different information layers. First, *interactions* of users with items of the service, containing user-item pairs. Second, *impressions with a direct link to interactions*, containing those recommendation lists that generated interactions. Third, *impressions without a direct link to interactions*, containing those recommendation lists that did not generate interactions.

3.6 Dataset format

We provide the dataset as three different splits: `interactions`, `impressions-direct-link`, and `impressions-non-direct-link`. These are stored using the Apache's Parquet format[13]. This format is open source, data is stored in columns, and parsers can read and write data faster than classic *CSV* parsers. There are several open-source tools for reading and writing Parquet files supporting several languages. We also include a human-readable *CSV* version of the dataset to ensure long term availability of this resource.

In Table 1, we provide the columns of the interactions and a description of them. All identifiers are anonymized, non-optional identifiers are always non-negative integers. Missing values are represented with -1. Similarly, in Table 2 and Table 3, we describe the columns for both impression sets. All row positions are non-negative integers, recommendation lengths are positive integers, and the recommendation list contains at least one recommendation.

4 DATASET BUILDING

In this section, we describe the process to build the data from its source, passing through the preprocessing, and anonymization of it.

[11] This license is available at https://creativecommons.org/licenses/by-nc-sa/4.0

[12] Specifically, the Coordinated Universal Time (UTC) UNIX timestamp, in milliseconds.

[13] https://parquet.apache.org/

Table 1: Columns and their description for the interactions

utc_ts_milliseconds	UTC Unix timestamp of the interaction
user_id	Numerical identifier of users
item_id	Numerical identifier of items
series_id	Numerical identifier of series
recommendation_id	Optional numerical identifier of the impression. If the impression is not present, then its value is −1
episode_number	Episode number of the item
series_length	Number of episodes of the series
item_type	Number to indicate the category of the item. Values range from 0 to 3
interaction_type	Number to indicate the type of the interaction. Values range from 0 to 3
explicit_rating	Rating value. If the interaction is not of type rating, then its value is −1
vision_factor	Vision factor value. If the interaction is not of type view, then its value is −1

Table 2: Columns and their description for the impressions with a direct link to interactions.

recommendation_id [a]	Numerical identifier of the impression.
row_position	Position on screen of recommendation.
recommendation_list_length	Number of recommended items
recommended_series_list [b]	Ordered recommendation list of series_id.

[a] This column is linked to the recommendation_id column present in Table 1
[b] The series are linked to the series_id column present in Table 1

Table 3: Columns and their description for the impressions without a direct link to interactions.

user_id [a]	Anonymized numerical identifier of the user that received the recommendation.
row_position	Position on screen of recommendation.
recommendation_list_length	Number of recommended items
recommended_series_list [b]	Ordered recommendation list of series_id.

[a] This column is linked to the user_id column present in Table 1
[b] The series are linked to the series_id column present in Table 1

4.1 Data acquisition

As mentioned in Section 3, ContentWise Impressions' data comes from an OTT service, i.e., on-demand media items, such as movies and TV series, which are streamed directly to the users via the Internet. We collected daily logs of interactions generated by the service and logs of recommendations made by our system.

4.2 Interactions preprocessing

As a first preprocessing step, we removed users and interactions that had missing values due to technical issues. We also removed users that did not have any view interaction.

When a user started watching an item, a view interaction was generated with a starting vision factor. When the user finished watching the same item, another view interaction was generated, indicating the final vision factor. A small percentage of users had incorrect view factors (e.g., always zero, no end view interaction) due to old software versions or technical issues. Users with invalid view factors have been removed.

Due to the significant size of the dataset, in order to make it suitable for research purposes, we built a subset of the original dataset via a uniform sampling of the users. The split we provide contains all interactions and impressions associated with the sampled users.

4.3 Impressions preprocessing

In this section, we describe the preprocessing of the impressions related to the interactions previously selected. Specifically, we grouped the impressions into the two disjoint sets described in Section 3.5: *Impressions with a direct link to interactions*, when the user interacted with at least one of the recommended items, and *Impressions without a direct link to interactions*, when the user did not interact with any recommendation. Lastly, initial impressions logs did not contain the recommendation list length; we calculated and included these values.

4.4 Data integrity

The additional material we provide with this dataset also contains several tests to ensure the data integrity and its correspondence with the description provided here. For the impressions, we ensured all were of valid types and had a value in the correct ranges, rows had at least one item, reported row length was the same as the actual row length. Lastly, we ensured that row positions were always non-negative. The complete list of the integrity checks on ContentWise Impressions is available in the online materials we provide.

4.5 Anonymization

A further preprocessing step involved the anonymization of all identifiers (i.e., users, items, and series). Each of the original identifiers has been replaced with a random unique, and anonymous numerical identifier. This step is meant to make it impossible to reconstruct the user identity or to find their original accounts. Again, note that de-anonymizing the data is expressly forbidden.

As mentioned in Section 3, the data has been collected over a period of four months between 2018 and 2019. The exact timestamps have been anonymized by applying a date and timezone shift. The day of the week has not been altered. After the date shift, the dataset contains timestamps from January, 7th 2019, to April, 15th 2019.

5 ANALYSIS AND DISCUSSION

In this section, we present an analysis of the ContentWise Impressions and compare it with other datasets containing impressions.

Table 4: Number of interactions grouped by their type.

Interaction Type	Count	Percentage
View	6, 122, 105	58.54%
Access	4, 105, 530	39.26%
Purchase	221, 066	2.11%
Rating	9, 109	0.09%
Total	10, 457, 810	100%

Table 5: Number of interactions grouped by the item type.

Item Type	Count	Percentage
Episodes of TV series	9, 076, 428	86.79%
Movies	987, 518	9.44%
TV Movies and shows	162, 574	1.56%
Movies and clips in series	231, 290	2.21%
Total	10, 457, 810	100%

Table 6: Number of items grouped by their type.

Item Type	Count	Percentage
Episodes of TV series	123, 831	85.36%
Movies	13, 733	9.47%
TV Movies and shows	5, 722	3.94%
Movies and clips in series	1, 788	1.23%
Total	145, 074	100%

5.1 Analysis of the dataset

ContentWise Impressions contains 10, 457, 810 interactions; 307, 453 impressions with direct links to interactions; and 23, 342, 617 impressions without direct link to interactions. The dataset also contains 42, 153 users; 145, 074 items and 28, 881 series.

In Table 4, we highlight the distribution of the interactions when grouped by interaction type, where 97.8% of the dataset is comprised of *view* and *access* interactions. Similarly, in Table 5, we present the distribution of interactions by item type, where 96.23% of the interactions correspond to *episodes of TV series* and *movies*. Lastly, in Table 6, we show the distribution of item types, where the same *episodes of TV series* and *movies* item types represent 94.83% of the total items.

We observed that users, items, and series, present long-tail distributions. For users, 27.96% most popular users are associated with 80% of the interactions. For items, 12.06% most popular items correspond with 80% of the interactions. For series, 4.05% most popular series appear in 80% of the interactions.

The average number of interactions per user is 248 (22 if counting direct interactions from impressions), where the maximum and the minimum number of interactions made by a single user are 13, 517 and 2 (2, 886 and 1 if counting direct interactions from impressions), respectively.

For items, the average number of interactions received per item is 72 (25 if counting interactions from impressions), where the maximum and the minimum number of interactions received by a single item are 23, 939 and 1 (6, 260 and 1 if counting interactions from impressions), respectively.

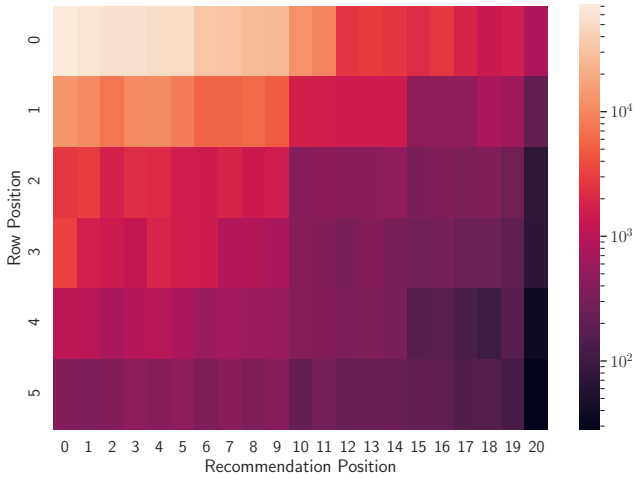

Figure 2: Heatmap of the number of interactions per position on the screen. Most interacted items are located in the first rows and on the first positions of the list. Values are log-scaled.

For impressions with direct links to interactions, the average number of interactions received per impression is 2, where the maximum and the minimum number of interactions received by a single item are 213 and 1, respectively.

In Figure 2, we show a heatmap that indicates the most interacted positions of the recommendation lists based on the row position on the screen. Specifically, we see that most interactions happen between the first three row positions, and the first ten item positions.

5.2 Comparison with other datasets

As previously mentioned in Section 2, currently, no impressions datasets are publicly available to the community. As such, we gathered and reported their statistics using the ones described on works that used those datasets.

To the best of our knowledge, ContentWise Impressions is the first dataset with impressions to be open-sourced. In previous years, other articles have used private datasets[7, 14], which were not released to the community. Others were disclosed under non-redistribution clauses on challenges[1, 2, 13, 20], where only a few researchers have access to them. Furthermore, ContentWise Impressions provides both impressions present in the interactions and without any associated interaction. Both LinkedIn PYMK Impressions and LinkedIn Skill Endorsement [14] also present both impressions. On the other hand, other datasets [1, 13] only provided impressions present in the interactions.

Another advantage of ContentWise Impressions is that it is subsampled in a way to be easily usable for research purposes without requiring significant computation resources. While researchers can indeed preprocess and subsample bigger datasets, if needed, this may result in different articles relying on different subsampling,

making it more difficult to compare research results and contributing to the reproducibility crisis in our field [9, 11]. For instance, Xing RecSys 2017[1, 2] contained around 1.5M users, 1.3M items, 322M interactions, and 314M impressions associated with these interactions. LinkedIn PYMK Impressions, LinkedIn Skill Endorsement Impressions, Tencent SearchAds Impressions [14] had 1.08, 0.19, and 0.15 billion impressions. For comparison, commonly used research datasets have a number of users and items in the range of tens of thousands and up to a few millions of interactions [11].

In Table 7, we provide a comparison of the density and Gini indexes of the datasets we could obtain[14]. The Gini Index is computed on the number of interactions associated with each user or on those associated with each item. As a reference, we also provide these values for the more-commonly used MovieLens 20M dataset[12]. From Table 7, we can see that ContentWise Impressions is significantly denser than other impression datasets, while sparser than Movielens. In terms of the Gini Index, higher values indicate the dataset is more biased towards popular items or users with long profiles. ContentWise Impressions exhibits a significantly lower popularity bias on both the items and the users, indicating that the data is more balanced.

Table 7: Comparison of ContentWise Impressions with others datasets based on their density, Gini indexes on item popularity and users.

Dataset	Density	Gini Items	Gini Users
ContentWise Impressions	$7.4 \cdot 10^{-4}$	0.3345	0.3316
Xing Recsys 2016	$6.4 \cdot 10^{-6}$	0.6890	0.7652
Xing Recsys 2017	$6.6 \cdot 10^{-6}$	0.6530	0.9241
Movielens 20M	$5.3 \cdot 10^{-3}$	0.5807	0.9048

6 EXPERIMENTS

The purpose of the experiments is both to report baseline results for the ContentWise Impressions as well as provide examples in the online materials that researchers can use and refer to. In this section, we describe the experiments we performed on the dataset.

We provide open-source materials written in Python to download the dataset, install the environment, read the data, parse it, and use several common recommendation models. The source code is available on Github[15]. The source code relies on common open-sourced scientific libraries. We ran the experiments on a single Linux Amazon EC2 r4.4xlarge instance. At the time of writing, this type of instance provides 16 vCPU and 128 GiB of RAM.

6.1 Recommendation task

We evaluated the models under a traditional *top-k* recommendation task with only collaborative information, i.e., user-item interactions and impressions. We rely on the publicly available evaluation framework[16] developed by Ferrari Dacrema et al. [9, 11]. We added a few changes to the framework in order to support the parallel evaluation of users and utility methods to extract, transform, and load ContentWise Impressions. We also included consistency checks of the dataset using unit tests.

The data is split via random holdout of the interactions in training (70%), validation (10%) and test (20%). All interactions are considered as implicit with a value of 1.

6.2 Baseline algorithms

We report the recommendation quality of several simple algorithms. As non-personalized baselines, we report a *Top Popular* recommender, which recommends the items having the highest number of interactions. As the personalized recommenders, we report *ItemKNN*, a simple neighborhood-based recommender using various similarities measures: cosine[4], dice[10, 18], jaccard[4], asymmetric[3], and tversky[19]. We also report a graph-based algorithm RP^3_β proposed in [15]. For latent-factor methods, we report *PureSVD* [8] and *MF BPR* [17].

In order to provide a simple example of how to use the impressions during the training phase of a model, we adapted the MF BPR algorithm. Traditional BPR requires to sample for each user, a positive interaction (i.e., an item the user interacted with), and a negative interaction (i.e., an item the user did not interact with). Specifically, we did not alter the positive sampling, but we experimented with three different strategies for the negative sampling: *uniform-at-random*, sampling uniformly among the items the user did not interact with; *uniform-inside-impressions*, sampling uniformly among the user impressions; and *uniform-outside-impressions*, sampling uniformly among the items not in the impressions. Items the user interacted with are never sampled as negatives.

6.2.1 Hyperparameter tuning. We tuned the hyperparameters of each recommendation algorithm, optimizing the recommendation quality on the validation data. We applied Bayesian Optimization[5, 6] and set the hyperparameter ranges and distributions according to those used in [9, 11]. When the Bayesian search ended, we trained the algorithm using the best hyperparameter found on the union of train and validation data and report the results obtained on the test data.

6.2.2 Evaluation. We measured the performance of the recommendation techniques using both accuracy and beyond-accuracy metrics at recommendation list lengths 20. We report results of Precision, Mean Average Precision (MAP), Normalized Discounted Cumulative Gain (NDCG), and Item Coverage (Item Cov), which represents the quota of items that were recommended at least once[17].

6.3 Experiments result

In Table 8, we report the results of the evaluation. We can observe that the best performing algorithm is ItemKNN, in particular with the tversky similarity. Other algorithms, like RP^3_β and PureSVD, have a lower recommendation quality. The recommendation quality of the simple MF BPR baseline is relatively low, achieving a similar recommendation quality as the Top Popular baseline. This suggests

[14] We could not acquire any of the private datasets and therefore could not compute those additional statistics.
[15] https://github.com/ContentWise/contentwise-impressions
[16] https://github.com/MaurizioFD/RecSys2019_DeepLearning_Evaluation
[17] We exported results with more metrics in the repository.

the need for further studies to develop a more suitable algorithmic solution.

When comparing the MF BPR negative items sampling strategies, we can observe that the recommendation quality overall does not change dramatically but shows a tendency to decrease in both cases when impressions are used. The recommendation quality decreases the most when negative items are sampled within the impressions. This behavior is expected for two reasons. First, the impressions are the recommendations that were provided to a user by another recommendation model. Therefore, they are unlikely to contain strongly negative items. Sampling negative items among impressions will result in considering as negatives those items that are close to the interests of the user, therefore steering the algorithm in the wrong direction. Second, sampling only outside of the impressions is, too, a limited strategy, as erroneous recommendations will not be sampled as negatives and will prevent the algorithm to further refine its quality. Both these results indicate that a more articulate sampling strategy can be developed, potentially merging the strengths of the two, while minimizing their weaknesses.

As another interesting observation, we can see that the Item Coverage of the MF BPR algorithm is much better than the Top Popular one, indicating that despite its similar recommendation quality, the MF BPR allows for a far greater exploration of the catalog. In this case, we can see a significant difference between negative sampling strategies. Sampling negatives within the impressions results in a markedly low item coverage, whereas sampling outside the impressions allows the model to improve the item coverage over the plain uniform negative sampling.

Table 8: Evaluation of different metrics on recommendation lists of length 20. Best results highlighted in bold.

	PREC	MAP	NDCG	Cov. Item
TopPop	0.0225	0.0387	0.0619	0.0006
ItemKNN CF cosine	0.2562	0.3972	0.4907	0.3431
ItemKNN CF dice	0.2565	0.3952	0.4878	0.3887
ItemKNN CF jaccard	0.2574	0.3979	0.4910	0.4203
ItemKNN CF asymmetric	0.2549	0.3949	0.4896	0.3225
ItemKNN CF tversky	**0.2587**	**0.4010**	**0.4935**	0.3791
RP3beta	0.1687	0.2641	0.3664	0.3502
MF BPR	0.0314	0.0531	0.0900	0.1012
MF BPR inside	0.0205	0.0323	0.0550	0.0006
MF BPR outside	0.0195	0.0395	0.0619	0.1202
PureSVD	0.1730	0.2416	0.3369	0.0897

7 CONCLUSION AND FUTURE WORKS

In this work, we presented ContentWise Impressions, a novel dataset with impressions, gathered from an industrial service provider which, to the best of our knowledge, is the first one to be publicly available to the research community. The dataset is licensed under a CC BY-NC-SA 4.0 license, allowing its wide usage for both academic and industry research.

We described the contents of the dataset, from its users, items, interactions, impressions, and format. We also documented how we built it, going from its source, preprocessing, and finally, its anonymization. We analyzed the dataset, compared it against other datasets, and presented the results of our experiments. In these, we observed how the use of impressions affects the performance of some state-of-the-art recommendation techniques. We open-sourced all the tools and documentation that we used so others can reproduce our observations. Moreover, inside these tools, we provided instructions to download, load, and use the dataset.

ContentWise Impressions can enable the community to further study how to embed the impression information in algorithmic solutions for recommendations. Possible research directions are, for example, refining the user model according to how many times they did not interact with a recommended item, when to stop to recommend an item to a user, reranking strategies to compensate known errors that the recommendation model generating the impressions has been found to make. Another possibility is to post-process the recommendations in order to mitigate biases that the impressions may exhibit. Lastly, if met with interest from the community, updated and bigger versions of this dataset can be released in the future.

REFERENCES

[1] Fabian Abel, András Benczúr, Daniel Kohlsdorf, Martha Larson, and Róbert Pálovics. 2016. RecSys Challenge 2016: Job Recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems* (Boston, Massachusetts, USA) *(RecSys '16)*. Association for Computing Machinery, New York, NY, USA, 425–426. https://doi.org/10.1145/2959100.2959207

[2] Fabian Abel, Yashar Deldjoo, Mehdi Elahi, and Daniel Kohlsdorf. 2017. RecSys Challenge 2017: Offline and Online Evaluation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems* (Como, Italy) *(RecSys '17)*. Association for Computing Machinery, New York, NY, USA, 372–373. https://doi.org/10.1145/3109859.3109954

[3] Fabio Aiolli. 2013. Efficient Top-n Recommendation for Very Large Scale Binary Rated Datasets. In *Proceedings of the 7th ACM Conference on Recommender Systems* (Hong Kong, China) *(RecSys '13)*. Association for Computing Machinery, New York, NY, USA, 273–280. https://doi.org/10.1145/2507157.2507189

[4] Xavier Amatriain and Josep M. Pujol. 2015. *Data Mining Methods for Recommender Systems*. Springer US, Boston, MA, 227–262. https://doi.org/10.1007/978-1-4899-7637-6_7

[5] Sebastiano Antenucci, Simone Boglio, Emanuele Chioso, Ervin Dervishaj, Shuwen Kang, Tommaso Scarlatti, and Maurizio Ferrari Dacrema. 2018. Artist-driven Layering and User's Behaviour Impact on Recommendations in a Playlist Continuation Scenario. In *Recommender Systems Challenge Workshop at the 12th ACM Conference on Recommender Systems*. 4:1–4:6.

[6] Eric Brochu, Vlad M. Cora, and Nando de Freitas. 2010. A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning. *CoRR* abs/1012.2599 (2010). arXiv:1012.2599 http://arxiv.org/abs/1012.2599

[7] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide & Deep Learning for Recommender Systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems* (Boston, MA, USA) *(DLRS 2016)*. Association for Computing Machinery, New York, NY, USA, 7–10. https://doi.org/10.1145/2988450.2988454

[8] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of Recommender Algorithms on Top-n Recommendation Tasks. In *Proceedings of the Fourth ACM Conference on Recommender Systems* (Barcelona, Spain) *(RecSys '10)*. Association for Computing Machinery, New York, NY, USA, 39–46. https://doi.org/10.1145/1864708.1864721

[9] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. 2019. Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches. In *Proceedings of the 13th ACM Conference on Recommender Systems* (Copenhagen, Denmark) *(RecSys '19)*. Association for Computing Machinery, New York, NY, USA, 101–109. https://doi.org/10.1145/3298689.3347058

[10] Lee R. Dice. 1945. Measures of the Amount of Ecologic Association Between Species. *Ecology* 26, 3 (1945), 297–302. https://doi.org/10.2307/1932409

arXiv:https://esajournals.onlinelibrary.wiley.com/doi/pdf/10.2307/1932409

[11] Maurizio Ferrari Dacrema, Simone Boglio, Paolo Cremonesi, and Dietmar Jannach. 2019. A Troubling Analysis of Reproducibility and Progress in Recommender Systems Research. *arXiv:1911.07698* (2019).

[12] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4, Article 19 (Dec. 2015), 19 pages. https://doi.org/10.1145/2827872

[13] Peter Knees, Yashar Deldjoo, Farshad Bakhshandegan Moghaddam, Jens Adamczak, Gerard-Paul Leyson, and Philipp Monreal. 2019. RecSys Challenge 2019: Session-Based Hotel Recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems* (Copenhagen, Denmark) *(RecSys '19)*. Association for Computing Machinery, New York, NY, USA, 570–571. https://doi.org/10.1145/3298689.3346974

[14] Pei Lee, Laks V.S. Lakshmanan, Mitul Tiwari, and Sam Shah. 2014. Modeling Impression Discounting in Large-Scale Recommender Systems. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, New York, USA) *(KDD '14)*. Association for Computing Machinery, New York, NY, USA, 1837–1846. https://doi.org/10.1145/2623330.2623356

[15] Bibek Paudel, Fabian Christoffel, Chris Newell, and Abraham Bernstein. 2016. Updatable, Accurate, Diverse, and Scalable Recommendations for Interactive Applications. *ACM Trans. Interact. Intell. Syst.* 7, 1, Article 1 (Dec. 2016), 34 pages. https://doi.org/10.1145/2955101

[16] Mirko Polato and Fabio Aiolli. 2016. A Preliminary Study on a Recommender System for the Job Recommendation Challenge. In *Proceedings of the Recommender Systems Challenge* (Boston, Massachusetts, USA) *(RecSys Challenge '16)*. Association for Computing Machinery, New York, NY, USA, Article 1, 4 pages. https://doi.org/10.1145/2987538.2987549

[17] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence* (Montreal, Quebec, Canada) *(UAI '09)*. AUAI Press, Arlington, Virginia, USA, 452–461.

[18] T.J. Sørensen. 1948. *A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and Its Application to Analyses of the Vegetation on Danish Commons*. I kommission hos E. Munksgaard. https://books.google.it/books?id=rpS8GAAACAAJ

[19] Amos Tversky. 1977. Features of similarity. *Psychological review* 84, 4 (1977), 327.

[20] Wentao Wang and Dongzhi He. 2018. Click-through Rate Estimates Based on Deep Learning. In *Proceedings of the 2018 2nd International Conference on Deep Learning Technologies* (Chongqing, China) *(ICDLT '18)*. Association for Computing Machinery, New York, NY, USA, 12–15. https://doi.org/10.1145/3234804.3234811