PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA

ALGORITHMS FOR VISUAL ART RECOMMENDATION: LEVERAGING VISUAL FEATURES, METADATA AND IMPLICIT FEEDBACK

PABLO MESSINA

Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

Advisor:
DENIS PARRA SANTANDER

Santiago de Chile, January 2019

PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA

ALGORITHMS FOR VISUAL ART RECOMMENDATION: LEVERAGING VISUAL FEATURES, METADATA AND IMPLICIT FEEDBACK

PABLO MESSINA

Members of the Committee:
DENIS PARRA SANTANDER
ALVARO SOTO ARRIAZA
EDUARDO GRAELLS GARRIDO
GONZALO YAÑEZ CARRIZO

Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

Santiago de Chile, January 2019

Gratefully to my parents and siblings

ACKNOWLEDGEMENTS

I would like to thank my advisor Denis Parra for his support and guidance throughout these two and half years of Master's degree. I would also like to give thanks to my fellow Master's students Vicente Domínguez and Felipe del Río, as well as research collaborators Christoph Trattner and Alvaro Soto, for the fruitful times of collaboration I shared with them at different stages of my research. I am also grateful to the members of my committee for their patience and valuable feedback in the elaboration of this thesis. Finally, I would like to thank my family: my parents, my brother and my sister for supporting me morally throughout writing this thesis as well as throughout my life in general.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Recommender Systems help us deal with information overload by suggesting relevant items based on our personal preferences. Although there is a large body of research in areas such as movies or music, artwork recommendation has received comparatively little attention, despite the continuous growth of the artwork market. Most previous research has relied on ratings and metadata, and a few recent works have exploited visual features extracted with convolutional neural networks (CNN) to recommend digital art. In this work, we contribute to the area of *one-of-a-kind* physical paintings recommendation by studying several recommendation algorithms, based on different sources of information: artwork metadata, handcrafted visual features, neural visual features and collaborative information from users' implicit feedback. We implement and evaluate our algorithms using transactional data from *UGallery.com*, an online artwork store. Furthermore, we propose a novel neural network model for the task of artwork recommendation that combines content and collaborative information. We name this network YT-VBPR since it is inspired by ideas from Youtube's deep learning recommender system and VBPR (a state-of-the-art visually-aware recommendation method). Our results show that among all methods tested, YT-VBPR achieves the best results. Furthermore, once trained, YT-VBPR only needs images as input to be able to recommend, allowing easy generalization to new users and items without further training. Our research can provide valuable insights to researchers and developers in the artwork recommendation domain in particular, and also to those interested in visually-aware recommendation methods in general.

Keywords: Artwork, Recommender systems, Content-based Recommendation, Collaborative Filtering, Hybrid Recommendations, Metadata, Visual Features, Implicit Feedback, Deep Neural Networks.

RESUMEN

Los Sistemas Recomendadores nos ayudan a lidiar con la sobrecarga de información mediante la sugerencia de ítems relevantes conforme a nuestras preferencias. Si bien hay una gran cantidad de investigación en áreas como películas o música, la recomendación de obras de arte ha recibido comparativamente poca atención, a pesar del continuo crecimiento del mercado de arte. La mayoría de la investigación previa ha dependido de ratings y metadatos, y unos pocos trabajos recientes han aprovechado descriptores visuales extraídos con redes neuronales convolucionales (CNN) para recomendar arte digital. En este trabajo, contribuimos al área de recomendación de pinturas físicas originales mediante el estudio de algoritmos de recomendación basados en diferentes fuentes de información: metadatos, descriptores visuales hechos a mano, descriptores visuales neuronales e información colaborativa de la retroalimentación implícita de los usuarios. Implementamos y evaluamos nuestros algoritmos usando datos transaccionales de *UGallery.com*, una tienda de arte en línea. Además, proponemos un modelo de red neuronal novedoso para la tarea de recomendación de arte que combina contenido e información colaborativa. Lo llamamos YT-VBPR ya que está inspirado en ideas del sistema de recomendación de aprendizaje profundo de Youtube y VBPR (un método de recomendación estado del arte que incorpora información visual). Nuestros resultados muestran que entre todos los métodos probados, YT-VBPR alcanza los mejores resultados. Además, una vez entrenado, YT-VBPR sólo necesita imágenes como entrada para recomendar, permitiendo generalizar fácilmente a nuevos usuarios e ítems sin entrenamiento adicional. Nuestra investigación puede proveer observaciones valiosas a investigadores y desarrolladores en el dominio de recomendación de arte en particular, y también a aquellos interesados en métodos de recomendación con contenido visual en general

Palabras claves: Obras de arte, Sistemas recomendadores, Recomendación basada en contenido, Filtrado colaborativo, Recomendaciones híbridas, Metadatos, Descriptores visuales, Retroalimentación implícita, Redes neuronales profundas.

1. GLOSSARY

- **Collaborative Filtering**: Collaborative filtering (CF) refers to the technique of exploiting interaction patterns between users and items to train/fit a recommendation model, without taking into consideration domain-specific features of users or items. In other words, without knowing any information about users or items beyond who interacted with what (e.g. views, likes, purchases, etc.), collaborative filtering analyzes these interactions in order to fit a recommender model. That's why CF in its purest form is agnostic to the domain and can be applied to any domain in which there are users, items and interactions between them (e.g. movies, videos, music, images, food, etc.). However, there are more sophisticated variants that can take advantage of domain-specific features and information to achieve better results.
- **Content-based Filtering**: Content-based filtering refers to the technique of exploiting features about the contents of items and/or users and the local user-item interaction history of each user to train/fit a recommendation model. Unlike collaborative filtering, content-based filtering takes advantage of the domain-specific features of items and users available, but lacks the ability of inferring patterns from user-item interactions across multiple users like collaborative filtering does. However, there are more sophisticated variants that can employ both content-based and collaborative filtering at the same time.
- **Convolutional Neural Network (CNN)**: A CNN is a specific type of neural network originally designed for processing images, although they can be and have been applied in other domains. The main idea is to have multiple layers, each layer comprising multiple convolutional filters. A convolutional filter is essentially a linear regression which is applied over a window or rectangular area of the image, followed by a nonlinear activation function in order to output a value (a real number). The filter is applied over the image multiple times as a sliding window, generating a grid of values from the image also known

as *activation map*. The activation maps of multiple filters are stacked together forming a single layer. Thus, each layer becomes the input for the next layer, until a final layer has a high level representation of the image and can be used for any downstream task, such as object detection, scene identification, etc.

- **Embedding**: An embedding is a function that maps vectors from a high-dimensional space into a lower-dimensional space, such that the embedded vectors hold properties which are relevant in some way in the original space. Embeddings make it easier to do machine learning on large inputs like sparse vectors representing words or RGB vectors representing images. Ideally, an embedding captures some of the semantics of the input by placing semantically similar inputs close together in the embedding space. An embedding can be learned and reused across models.
- **Handcrafted Visual Features**: By handcrafted or manually-engineered visual features we mean any kind of features calculated from an image following an algorithm in which all the steps were carefully designed by a human, and therefore are explainable. This contrasts with the machine learning approach, in which the algorithm for feature extraction is learned automatically by a machine from data, and the human only intervenes indirectly by providing the data and the supervision signals that guide the learning process.
- **Implicit Feedback**: In the context of Recommender Systems, implicit feedback refers to any kind of behavioral traces left by users of a system from which users' interests can be inferred. Examples of implicit feedback are purchases, watches, reproductions, time spent watching or listening to video, audio, etc. Compared to explicit feedback (e.g. ratings, likes, stars), implicit feedback has the advantage of being much more abundant and easier to collect, at the expense of being more prone to noise (watching a movie does not necessarily mean the user likes it, whereas an explicit "like" or "thumb up" leaves no room for doubts).
- **Matrix Factorization**: In the context of Recommender Systems, Matrix Factorization is the technique of thinking of users and items as rows and columns

respectively of a matrix, in which each cell stores a value that indicates the degree of preference a user has for an item, and then approximating this matrix as a product of two lower-dimensional matrices: a low-dimensional user matrix and a low-dimensional item matrix. Matrix Factorization is an example of Collaborative Filtering, since the technique only works on the interactions and does not assume anything about the particular domain. There are many variants in the literature that has been inspired by this simple idea.

- **Neural Network**: (Artificial) Neural Networks (NN) are a class of computational models within machine learning inspired by the structure and functions of biological neural networks. A neural network is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal from one neuron to another. Neural Networks are known for being general function approximators, which is why they can be applied to almost any machine learning problem about learning a complex mapping from the input to the output space.
- **Recommender System**: A recommender system is a subclass of information filtering system that seeks to predict the "rating" or "preference" a user would give to an item. They are primarily used in commercial applications. Recommender systems are utilized in a variety of areas, and are most commonly recognized as playlist generators for video and music services like Netflix, YouTube and Spotify, product recommenders for services such as Amazon, or content recommenders for social media platforms such as Facebook and Twitter.

2. INTRODUCTION

Despite the financial crisis of 2007-2008 which shook the markets worldwide, the global artwork market has kept growing over the years. For instance, in 2011, art received $11.57 billion in total global annual revenue, over $2 billion versus 2010 (Esman, 2012). Particularly, online artwork sales are booming mostly due to the influence of social media and new consumption behavior of millennials (Weinswig, 2016). Online art sales reached $3.27 billion in 2015, and at the current growth rate, it will reach $9.58 billion by 2020. Notably, although many online businesses utilize recommendation systems to boost their revenue, online artwork recommendation has received little attention from the research community compared to other areas such as movies (Amatriain, 2013; Gomez-Uribe & Hunt, 2016) or music (Maes et al., 1994; Celma, 2010), probably because artworks are not as popular and as frequently consumed as these aforementioned consumer goods.

There are several stores nowadays that sell artworks online, such as UGallery[1], Singulart[2], and Artspace[3]. However, finding the right artwork for people's personal taste is a tricky task, as several properties need to be considered. Recommender systems could indeed help in this task, since previous research have been tailored explicitly towards helping people find relevant artworks, specially in the context of museum collections (Aroyo et al., 2007; Albanese, d'Acierno, Moscato, Persia, & Picariello, 2011; Semeraro, Lops, De Gemmis, Musto, & Narducci, 2012). Most of these works have dealt with recommendation in museum collections using traditional methods and data such as ratings, textual descriptions and social tags (Aroyo et al., 2007; Albanese et al., 2011; Semeraro et al., 2012). The earliest of these works was the CHIP project (Aroyo et al., 2007), which implemented well-known techniques such as content-based and collaborative filtering for artwork recommendation in the Rijksmuseum. More recently, He et al. (2016) used pre-trained deep convolutional neural networks (CNN), combined with collaborative information, for the recommendation of digital art online. This is a very promising technique,

[1] www.ugallery.com
[2] www.singulart.com
[3] www.artspace.com

since the development of deep neural networks has increased by orders of magnitude the performance on visual tasks such as image classification (Krizhevsky, Sutskever, & Hinton, 2012) or scene identification (Sharif Razavian, Azizpour, Sullivan, & Carlsson, 2014). However, He et al. (2016) only studied digital art rather than physical artifacts such as paintings or sculptures, which is what most of the aforementioned online art stores sell.

Unlike the aforementioned works (Aroyo et al., 2007; Albanese et al., 2011; Semeraro et al., 2012; R. He et al., 2016), in this article we address the problem of artwork recommendation for *one-of-a-kind* paintings in online art stores. We call a painting *one-of-a-kind* when only one instance is available, thus running out of stock with the first purchase. This means that if the only user feedback available is purchases, then 1) it is not possible to find positive co-occurrences on the same items and 2) all items still in stock that could be recommended have not been purchased by anyone yet, therefore there is no direct user feedback for any of those items either. This situation limits the applicability of many collaborative filtering methods, which usually rely on overlapping interaction patterns between users and items. However, the availability of information about items' content (e.g. images and metadata) gives the opportunity for exploring content-based methods, which do not depend on collaborative information. Moreover, by taking advantage of item's content, it is still possible to apply collaborative filtering in the *one-of-a-kind* setting: if we think of items as vectors in a content embedding space, then co-ocurrences can still happen, i.e. if two users have similar tastes, it's reasonable to expect similar purchase patterns in the content embedding space. For this reason, we address this problem with two general approaches: 1) as a purely content-based recommendation problem and 2) as a hybrid content-collaborative recommendation problem. With respect to content-based methods, we focus on different types of content –including metadata, automatically learned features from deep convolutional neural networks (CNN) as well as manually-engineered / handcrafted visual features (HVF)– and also on how to combine them for personalized recommendation. With respect to combining content and collaborative information, we propose a new deep neural network that leverages the visual content of items to map both

users and items into a latent space, with a training scheme that combines item's content with the implicit feedback from users' purchase history.

Outline. In this thesis, our work is divided into two parts. In the first part, we study the impact of different features for content-based recommender systems of physical artworks. In particular, we investigate the utility of artwork metadata (curated attributes and artist), neural (CNN) and handcrafted (HVF) visual features extracted from images as well as user transactions from the online store *UGallery*. We perform two evaluations: an offline evaluation with a dataset provided by *UGallery*, and then a small online study with 8 *UGallery* expert curators. In the second part, we propose and evaluate a new neural network we call YT-VBPR, a hybrid content-collaborative recommendation model which combines domain-specific insights from the first part with ideas from Youtube's recommender system (Covington, Adams, & Sargin, 2016) and VBPR (R. He & McAuley, 2016).

Research Questions. To drive our research five questions were defined. They are as follows:

- *RQ1*. To what extent is it possible to predict people's purchases based on content features? Since we have several types of content features, we answer this question by splitting the analysis into two subgroups:
  - *RQ1.1* Which is the best metadata-based feature?
  - *RQ1.2* Which is the best visual feature?
- *RQ2*. How do different sets of features (metadata vs. visual) compare in the artwork recommendation domain? Although both feature sets could potentially be useful, curated metadata is not always available. Visual features, which can be calculated for every image, have then the potential to alleviate the new item problem.
- *RQ3*. Is there an optimal way of combining features with hybrid methods to maximize the recommendation performance?
- *RQ4*. To what extent is an offline evaluation consistent with an expert user validation?

- *RQ5*. Is it possible to develop a model that combines content and collaborative information in order to achieve better performance than content-based baselines?

Contributions. (1) In general, the work outlined in this article makes a contribution to the yet sparsely explored problem of recommending physical artworks to people online. To make this happen, we study and compare the utility of several sources of information (content metadata, visual features and purchase records), typically available in online galleries. We do this by running an extensive set of simulated experiments with real-world data provided by a large online artwork store based in CA, USA called *UGallery*. (2) Furthermore, our work contributes to the *one-of-a-kind* recommender system problem – i.e., items that run out of stock with the first purchase – by leveraging items' content. We evaluate a wide range of content-based methods, including methods that use individual features and hybrid methods that combine multiple features simultaneously. (3) We conduct an online evaluation with UGallery expert curators which confirms the benefits of combining multiple content features as observed in the offline results. And (4) we propose a new neural network model that combines content features with collaborative information, outperforming all other methods we tested. To the best of our knowledge, we are 1) the first to make an exhaustive study of multiple visual (CNN and handcrafted) and metadata (artist and curated attributes) features for physical artwork recommendation, and 2) the first to propose a novel neural network for artwork recommendation with a fixed set of parameters that can generalize to new users and items without further training by operating entirely on visual content, by mapping artwork images to vectors and by aggregating the visual features of a user's purchased artworks to produce a user vector, and furthermore, with a training protocol that incorporates both content and collaborative information at training time.

3. RELATED WORK

In this section we provide an overview of relevant related work. The section is split into two parts: *Artwork Recommender Systems* (3.1) and *Visually-aware Recommender Systems* (3.2). Both sub-sections are important to better understand our contribution and the problem we are targeting in this thesis. A final Section *Differences to Previous Research* (3.3) highlights what we add with our work to the already existing literature in the area.

3.1. Artwork Recommender Systems

In the context of artwork recommender systems, one of the first contributions was made by the CHIP Project (Aroyo et al., 2007). The aim of the project was to build a recommendation infrastructure for the Rijksmuseum in the Netherlands. The project used several techniques such as content-based filtering based on metadata provided by experts, as well as collaborative filtering based on users' ratings given to artworks of the Rijksmuseum.

Another important contribution in the field is the work developed by Semeraro et al. (Semeraro et al., 2012). In their paper, they introduce an artwork recommender system called FIRSt (Folksonomy-based Item Recommender syStem) which utilizes social tags given by experts and non-experts over 65 paintings of the Vatican picture gallery. They focused their research on making recommendations using textual features (textual painting descriptions and user tags), but did not employ visual features among their methods.

More complex methods were implemented recently by Benouaret Lenne (2015), who improve the current state-of-the-art in artwork recommender systems using context obtained through a mobile application. The particular research question they address is to what extent it is possible to make museum tour recommendations more useful. They propose a hybrid recommender that combines multiple signals: hierarchical and non-hierarchical artwork metadata (e.g. artist, style, age) obtained from external knowledge

bases (ontologies and thesauruses), users' demographic (e.g. sex, age) and contextual (e.g. time, location) information as well as ratings given by users to artworks during tours using the mobile app.

Finally, the recent work of He et al. addresses digital artwork recommendations based on pre-trained deep neural visual features (R. He et al., 2016). In this case, the experiments were conducted on a virtual art gallery, with the advantage of items always available and explicit user feedback in the form of ratings.

3.2. Visually-aware Recommender Systems

Manually-engineered visual features extracted from images (texture, sharpness, brightness, etc.) have been used in several tasks for information filtering, such as retrieval (Rui, Huang, Ortega, & Mehrotra, 1998; La Cascia, Sethi, & Sclaroff, 1998) and ranking (San Pedro & Siersdorfer, 2009). More recently, very promising results have been shown for the use of low-level handcrafted stylistic visual features automatically extracted from video frames for content-based video recommendation (Deldjoo et al., 2016). By extracting and aggregating 5 stylistic visual features per video and using cosine similarity for pairwise comparison, Deldjoo et al. achieved higher recommendation accuracy than traditional recommendation methods based on high-level expert annotated metadata. Even better results are obtained when both stylistic visual features and annotated metadata are combined in a hybrid recommender, as shown in the work of Elahi et al. (2017).

In the latest years, many works in image processing and computer vision such as object recognition (Akçay, Kundegorski, Devereux, & Breckon, 2016), image classification (Krizhevsky et al., 2012) and scene identification (Sharif Razavian et al., 2014) have shown significant performance improvements by using visual embeddings obtained from pre-trained deep convolutional neural networks (Deep CNN) such as AlexNet (Krizhevsky et al., 2012), VGG (Simonyan & Zisserman, 2014) and ResNet (K. He, Zhang, Ren, & Sun, 2016). These are examples of transfer learning methods, i.e., visual embeddings

trained for specific tasks (e.g. image classification) which perform well in other tasks (e.g. image segmentation) and have been adopted for the recommendation problem.

Motivated by these results, MacAuley et al. (2015) introduced an image-based recommendation system based on styles and substitutes for clothing using visual embeddings pre-trained on a large-scale dataset obtained from Amazon.com. Recently, He et al. (R. He & McAuley, 2016) went further in this line of research and introduced a visually-aware matrix factorization approach that incorporates visual signals (from a pre-trained CNN) into predictors of people's opinions. Their training model is based on Bayesian Personalized Ranking (BPR), a model previously introduced by Rendle et al. (Rendle, Freudenthaler, Gantner, & Schmidt-Thieme, 2009).

The latest work by He et al. (2016) deals with visually-aware artistic recommendation, building a model which combines implicit (*clicks*) and explicit (*appreciates*) feedback, visual features, social dynamics (preference for certain artists) and temporal dynamics (short/long term preferences). Another relevant work was the research by Lei et al. (2016) who introduced comparative deep learning for hybrid image recommendation. In this work, they use a neural network architecture for making recommendations of images using user information (such as demographics and social tags) as well as images in pairs (one liked, one disliked) in order to build a ranking model.

3.3. Differences to Previous Research

Almost all the surveyed articles on artwork recommendation have in common that they used standard collaborative techniques (exploting user-item co-ocurrence patterns) and content-based techniques (using textual metadata), but without exploiting visual features extracted from images.

For example, the work by Benouaret Lenne (2015) addresses the artwork recommendation problem in museums, yet their solution cannot be fully applied to the *one-of-a-kind* problem in online stores as we approach it in this research: in our setting most artists are

new emergent artists who create novel artworks, so there are no ontologies or thesauruses containing relevant information about them like what one could easily find about e.g. da Vinci or Picasso. In our setting *one-of-a-kind* artworks are not usually rated either and they can only be purchased once, so similarities like the pearson correlation used by these authors are not applicable either.

In terms of content-based filtering, unlike the previous works we extract, compare and combine metadata, neural visual features and handcrafted visual features. In terms of collaborative filtering, as mentioned before our problem setting is not well suited for a direct application of traditional collaborative filtering methods due to the *one-of-a-kind* condition of physical artworks. Instead, we use the collaborative information implicit in users' purchases as part of the training signals of a general content-based deep neural network, so everything learned by the network can be easily generalized to new users and items based on visual content without further training.

With regards to the related work on visually-aware recommender systems, almost all of the surveyed articles have focused on tasks different from artwork recommendation, such as video recommendation and clothing recommendation. Nonetheless, there are certain ideas from these works which are still applicable to some extent to the *one-of-a-kind* artwork domain. For instance, the work by MacAuley et al. (2015) on clothing recommendation relies heavily on items co-occurrence patterns to generate the ground truth labels for their loss function, which is not feasible in a *one-of-a-kind* setting. However, their approach has valuable properties, such as using collaborative information during training to learn an image embedding and only depending on images (visual content) during evaluation. The work by He et al. (2016) enhances BPR (Rendle et al., 2009) with visual information from items' images, which alleviates the item cold-start problem in the clothing domain. Instead we go one step further and propose a model which relies completely on visual content, since in our *one-of-a-kind* setting all items are cold-start items. We also propose new domain-specific ranking strategies that go beyond the basic ranking strategy used in BPR.

Only one work, the research by He et al. (2016) resembles ours in terms of the topic (artwork recommendation) and the use of visual features. However, there are important differences. The most important one is that the Behance's dataset used by He et al. is orders of magnitude bigger and consists of digital art, which means items are usually available for very long periods of time, if not always, and they are a few clicks away from users. This allows many more explicit and implicit interactions between users and items, which favors more traditional collaborative filtering practices, such as learning user and item specific variables. In contrast, we use an orders of magnitude smaller dataset of physical artwork purchases, so the collaborative signals are much weaker. This motivated us to design a neural network model that maximizes generalization to new users and items. We do so by leveraging the visual content of artworks and without learning user or item specific variables: the network's weights implicitly capture all the collaborative information and domain-specific heuristics used during training, but the network works as a content-based recommender during testing. Some other differences: in our work we evaluate several handcrafted and CNN visual features, whereas He et al. only use features extracted with VGG19. In our work we report the results of a small user study with expert curators to shade more light on the offline results, whereas He et al. only conducted offline experiments.

4. MATERIALS

The online web store *UGallery* has been selling artworks for more than 10 years (Weinswig, 2016). They support emergent artists by helping them sell their artworks online. The *UGallery* website allows users (customers) to search for items and browse the catalog based on different attributes with a predefined order: orientation, size, medium, style and others, as seen on the left side of Figure 4.1. However, what their current system does not support is the exploration of items via personalized recommendations, which is exactly what we aim for in this paper.

UGallery provided us with an anonymized dataset of $2,378$ users, $6,040$ items and $5,336$ purchases (transactions) of artistic artifacts, where all users have made at least one transaction. In average, each user has bought 2-3 items in the latest years[1]. Figure 4.2 shows the distribution of purchases per user. The distribution is skewed since most users

[1]Our collaborators at UGallery requested us not to disclose the exact dates.

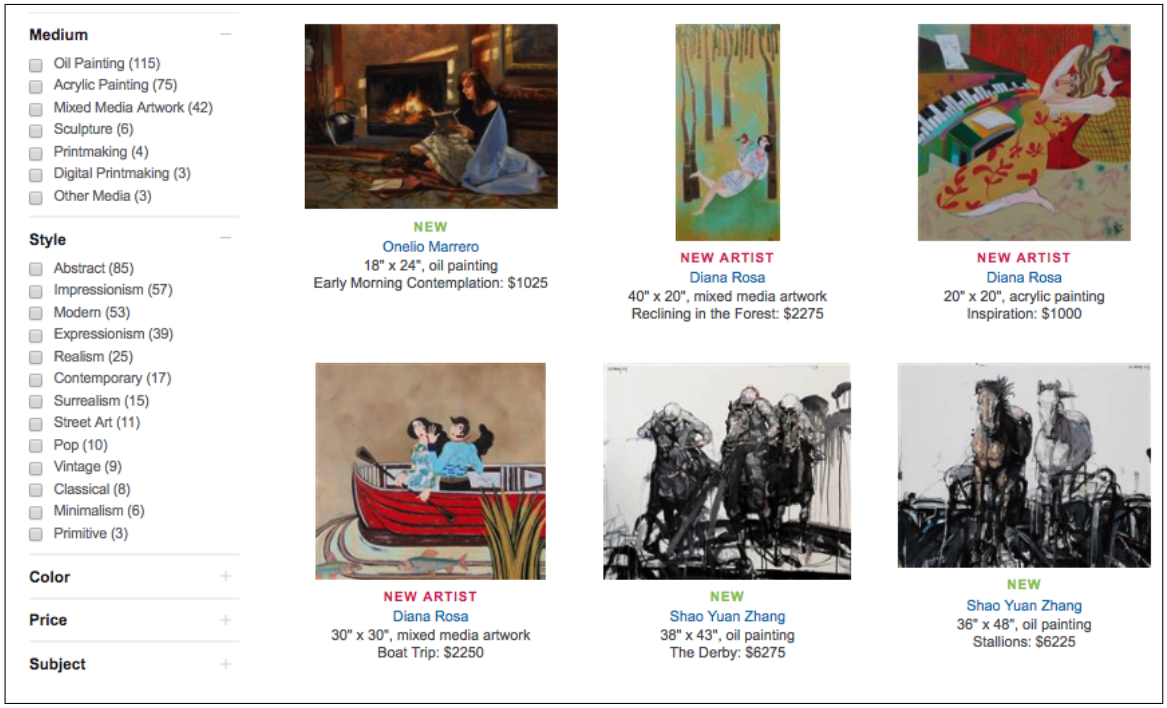

Figure 4.1. Screenshot of the search interface of *UGallery*. Users can filter by different facets on the left side.

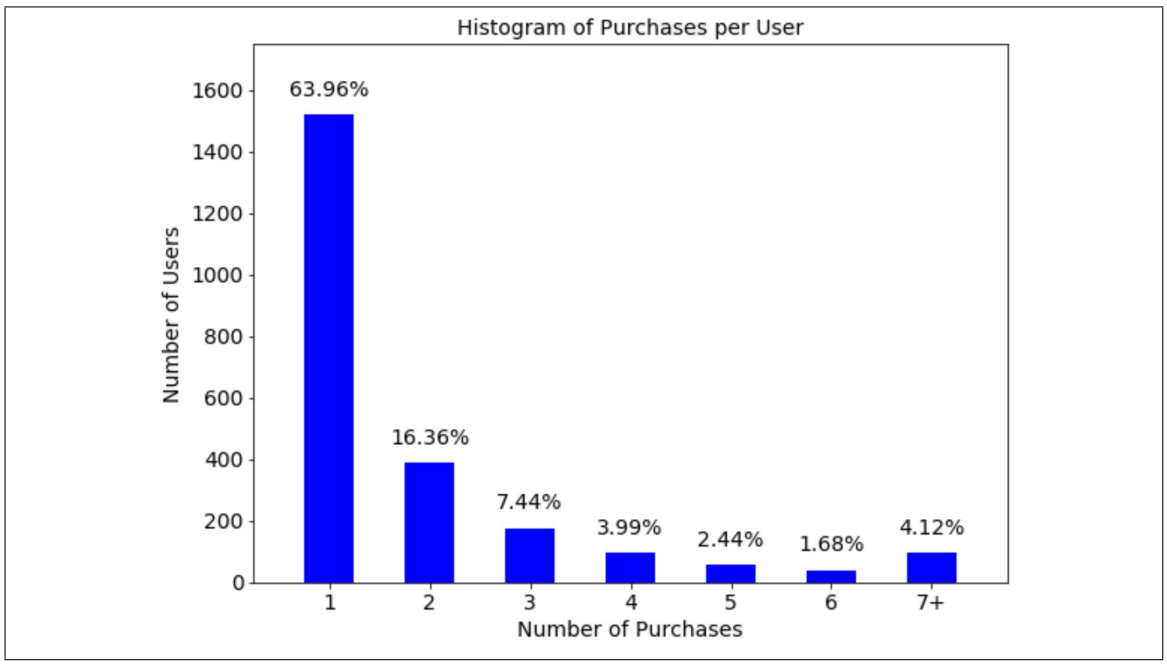

Figure 4.2. Distribution of purchases per user. It resembles the typical skewed user consumption behavior in online websites.

(1521 in total) bought only one item, and only a few users (98 in total) have bought 7 or more items. Our data is not atypical, since it resembles the rating distribution of the Netflix prize or the Movielens dataset, where a few users account for most of the activity and most users have little or none (Harper & Konstan, 2015; Bennett, Lanning, et al., 2007).

The artworks in the *UGallery* dataset were manually curated by experts. Hence, every artwork has been described with metadata *attributes* to enable the user to filter and browse in the *UGallery* interface. In total, there are four attributes (Color, Subject, Style and Medium), which are described with their respective *attribute values* in Table 4.1. It is important to note that only from the very latest years onwards the artworks started being filled with all their attributes more systematically. As such, there is a distribution of attributes present and absent in the artworks, which is shown in Table 4.2. For instance, *Color* (97.05%) is present in almost all the artworks, whereas *Subject* is only present in 45,10%.

Table 4.1. Metadata attributes and attribute values for artworks in the *UGallery* dataset

Attribute	**Values**
Color	B&W, Beige, Black, Blue, Brown, Dark Blue, Dark Green, Dark Red, Green, Grey, Orange, Pink, Purple, Red, Turquoise, Violet, White, Yellow
Subject	Animals, Architecture, Cuisine, Fantasy, Fashion, Flora, Landscape, Nature, Nudes, People, Religion, Seascape, Sports, Still Life, Travel, Western
Style	Abstract, Classical, Expressionism, Impressionism, Minimalism, Modern , Non-representational, Pop, Primitive, Realism, Representational, Street Art, Street Photography, Surrealism, Vintage
Medium	Acrylic Painting, Ceramic Artwork, Chalk Drawing, Charcoal Drawing, Colored Pencil, Digital Printmaking, Drawing Artwork, Encaustic Artwork, Gouache Painting, Ink Artwork, Marker Artwork, Mixed Media Artwork, Oil Painting, Other Media, Pastel Artwork, Pencil Drawing, Photography, Printmaking, Sculpture, Watercolor

In addition to these curated attributes, the artwork metadata also includes another important source of information: the artwork's artist. In the *UGallery* dataset, each artwork is associated to a unique artist. In total, there are 573 artists, who have 10.54 artworks in average each for sale. Figure 4.3 shows more details on the artist distribution.

Table 4.2. Statistics of attributes' presence among artworks in the *UGallery* dataset

	Color	**Style**	**Subject**	**Medium**
Present	5,862 (97.05%)	3,230 (53.48%)	2,724 (45.10%)	6,040 (100%)

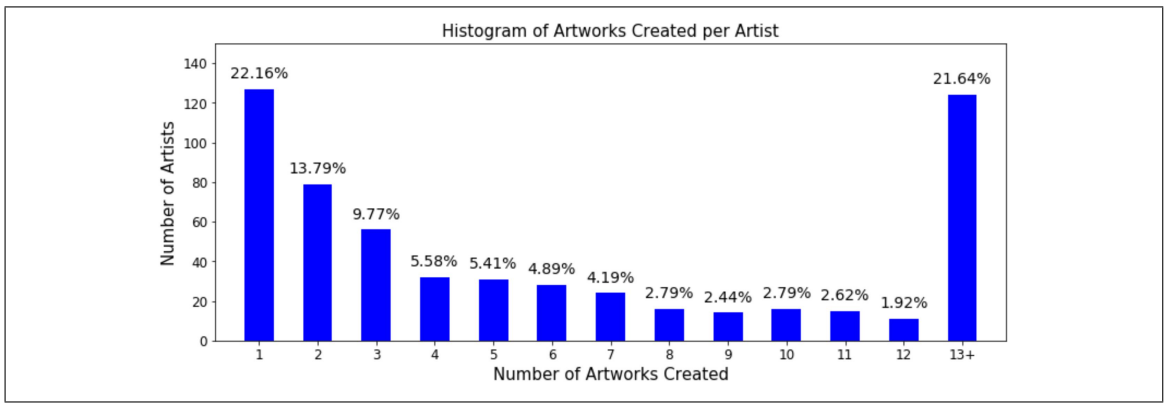

Figure 4.3. Distribution of created artworks per artist

5. FIRST APPROACH: CONTENT-BASED RECOMMENDATION

5.1. Problem Statement: Content-Based Recommendation of Artworks

Based on the formulation of the recommendation problem by Adomavicious and Tuzhilin (2005), we formalize our content-based recommendation problem with the following definitions.

Let U be the set of all users and I be the set of all items (physical artworks) available in the inventory. Let s be a function which measures the utility of an item i to a user u, $s : U \times I \rightarrow R$, where R is a totally ordered set (e.g., non-negative real numbers within a certain range). In other words, a utility function s, which, given a user $u \in U$ and an item $i \in I$, returns a predicted utility score r. Now, our end goal is to identify the set R_u of "top-k items" $\{i_1..i_k\}$ which maximize the utility of the user u, i.e., the list of recommended items:

$$R_u = \underset{\{i_1..i_k\}}{\operatorname{argmax}} \sum_{j=1}^{k} s(u, i_j) \tag{5.1}$$

. Due to the *one-of-a-kind* nature of our artwork items, which hinders the application of collaborative filtering, here we will formulate our utility function as a content-based recommendation problem. In a content-based recommender, the utility function $s(u, i)$ in Adomavicious and Tuzhilin is defined as:

$$s(u, i) = score(ContentBasedProfile(u), Content(i)) \tag{5.2}$$

, where $score(x, y)$ usually represents a similarity function (such as cosine or BM25 in the case of documents), and $ContentBasedProfile$ of user u and $Content$ of item i can be respectively represented as vectors, such as TF-IDF vectors using the bag-of-words document model. In our case, $ContentBasedProfile(u)$ will be the set of artworks P_u already purchased by user u. $Content(i)$ is a vector representation of the artwork i, its dimensions can represent different features. In this particular research, these features can

Table 5.1. Symbols used in the formulation of content-based methods

Symbol	Description
U, I	user set, item set
u, i	a specific user or item (resp.)
P	set of all items purchased in the system up to an arbitrary point in time
P_u	set of all items purchased by user u up to an arbitrary point in time, we refer to these items as the *user profile* or the *user model*, indistinctly
CAV_i^X	set of all curated attribute values of type X present in item i, where X can be either *Color*, *Subject*, *Style*, *Medium* or *All* (all curated attributes at the same time)
a_i	the artist (creator) of item i
V_i	vector of visual features of item i, either manually engineered or obtained with a pre-trained CNN
V_i^X	vector of visual features (of item i) of the specific type X (where X can be e.g. *AlexNet*, *VGG*, *ResNet*, *LBP*, etc.)

be: i) curated attributes, ii) the artist (artwork's creator), iii) visual features extracted with pre-trained CNNs, e.g. AlexNet, VGG and ResNet and iv) handcrafted visual features, e.g. attractiveness features and local binary patterns (LBP).

In the following section we will explain in detail which form the function $score(x, y)$ takes depending on the different features used.

5.2. Content-based Artwork Recommender Methods

In this section we describe six different content-based artwork recommender methods, which we have implemented to tackle the *one-of-a-kind* recommendation problem. Table 5.1 contains an overview of symbols used in the following sub-sections.

5.2.1. Most Popular Curated Attribute Value (MPCAV)

The Most Popular Curated Attribute Value method is the first and most simple approach we tested. Since the concept of "popular item" is meaningless in a *one-of-a-kind* setting, instead we recommend based on the most popular *curated attribute values*. Given an artwork i and its set of curated attribute values CAV_i^X (where X can be either *Color*,

Subject, *Style*, *Medium*, or *All*), we compute the MPCAV score as the sum of the frequencies (popularities) of each of its curated attribute values. More formally, the MPCAV score is calculated as follows:

$$score(i)_{MPCAV} = \sum_{v \in CAV_i^X} \sum_{j \in P} \mathbb{1}(j, v) \tag{5.3}$$

, where P is the set of products purchased so far, and $\mathbb{1}(j, v)$ is an indicator function, which returns 1 if item j has curated attribute value v or 0 otherwise. Intuitively, an item will have a higher score if its curated attribute values are more frequent (popular) among items already purchased in the system. Finally, we rank the items based on this score and recommend the top-n.

5.2.2. Personalized Most Popular Curated Attribute Value (PMPCAV)

This method is equivalent to MPCAV, with the only difference that we just look at the past purchases of user u instead of the past purchases of the whole system. More formally, the formula for the PMPCAV scoring function is:

$$score(u, i)_{PMPCAV} = \sum_{v \in CAV_i^X} \sum_{j \in P_u} \mathbb{1}(j, v) \tag{5.4}$$

, which is almost exactly as Equation 5.3, but here we consider only the set of items purchased by the user u, i.e., the set P_u. Then we can rank items and recommend the top-n based on this score. However, due to the absence of curated tags in many items, sometimes a user profile cannot be built from past untagged items, or we cannot find enough items to recommend matching a certain user profile. In those cases, we switch to MPCAV to fill up the top-n recommendation.

A weakness of this method compared to MPCAV is that it requires at least one previous purchase from the user to make a personalized recommendation. On the positive side,

by considering the user's preferences, one should expect recommendations to be more accurate.

5.2.3. Personalized Favorite Artist (FA)

Besides curated attributes, the artwork metadata also includes another important source of information: the artist who created the painting. The FA method leverages this information by recommending artworks created by artists that the user has shown favoritism for. More formally, given a user u and an item i, the FA scoring function is defined as follows:

$$score(u, i)_{FA} = \sum_{j \in P_u} \mathbb{1}(j, a_i) \tag{5.5}$$

, where $\mathbb{1}(j, a_i)$ is an indicator function that returns 1 if the artist a_i of artwork i is also the creator of artwork j, or 0 otherwise (in our dataset, each artwork is associated to a single creator). Intuitively, an artwork has a higher score if the user has purchased more artworks from the same artist in the past. Then we rank and recommend the top-n artworks based on this score. In case there are too few items with a positive score to recommend (i.e. not enough artworks from the user's favorite artists in stock), we resort to the globally most favorite artists to rank the remaining artworks and fill the top-n recommendation.

5.2.4. Learned Visual Features: Deep Convolutional Neural Network Embeddings (CNN)

Since the dataset contains one image for every item, we also tested using visual features from images for content-based artwork recommendation. Taking advantage of the latest advances in computer vision, we used several CNNs to extract visual features from each image. The CNNs used in our experiments are AlexNet (Krizhevsky et al., 2012), VGG19 (Simonyan & Zisserman, 2014), Inception-V3 (Szegedy, Vanhoucke, Ioffe, Shlens, & Wojna, 2016), ResNet50 (K. He et al., 2016), Inception-ResNet-V2 (Szegedy,

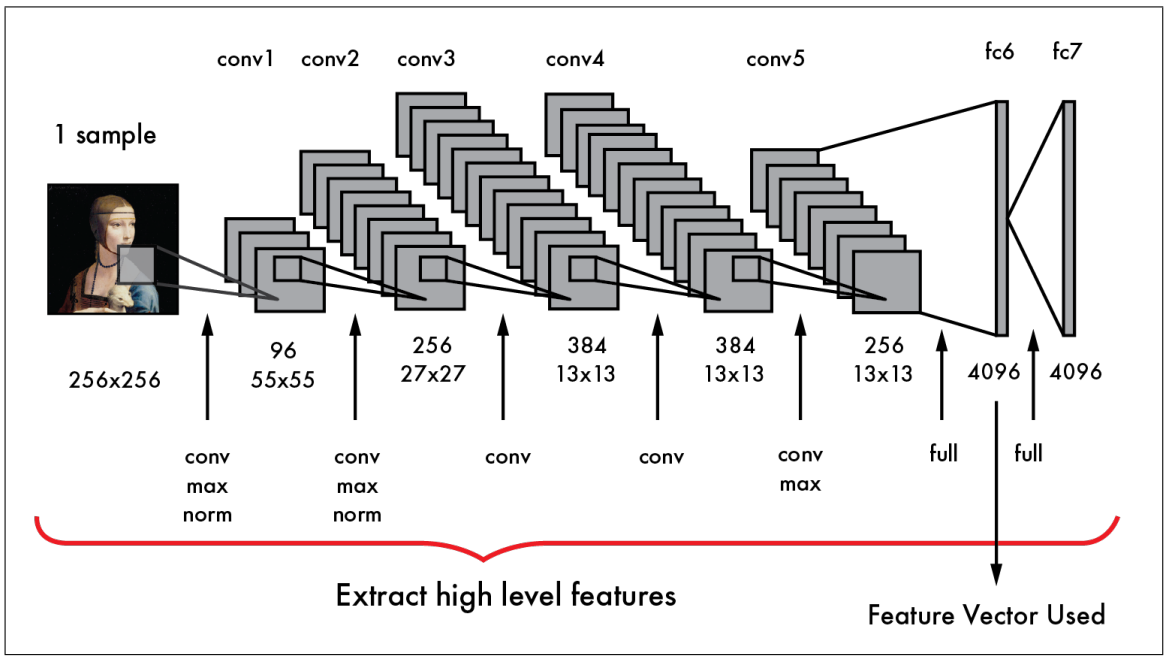

Figure 5.1. AlexNet's architecture. This illustrates the process to obtain a feature vector from an image. A convolutional window passes over the image, from each layer to the next layer, with different shapes and strides in every layer. In this example, the activation values at layer fc6 are finally used. For other CNNs the process is analogous. This figure is inspired by (Karnowski, 2015).

Ioffe, Vanhoucke, & Alemi, 2017) and NasNet Large (Zoph, Vasudevan, Shlens, & Le, 2017). All of these networks were developed for the task of image classification and both pre-trained and tested with the ImageNet Large Scale Visual Recognition Challenge dataset (Deng et al., 2009), in which they achieved state-of-the-art results at their respective times of publication. Given any of these CNNs, a common procedure for feature extraction consists of providing an image as input to a CNN and saving the activation values of a hidden layer, e.g. the activations of the last convolutional layer or the activations of the first fully connected layer in the network. For instance, in the case of AlexNet one could save the activations of the first fully connected layer known as fc6. This is illustrated in Figure 5.1.

CNN utility score. Given a user u who has consumed a set of artworks P_u, and an arbitrary artwork i from the inventory, the score of this item i to be recommended to u is

defined as:

$$score(u,i)_X = \begin{cases} \max\limits_{j \epsilon P_u} \{sim(V_i^X, V_j^X)\} & (maximum) \\ \frac{\sum\limits_{j \in P_u} sim(V_i^X, V_j^X)}{|P_u|} & (average) \\ \frac{\sum\limits_{r=1}^{\min\{K,|P_u|\}} \max\limits_{j \epsilon P_u}{}^{(r)} \{sim(V_i^X, V_j^X)\}}{\min\{K,|P_u|\}} & (average\ top\ K) \end{cases} \tag{5.6}$$

, where V_z^X is a feature vector of type X associated to item z. In this particular case V_z^X stands for the vector embedding of item z obtained with a pre-trained CNN of type X, e.g. VGG, AlexNet. $\max^{(r)}$ denotes the r-th maximum value, e.g. if r = 1 it is the overall maximum, if r = 2 it is the second maximum, and so on. $sim(V_i, V_j)$ denotes a similarity function between vectors V_i and V_j. In this particular case, the similarity function used was cosine similarity, expressed as:

$$sim(V_i, V_j) = cos(V_i, V_j) = \frac{V_i \cdot V_j}{\|V_i\| \|V_j\|} \tag{5.7}$$

Essentially, the score in Equation 5.6 looks at the similarity between item i and each item j in the user profile P_u, and then aggregates these similarities in 3 possible ways: taking either (a) the maximum, (b) the average or (c) the average of the top-K most similar items, where K can be tuned empirically.

In addition, we also studied the performance of using multiple CNNs at the same time. For this purpose, we implemented the following hybrid score:

$$\begin{aligned} score(u,i)_{CNN} = \alpha_1 \cdot score(u,i)_{AlexNet} \\ + \alpha_2 \cdot score(u,i)_{VGG19} \\ + \alpha_3 \cdot score(u,i)_{InceptionV3} \\ + \alpha_4 \cdot score(u,i)_{ResNet50} \\ + \alpha_5 \cdot score(u,i)_{InceptionResNetV2} \\ + \alpha_6 \cdot score(u,i)_{NasNetLarge} \end{aligned} \tag{5.8}$$

, where each specific $score(u,i)_X$ is calculated following Equations 5.6 and 5.7, and coefficients α_i are weights to perform a linear combination between the scores. These weights are optimized by multiple iterations of greedy weight sampling. As will be shown in section 5.4.2, the hybrid approach achieved the best results.

5.2.5. Handcrafted Visual Features (HVF)

The visual features obtained with CNN techniques are of latent nature, i.e., they are not easily interpretable in terms of more intuitive features such as image colorfulness or brightness. To mitigate this problem, one might want to take advantage of manually engineered visual features, which usually are much more intuitive and explainable than neural features. Moreover, they are suitable to be used in a search interface to support navigation. For example, imagine a use case where a content-based recommender uses the brightness of an image to find similar items. This information could be used to make an explanation –*you might like this image because of its brightness*– or to allow the user to filter search results based on the paintings' level of brightness.

In order to choose which visual features to extract, we surveyed related work and found features related to *attractiveness* as potentially useful.

Attractiveness. San Pedro and Siersdorfer in (2009) proposed several explainable visual features that can capture to a great extent the attractiveness of an image posted on Flickr[1]. Following their procedure, for every image in our *UGallery* dataset we calculated: (a) average brightness, (b) saturation, (c) sharpness, (d) RMS-contrast, (e) colorfulness and (f) naturalness. In addition, we added (g) entropy, which is a good way to characterize and measure the texture of an image (Gonzalez, Eddins, & Woods, 2004). These metrics have also been used in another study (Trattner & Elsweiler, 2017), where they are successfully used to nudge people with attractive images to take up more healthy recipe recommendations.

Since each feature varies within different value ranges (e.g. 0-1, 10-100), we applied a feature-wise min-max normalization to prevent biases in similarity calculations. Following, we provide a more detailed description of these attractiveness-based features:

- *Brightness* measures the level of luminance of an image. For images in the *YUV* color space, we obtain the average of the luminance component *Y* as follows:

$$B = \frac{1}{N} \sum_{x,y} Y_{x,y} \tag{5.9}$$

  , where N is the amount of pixels and $Y_{x,y}$ is the value of the luminance in the pixel (x, y)

- *Saturation* measures the vividness of an image. For images in the *HSV* or *HSL* color space, we obtain the average of the saturation component *S* as follows:

$$S = \frac{1}{N} \sum_{x,y} S_{x,y} \tag{5.10}$$

  , where N is the amount of pixels and $S_{x,y}$ is the value of the saturation in the pixel (x, y)

[1] `https://www.flickr.com/`

- *Sharpness* measures the detail level of an image. For an image in gray-scale, it can be obtained using a Laplacian filter and luminance around every pixel:

$$L(x,y) = \frac{\delta^2 I}{\delta x^2} + \frac{\delta^2 I}{\delta y^2} \tag{5.11}$$

$$Sh = \frac{\sum_{x,y} \frac{L(x,y)}{\mu_{x,y}}}{n} \tag{5.12}$$

, where n is the number of pixels and $\mu_{x,y}$ is the average luminance of the pixels around the pixel (x, y).

- *Colorfulness* measures how distant the colors are from the gray color. For images in the RGB space, it can be obtained with the following formulas:

$$C = \sigma_{rgyb} + 0.3 \cdot \mu_{rgyb} \tag{5.13}$$

$$\sigma_{rgyb} = \sqrt{\sigma_{rg}^2 + \sigma_{yb}^2} \tag{5.14}$$

$$\mu_{rgyb} = \sqrt{\mu_{rg}^2 + \mu_{yb}^2} \tag{5.15}$$

, where μ_{rg}^2, μ_{yb}^2 are the means of the components of the opponent color space. σ_{rg}^2, σ_{yb}^2 are the standard deviations of the component of opponent color space. This color space is defined as:

$$rg = R - G \tag{5.16}$$

$$yb = \frac{1}{2}(R + G) - B \tag{5.17}$$

- *Naturalness* measures the naturalness of an image by grouping the pixels into Sky, Grass and Skins pixels and applying the formula in San Pedro and Siersdorfer (2009). First, using the HSL color space, the pixels are filtered considering only the ones with $20 \leq L \leq 80$ and $S > 0.1$. Then, they are grouped by their hue value in three classes “A - Skin”, “B - Grass” and “C - Sky”, which are defined as follows:
  - pixels with $25 \leq hue \leq 70$ belong to the “A - Skin” set.
  - pixels with $95 \leq hue \leq 135$ belong to the “B - Grass” set.

– pixels with $185 \leq hue \leq 260$ belong to the "C - Sky" set.

For each set, average saturation is calculated and denoted as μ_S. Then, local naturalness for each set is calculated using the following formulas:

$$N_{skin} = e^{-0.5\left(\frac{\mu_S^A - 0.76}{0.52}\right)^2} \tag{5.18}$$

$$N_{Grass} = e^{-0.5\left(\frac{\mu_S^B - 0.81}{0.53}\right)^2} \tag{5.19}$$

$$N_{Sky} = e^{-0.5\left(\frac{\mu_S^C - 0.43}{0.22}\right)^2} \tag{5.20}$$

After this, the Naturalness value is obtained by:

$$Na = \sum_i \omega_i N_i,\ i \in \{"Skin", "Grass", "Sky"\} \tag{5.21}$$

, where ω_i is the amount of pixels of set i divided by the total pixels in the image.

- *RMS-contrast* measures the variance of luminance in an image using the intensity of each pixel:

$$C^{rms} = \frac{1}{n} \sum_{x,y}^{n} (I_{x,y} - \bar{I})$$

, where $I_{x,y}$ is the intensity of the pixel (x, y) and $\bar{I}$ is the average intensity.

- *Entropy*: The entropy of a gray-scale image is a way to measure and characterize the texture of the image (Gonzalez et al., 2004). Shannon's entropy is applied to the histogram of values of every pixel in a gray-scale image. The formula is defined as follows:

$$E = -\sum_{x \in [0..255]} p(x) \log p(x) \tag{5.22}$$

, where $p(x)$ is the probability of finding the gray-scale value x among all the pixels in the image.

Attractiveness utility score. We put the 7 attractiveness-based features into a single $\mathbb{R}^7$ vector, which we denote as $V_i^{Attract}$. Then, to calculate the similarity between two vectors

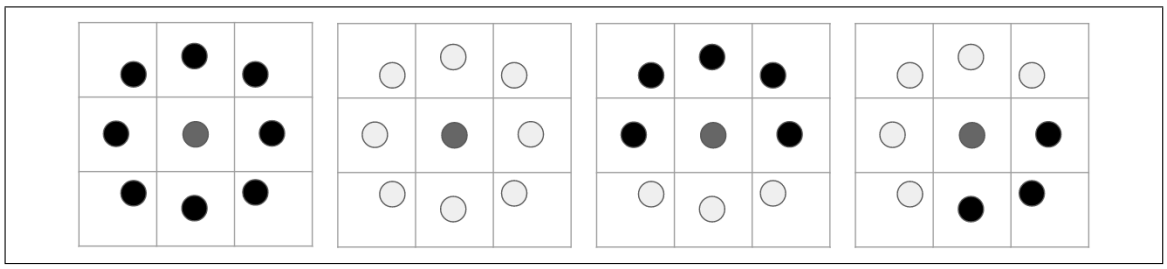

Figure 5.2. Examples of pixelwise patterns extracted with local binary patterns (LBP). Each small square is a pixel, and these boxes with 9 pixels each represent patterns. The black circles represent pixels with value over a threshold (=1), while gray circles represent pixels with value below a threshold (=0). The threshold is set by the value of the pixel in the center of the pattern.

$V_i^{Attract}$ and $V_j^{Attract}$ we used cosine similarity, as per Equation 5.7:

$$sim(V_i^{Attract}, V_j^{Attract}) = cos(V_i^{Attract}, V_j^{Attract}) \tag{5.23}$$

Finally, the utility score ($score(u,i)_X$) is computed with the same similarity aggregation techniques outlined in Equation 5.6: *maximum*, *average* and *average-top-k*.

LBP. Another set of handcrafted features we explored are the *Local Binary Patterns* (LBP) (Ojala, Pietikäinen, & Harwood, 1996). Although this is not an actual "explicit" visual feature, it is a traditional baseline in several computer vision tasks such as image classification, so we tested it for the task of recommendation too. LBP is not represented as a scalar value, but rather as a feature vector of 59 dimensions. The values in the LBP feature vector represent counts in a histogram of the patterns found on an image. Figure 5.2 shows four of such patterns as example.

LBP utility score. Since the output of LBP is a feature vector, we calculated the similarity between two vectors V_i^{LBP} and V_j^{LBP} as we did with most of the feature vectors, using cosine similarity:

$$sim(V_i^{LBP}, V_j^{LBP}) = cos(V_i^{LBP}, V_j^{LBP}) \tag{5.24}$$

Finally, the utility score ($score(u,i)_{LBP}$) is calculated using the same similarity aggregation techniques outlined in Equation 5.6: *maximum*, *average* and *average-top-k*.

MEVF hybrid utility score. Like we did with CNNs, we also studied the performance of combining Attractiveness and LBP together by merging both scores with a linear combination, analogous to Equation 5.8.

5.2.6. Hybrid Recommendations (Hybrid)

Since different methods can measure different sources of similarity between items and the user profile, we developed a hybrid recommender model which integrates the previous approaches. The basic idea is to compute a hybrid score as a convex linear combination of the scores of individual methods. We took the best performing version of each individual method and tested multiple hybrid combinations of them.

Formally, given a user u who has purchased a set of artworks P_u, and an arbitrary artwork i from the inventory, we compute the hybrid score of item i for user u as a convex linear combination of multiple scores, which for general case is defined as:

$$score(u,i)_{Hybrid} = \beta_1 \cdot score(u,i)_{\text{FA}} \tag{5.25}$$

$$+ \beta_2 \cdot score(u,i)_{\text{CNN}}$$

$$+ \beta_3 \cdot score(u,i)_{\text{HVF}}$$

$$+ \beta_4 \cdot score(u,i)_{\text{PMPCAV}}$$

(5.26)

, where β are global (non-personalized) coefficients such that $0 \leqslant \beta_i \leqslant 1$ and $\sum_i \beta_i = 1$. The β coefficients were tuned empirically by multiple iterations of a greedy weight sampling search algorithm[2]. In the equation, $score(u,i)_{\text{CNN}}$ and $score(u,i)_{\text{HVF}}$ are calculated

[2]To obtain the weights for the different methods, we initialize the coefficients proportional to the individual performance (concretely, Precision@20) of each method and randomly sample weights around them. We evaluate all these weights, keep the best ones and reiterate, each time narrowing the weight search space in a greedy fashion. The performance tends to converge after 3 to 4 iterations.

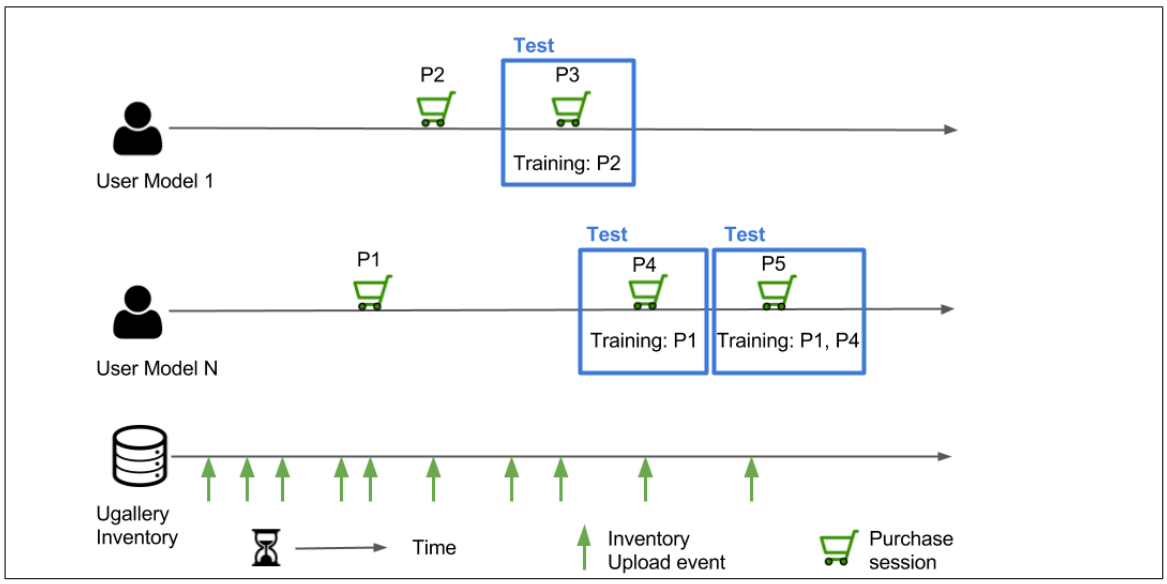

Figure 5.3. Offline evaluation procedure. Each surrounding box represents a test, where we predict the items of the purchase session. In the figure, we predict which artworks User 1 bought in purchase P3. 'Training:P2' means we used items from purchase session P2 to train the model.

as in Equation 5.8. Meanwhile, $score(u,i)_{\text{PMPCAV}}$ and $score(u,i)_{\text{FA}}$ had to be slightly modified to ensure normalized values in the range $[0,1]$:

$$score(u,i)_{\text{PMPCAV}} = \frac{\sum\limits_{v \in CAV_i^{All}} \sum\limits_{j \in P_u} \mathbb{1}(j,v)}{\sum\limits_{j \in P_u} |CAV_j^{All}|} \tag{5.27}$$

$$score(u,i)_{\text{FA}} = \frac{\sum\limits_{j \in P_u} \mathbb{1}(j,a_i)}{|P_u|} \tag{5.28}$$

, which are almost the same as Equations 5.4 and 5.5 but with the addition of a normalizing denominator that represents the theoretical maximum of the score in each case.

5.3. Evaluation Methodology

The evaluation had two stages. The first was an offline evaluation, conducted using the dataset of transactions (purchases) described in Section 4. With this offline evaluation we

can answer research questions RQ1, RQ2 and RQ3. The second stage was performed with expert curators from the UGallery store. We developed a web interface where the experts could rate recommendations based on algorithms selected from the offline evaluation, and we analyzed consistency between results of both stages (RQ4).

5.3.1. Offline Evaluation

The evaluation protocol we follow in this paper is the one usually used in order to evaluate predictive models and recommender systems offline in a time-based manner (Macedo, Marinho, & Santos, 2015). Hence, the *UGallery* dataset was split into training and test samples according to the time line of every user, as seen in Figure 5.3. With this setting, we attempt to predict the items purchased by the user in every transaction, where the training set contains all the artworks bought by a user previous to the transaction to be predicted.

Figure 5.3 shows that for every user we test the predictions made for every purchase session except for the first one of each user. For instance, for User 1 we tested the predicted items of purchase *P3* using items in *P2* as training. In the same Figure, for User N we performed two predictions tasks: the first one predicting items bought in purchase *P4* using *P1* as training, and then testing a prediction on purchase *P5* using *P1* and *P4* as training. In our evaluation, most of the experiments considered only users who had at least 2 purchase sessions. Users who only had a single purchase session in their whole history were considered *cold start* users (only MPCAV and Random were able to make predictions in those cases, since they are non-personalized methods).

5.3.2. Online Evaluation

The online evaluation involved 8 expert curators from UGallery. We asked each expert to send us a list of 10 of their preferred paintings from the current UGallery dataset, which they sent us via email. For each expert we created five lists of recommendations based on different methods: FA, HVF, CNN, and the hybrids CNN+HVF, and FA+CNN+HVF.

Figure 5.4. Screenshot of the upper part of the interface used in the expert evaluation. On the left the items liked by the user. The large table to the right shows one column per each method used to make recommendations.

Each recommendation list had 10 items, and the experts had to rate each painting recommended with stars in a scale from 1 to 5. We used ratings rather than likes/dislikes to evaluate the recommendations in order to give experts the chance to express their perception of relevance with higher granularity. Unlike regular art consumers for which a preference rating of two or three stars might be hard to discriminate, experts are more likely to understand detailed levels of relevance of the paintings recommended. In total, each expert rated 50 items. A screenshot of the rating interface for a ficticious user called "Madeline" is shown in Figure 5.4. We stored the user id, item id and the ratings over every painting for each method, to calculate the evaluation metrics and compare the results.

5.3.3. Evaluation Metrics

Table 5.2 shows a summary of symbols used in this section. As suggested by Cremonesi et al. (Cremonesi, Koren, & Turrin, 2010) for Top-N recommendation, for our offline evaluations we used Recall@k ($R@k$), Precision@k ($P@k$) and F1-score@k ($F1@k$), as shown in the equations below:

$$p@k(t) = \frac{|r_t^k \cap R_t|}{k} \tag{5.29}$$

Table 5.2. Symbols used in the formulation of evaluation metrics

Symbol	Description
t	a test case during the execution of an offline evaluation of a certain recommendation algorithm
u_t	user whose shopping basket is predicted during offline test case t
r_t^k	list of top-k items recommended to user u_t at offline test case t
R_t	the set of relevant items (i.e. items in the shopping basket) of user u_t during offline test case t
T_u	the set of all test cases performed with purchase sessions of user u
U_r	set of all users who received at least 1 recommendation during a certain offline evaluation (i.e., all $u \in U$ such that $\|T_u\| \geq 1$)
$i_{t,z}$	item appearing at position z in the recommended list at offline test t
vc_i	the visual cluster that item i belongs to
PS	total number of purchase sessions in the system

$$r@k(t) = \frac{|r_t^k \cap R_t|}{|R_t|} \tag{5.30}$$

$$f1@k(t) = 2 \cdot \frac{p@k(t) \cdot r@k(t)}{p@k(t) + r@k(t)} \tag{5.31}$$

$$P@k = \frac{1}{|U_r|} \sum_{u \in U_r} \left(\frac{1}{|T_u|} \sum_{t \in T_u} p@k(t) \right) \tag{5.32}$$

$$R@k = \frac{1}{|U_r|} \sum_{u \in U_r} \left(\frac{1}{|T_u|} \sum_{t \in T_u} r@k(t) \right) \tag{5.33}$$

$$F1@k = \frac{1}{|U_r|} \sum_{u \in U_r} \left(\frac{1}{|T_u|} \sum_{t \in T_u} f1@k(t) \right) \tag{5.34}$$

, where $p@k(t)$, $r@k(t)$ and $f1@k(t)$ are precision, recall and f1-score at k, respectively, measured during the test case t, whereas $P@k$, $R@k$ and $F1@k$ are the overall aggregations of precision, recall and f1-score at k, respectively, by first calculating user averages and then the average of these averages. These are the evaluation metrics that we report in Section 5.4.

In addition, we also report *Normalized Discounted Cumulative Gain (nDCG)* (Manning, Raghavan, Schütze, et al., 2008) which is a ranking-dependent metric that not only measures how relevant the items are but also takes the position of the items in the recommended list into account. The $nDCG$ metric with a cut-off of k items in the recommended list is based on the *Discounted Cumulative Gain (DCG@k)* which is defined as follows:

$$DCG@k(t) = \sum_{z=1}^{k} \frac{2^{B_t(i_{t,z})} - 1}{log_2(1+z)} \tag{5.35}$$

, where $B_t(i_{t,z})$ is a function that returns the graded relevance of item $i_{t,z}$ appearing at position z in the recommended list during the test case t. In our case, $B_t(i_{t,z})$ basically returns 1 if item $i_{t,z}$ was present in the shopping basket of test case t, and 0 otherwise. $nD@k$ is calculated as $DCG@k$ divided by the ideal $DCG@k$ value $iDCG@k$ which is the highest possible $DCG@k$ value that can be achieved if all the relevant items were recommended in the correct order (i.e., all shopping basket items appearing first in the recommended list). Taken together, the overall $nDCG@k$ is defined as follows:

$$nD@k = \frac{1}{|U_r|} \sum_{u \in U_r} \left(\frac{1}{|T_u|} \sum_{t \in T_u} \frac{DCG@k(t)}{iDCG@k(t)} \right) \tag{5.36}$$

In addition, we calculated *user coverage* (UC), expressed as:

$$UC = \frac{|U_r|}{|U|} \tag{5.37}$$

User Coverage is defined as the number of users for whom at least one recommendation could be generated ($|Ur|$) divided by total number of users $|U|$ (Lacic et al., 2015).

We also report *session coverage* (SC), expressed as:

$$SC = \frac{\sum_{u \in U_r} |T_u|}{PS} \tag{5.38}$$

Session Coverage is defined as the number of purchase sessions in which the recommender was able to generate a recommendation (i.e., total number of valid test cases) divided by the total number of purchase sessions of the system (PS).

Content-based recommendation techniques are usually much more susceptible to over-specialization than other recommendation techniques, such as e.g. collaborative filtering (Parra & Sahebi, 2013). Therefore, in order to measure the degree of this effect we also calculated several diversity metrics.

The first of these metrics is the *Artist Diversity* ($^{\text{D@k}}_{\text{artist}}$), defined as:

$$^{\text{D@k}}_{\text{artist}} = \frac{\sum_{u \in U_r} \sum_{t \in T_u} \left| \bigcup_{i \in r_t^k} \{ a_i \} \right|}{\sum_{u \in U_r} |T_u|} \tag{5.39}$$

, where a_i is item i's artist. The Artist Diversity measures the average number of distinct artists per recommendation. This metric is useful for getting a notion of how diverse a recommendation is in terms of the different artists recommended. The larger the metric, the more the chances of recommending items from novel artists to users.

Similarly, we also calculate *Color Diversity* and *Medium Diversity*, which are formally defined as:

$$^{\text{D@k}}_{\text{color}} = \frac{\sum_{u \in U_r} \sum_{t \in T_u} \left| \bigcup_{i \in r_t^k} \text{CAV}_i^{\text{color}} \right|}{\sum_{u \in U_r} |T_u|} \tag{5.40}$$

$$\underset{\text{medium}}{\text{D@k}} = \frac{\sum_{u \in U_r} \sum_{t \in T_u} \left| \bigcup_{i \in r_t^k} \text{CAV}_i^{\text{medium}} \right|}{\sum_{u \in U_r} |T_u|} \tag{5.41}$$

, where CAV_i^{color} and CAV_i^{medium} are defined in Table 5.1. *Color Diversity* and *Medium Diversity* measure the average number of distinct color values and medium values per recommendation, respectively. We do not use other curated attributes besides color and medium because these are the only ones present in (almost) all artworks, as previously shown in Table 4.2.

In addition to Artist, Color and Medium, it is also possible to learn visual categories directly from images by means of unsupervised techniques, e.g. clustering. To this end, we crawled 13,297 images from the *UGallery* website (a superset of the 6,040 images used in offline evaluations), and for each of these images we obtained a $\mathbb{R}^{13,824}$ feature vector by concatenating ResNet50 ($\mathbb{R}^{2,048}$) + AlexNet ($\mathbb{R}^{4,096}$) + Inception V3 ($\mathbb{R}^{2,048}$) + VGG19 ($\mathbb{R}^{4,096}$) + InceptionResNet V2 ($\mathbb{R}^{1,536}$). Then we applied z-score normalization and PCA to reduce the vector dimensionality to $\mathbb{R}^{200}$, so as to retain the most relevant visual features according to the natural distribution of the images. Finally, we used *K-Means* clustering to fit 100 clusters to this augmented image dataset (we tried multiple initializations and chose the one with top silhouette score). Thus, we calculate *Visual Cluster Diversity*, which is formally defined as:

$$\underset{\substack{\text{visual} \\ \text{cluster}}}{\text{D@k}} = \frac{\sum_{u \in U_r} \sum_{t \in T_u} \left| \bigcup_{i \in r_t^k} \{\, vc_i \,\} \right|}{\sum_{u \in U_r} |T_u|} \tag{5.42}$$

, where vc_i is defined in Table 5.2. This metric measures the average number of distinct visual clusters per recommendation.

In addition to clustering, the aforementioned $\mathbb{R}^{200}$ visual feature vectors can also be used for pairwise comparisons. Thus, we also calculate *Visual Pairwise Diversity* which we formally define as follows:

$$\underset{\text{visual}}{\overset{\text{D@k}}{\text{pairwise}}} = \frac{\sum_{u \in U_r} \sum_{t \in T_u} \underset{\text{visual}}{\overset{\text{D@k}}{\text{pairwise}}}(t)}{\sum_{u \in U_r} |T_u|} \tag{5.43}$$

$$\underset{\text{visual}}{\overset{\text{D@k}}{\text{pairwise}}}(t) = \frac{\sum_{y=1}^{k-1} \sum_{z=y+1}^{k} 0.5 \cdot \left[1 - cos(V_{i_{t,y}}^{\text{PCA(200)}}, V_{i_{t,z}}^{\text{PCA(200)}})\right]}{\frac{k \cdot (k-1)}{2}} \tag{5.44}$$

, where $\underset{\text{visual}}{\overset{\text{D@k}}{\text{pairwise}}}(t)$ is the average of the pairwise cosine distances between the top-k items of test case t's recommended list, $i_{t,y}$ and $i_{t,z}$ are the items at positions y and z, respectively, of test case t's recommended list, $V_i^{\text{PCA(200)}}$ is item i's $\mathbb{R}^{200}$ visual feature vector obtained with PCA, and $cos(x, y)$ stands for cosine similarity. In short, this metric gives a numeric estimate of how different two images are on average in a recommendation. The more different images are to each other, the bigger this score should get.

Finally, we can also compute a pairwise diversity metric based on the whole metadata. By combining Artist, Colors and Medium in a single set of metadata attribute values per item, we can use Jaccard Index to calculate *Jaccard Pairwise Diversity*, which we formally define as:

$$\underset{\text{jaccard}}{\overset{\text{D@k}}{\text{pairwise}}} = \frac{\sum_{u \in U_r} \sum_{t \in T_u} \underset{\text{jaccard}}{\overset{\text{D@k}}{\text{pairwise}}}(t)}{\sum_{u \in U_r} |T_u|} \tag{5.45}$$

$$\underset{\text{pairwise}}{\overset{\text{D@k}}{\underset{\text{jaccard}}{}}}(t) = \frac{\sum_{y=1}^{k-1}\sum_{z=y+1}^{k}\left[1 - \text{jaccard\_index}(S_{i_{t,y}}, S_{i_{t,z}})\right]}{\frac{k\cdot(k-1)}{2}} \tag{5.46}$$

$$\text{jaccard\_index}(S_i, S_j) = \frac{\left|S_i \cap S_j\right|}{\left|S_i \cup S_j\right|} \tag{5.47}$$

$$S_i = \text{CAV}_i^{\text{color}} \cup \text{CAV}_i^{\text{medium}} \cup \{\ a_i\ \} \tag{5.48}$$

In addition to these offline evaluation metrics, we also report Precision@k and nDCG@k for the online evaluation with 8 UGallery expert curators. In this setting, the metrics were calculated as follows:

$$nD@k = \frac{1}{8}\sum_{x=1}^{8}\frac{DCG@k(x)}{iDCG@k(x)} \tag{5.49}$$

$$DCG@k(x) = \sum_{z=1}^{k}\frac{2^{B_x(i_{x,z})} - 1}{log_2(1+z)} \tag{5.50}$$

$$P@k = \frac{1}{8}\sum_{x=1}^{8}p@k(x) \tag{5.51}$$

$$p@k(x) = \frac{1}{k}\sum_{z=1}^{k}\mathbb{1}_x(i_{x,z}) \tag{5.52}$$

, where x stands for the x-th expert curator, $i_{x,z}$ is the item appearing at position z in the list recommended to expert x, $B_x(i_{x,z})$ returns the original rating $S_x(i_{x,z})$ given by expert x to item $i_{x,z}$ if $S_x(i_{x,z}) >= 4$, or 0 otherwise, and $\mathbb{1}_x(i_{x,z})$ is an indicator function that

returns 1 if rating $S_x(i_{x,z}) \geq 4$, or 0 otherwise (i.e., we used 4 as the relevance threshold for the calculation of these metrics).

5.4. Results

In this section, we report the results focusing on different aspects. With respect to research question RQ1 –analyzing the impact of each single feature–, we analyze: a) metatadata features (personalized and non-personalized), and b) visual features (CNN and HVF). For RQ2, we compare between visual features and metadata. Regarding research question RQ3, we test several combinations of features to identify the best hybrid recommender in terms of ranking and accuracy. In Subsection 5.4.5 we assess RQ2 and RQ3 with respect to metrics of diversity. Finally, regarding research question RQ4, the online validation, we report and discuss the results of recommendations evaluated by expert curators from UGallery.

5.4.1. Metadata features (RQ1.1)

Table 5.3 summarizes all the results for this analysis of metadata features. Here we report MPCAV, its personalized version PMPCAV, and Favorite Artist (FA).

Most Popular Curated Attribute Value (MPCAV). We tested the performance of MPCAV features separately as well as combined (*MPCAV(All)*). Table 5.3 shows that these results are for the most part significantly better than random prediction but not by a wide margin. The highest performance is achieved by MPCAV (Medium), about 200% above random, whereas MPCAV (Subject) performs not significantly different from random prediction.

Personalized MPCAV (PMPCAV). In Table 5.3 we observe that the use of personalization causes a significant improvement in the ranking metrics over MPCAV. However,

Table 5.3. nDCG (nD), Recall (R), Precision (P), F1 Score (F1) and Coverage (UC and SC) for metadata based methods: MPCAV (by attribute), PMPCAV (by attribute), and FA. The best result for each metric and method group are highlighted. The superindex indicates the ID of the method with the closest but still significantly smaller result. For instance, FA R@20 = $.2113^{7}$ tells that FA is significantly larger than at least (7) PMPCAV(Medium) R@20 = .1053, as well as significantly larger than all the other methods with R@20 $<$ .1053

ID	Method	nD@20	R@20	P@20	F1@20	UC	SC
1	MPCAV(Subject)	.0091	.0179	.0013	.0022	.9992	.9995
2	MPCAV(Medium)	**.0337**^{8}	**.0658**^{3}	**.0045**^{9}	**.0081**^{3}	**.9996**	**.9998**
3	MPCAV(Style)	.0199^{6}	.0412^{12}	.0031^{4}	.0055^{6}	.9987	.9985
4	MPCAV(Color)	.0135^{12}	.0263	.0021^{12}	.0034^{12}	**.9996**	**.9998**
5	MPCAV(All)	.0171^{4}	.0336^{12}	.0024^{4}	.0043^{4}	**.9996**	**.9998**
6	PMPCAV(Subject)	.0183^{5}	.0366^{4}	.0032^{3}	.0052^{5}	**.2599**	**.4199**
7	PMPCAV(Medium)	.0644^{8}	**.1053**^{2}	.0084^{8}	.0145^{2}	**.2599**	**.4199**
8	PMPCAV(Style)	.0292^{3}	.0542^{3}	.0050^{2}	.0078^{3}	**.2599**	**.4199**
9	PMPCAV(Color)	.0275^{3}	.0404^{4}	.0043^{6}	.0066^{3}	**.2599**	**.4199**
10	PMPCAV(All)	**.0681**^{7}	.1051^{2}	**.0099**^{7}	**.0156**^{7}	**.2599**	**.4199**
11	FA	**.1743**^{10}	**.2113**^{7}	**.0180**^{10}	**.0295**^{10}	**.2599**	**.4199**
12	Random	.0097	.0200	.0015	.0025	1.0000	1.0000

Stat. significance by multiple t-tests, Bonferroni corr.
α_{bonf} = α/n = 0.05/66 = .00076

personalization has the negative side effect of dropping user and session coverages, because of the *user cold start* problem, which is inherent to personalization. From an accuracy standpoint, *PMPCAV(All)* outstands overall: by combining all attributes it achieves top score in almost all metrics among methods based on expert-annotated metadata, with about 400% relative improvement over random.

Favorite Artist (FA). One result that outstands overall is the performance of the artist feature. In this method, we tested whether making personalized recommendations from the user's most frequently purchased artists could yield good results. In effect, our results indicate that FA is the most accurate single method ($nD@20 = 0.1743$, $R@20 = 0.2113$), between 2 to 3 times better than the second best metadata based method – *PMPCAV(All)*.

MPCAV vs PMPCAV. The most important lesson to learn from comparing these methods is the significant performance improvements gained by applying personalization. This is also confirmed by FA and all the other personalized methods we tested (HVF, CNN and Hybrids) which are significantly better than MPCAV, as shown in Table 5.5.

5.4.2. CNN and HVF Visual Features (RQ1.2)

To the best of our knowledge, our work presents the first analysis comparing handcrafted visual features (brightness, contrast, etc.) versus automatically learned features (CNN) for the task of recommending artworks. Table 5.4 presents the results, where it is clear that CNN embeddings yield a significant improvement over HVF features, either combined or in isolation, more than doubling their performance in almost all ranking metrics. These results are in line with the current state-of-the-art of deep neural networks in computer vision, which report better results than other methods in several tasks (Sharif Razavian et al., 2014; R. He & McAuley, 2016).

Individually, the best CNN was ResNet50. Surprisingly the second best CNN was AlexNet, which achieved better results than the rest of the CNNs that were published years later. This indicates than better performance on ImageNet (Deng et al., 2009) does not necessarily transfer into a better performance on a different task, such as artwork recommendation. In addition, the method CNN (All) that combines all CNNs together with a hybrid score achieved the best results. In particular, precision ($P@20 = .0151$) and f1score ($F1@20 = .0248$) were significantly larger than all the other visual methods tested according to the statistical tests.

With respect to HVF methods, individually we observe that *LBP* performs the best (about 300% better than Random). We think this is because LBP is able to capture more fine-grained patterns than Attractiveness features can. Attractiveness features are good enough to perform significantly better than Random nonetheless (more than 200% of improvement). In contrast to the CNN case, however, for HVF features we were not able

Table 5.4. nDCG (nD), Recall (R), Precision (P), F1 Score (F1) for image based methods. User and Session Coverage are all the same for every experiment, UC = .2599 and SC = .4199. The best absolute result of each metric is highlighted. The superindex indicates the ID of the method with the closest but still statistically significant difference. For instance, CNN (All) R@20 = .1702^4 indicates that CNN (All) is significantly larger than (4)CNN (VGG19) R@20 = .1398, as well as significantly larger than all the other methods with R@20 $<$ .1398.

ID	Method	nD@20	R@20	P@20	F1@20
1	CNN (All)	**.1295**^{3}	**.1702**^{4}	**.0151**^{2}	**.0248**^{2}
2	CNN (ResNet50)	.1247^3	.1628^4	.0145^5	.0236^4
3	CNN (AlexNet)	.1081^4	.1461^4	.0135^5	.0216^4
4	CNN (VGG19)	.1008^5	.1398^8	.0124^6	.0205^5
5	CNN (InceptionV3)	.1007^6	.1332^8	.0125^4	.0201^6
6	CNN (NASNet Large)	.0998^8	.1379^8	.0120^8	.0197^7
7	CNN (InceptionResNetV2)	.0932^8	.1300^8	.0119^8	.0192^8
8	HVF (LBP)	.0507^9	.0736^{11}	.0068^9	.0107^9
9	HVF (LBP + Attr.)	.0493^{11}	.0728^{11}	.0064^{10}	.0103^{11}
10	HVF (Attractiveness)	.0407^{11}	.0628^{11}	.0059^{11}	.0095^{11}
11	Random	.0097	.0200	.0015	.0025

Stat. significance by multiple t-tests, Bonferroni corr.
α_{bonf} = α/n = 0.05/55 = .00091.

to find an optimal linear combination that outperformed both LBP and Attractiveness in isolation.

In summary, these results provide evidence in favor of the use of pre-trained deep convolutional neural networks for transfer learning. Their only drawback is the great difficulty in interpreting the neural image embedding in order to explain recommendations to users. Recent works unveil which features are learned by certain neurons (Olah, Mordvintsev, & Schubert, 2017), but knowing whether those features are actually influencing the user towards a purchase decision is still difficult to know. In general terms the features automatically learned by neural networks are discriminating but difficult to explain, and this lack of transparency and explainability might potentially hinder the user acceptance of these recommendations (Konstan & Riedl, 2012; Verbert, Parra, Brusilovsky, & Duval, 2013; Nunes & Jannach, 2017).

5.4.3. Comparing Visual Features vs Metadata (RQ2)

Visual Features vs Curated Attributes. From Table 5.5, which shows results of the overall analysis, we observe that all CNN methods outperformed PMPCAV(All), in particular CNN(All) achieved about 70%-100% relative improvement over PMPCAV(All). However, HVF features did not place very well: in general they did better than MPCAV(Medium), but were significantly outperformed by PMPCAV(All). Based on these results, we conclude that 1) it is possible for a content-based recommender to leverage state-of-the-art CNNs to extract visual features from artwork images and achieve better performance than a baseline based on expert-annotated metadata (the production of which can be very time-consuming), and 2) the HVF features tested (LBP and Attractiveness) are clearly of a lower quality than CNN features, as evidenced by their underperformance when compared to the expert-annotated metadata baseline.

Visual Features vs Favorite Artist (FA). In total contrast to curated attributes, recommending based on the user's favorite artists surprisingly outperforms both HVF and CNN in terms of ranking metrics in the offline evaluation, as can be seen in Table 5.5. In fact, FA ($nD@20 = 0.1743$ and $P@20 = 0.0180$) achieves significantly better metrics than the best CNN method ($nD@20 = 0.1295$ and $P@20 = 0.0151$) by more than 20% to 34%. These offline results may be explained by the fact that users are probably biased to keep exploring and finding items they like from artists they are already familiar with. However, when we look at the online results with expert curators (Table 5.7), the differences between FA and visual methods become much narrower, where in fact CNN and the hybrid CNN+HVF show better results than FA in practically all metrics. We believe this shows that FA is a very good heuristic for filtering the item search space when predicting next purchases (as reflected offline), but its lack of visual content awareness renders it incapable of performing fine-grained visual discrimination, which is reflected in the less favorable results in the online evaluation compared to visual methods[3].

[3]When we carried out the online evaluation with expert curators, AlexNet was the only CNN we used because we had not yet tested other CNNs at the time.

Table 5.5. nDCG (nD), Recall (R), Precision (P), F1 Score (F1), User Coverage (UC) and Session Coverage (SC) for all content-based methods. The best three absolute results of each metric are highlighted. The superindex indicates the ID of the method with the closest but still significantly smaller result. For instance, Hybrid_1 R@20 = $.2362^2$ tells that Hybrid_1 is significantly larger than (2)Hybrid_2 R@20 = .2285, as well as significantly larger than all the other methods with R@20 $<$ .2285.

ID	Method	nD@20	R@20	P@20	F1@20	UC	SC
1	Hybrid_1 (FA+CNN+PMPCAV)	**.1790**^{2}	**.2362**^{3}	**.0201**^{2}	**.0333**^{2}	**.2599**	**.4199**
2	Hybrid_2 (FA+CNN)	**.1759**^{5}	**.2309**^{5}	**.0197**^{5}	**.0325**^{5}	**.2599**	**.4199**
3	Hybrid_3 (FA+PMPCAV)	.1731^{5}	**.2228**^{5}	**.0190**^{5}	**.0312**^{4}	**.2599**	**.4199**
4	FA	**.1743**^{5}	.2113^{5}	.0180^{5}	.0295^{5}	**.2599**	**.4199**
5	CNN (All)	.1295^{8}	.1702^{8}	.0151^{6}	.0248^{6}	**.2599**	**.4199**
6	CNN (ResNet50)	.1247^{7}	.1628^{8}	.0145^{9}	.0236^{8}	**.2599**	**.4199**
7	CNN (AlexNet)	.1081^{8}	.1461^{8}	.0135^{9}	.0216^{8}	**.2599**	**.4199**
8	CNN (VGG19)	.1008^{9}	.1398^{13}	.0124^{10}	.0205^{9}	**.2599**	**.4199**
9	CNN (InceptionV3)	.1007^{10}	.1332^{13}	.0125^{8}	.0201^{10}	**.2599**	**.4199**
10	CNN (NASNet Large)	.0998^{13}	.1379^{13}	.0120^{12}	.0197^{12}	**.2599**	**.4199**
11	CNN (InceptionResNetV2)	.0932^{13}	.1300^{13}	.0119^{13}	.0192^{13}	**.2599**	**.4199**
12	PMPCAV(All)	.0681^{13}	.1051^{13}	.0099^{13}	.0156^{13}	**.2599**	**.4199**
13	HVF (LBP)	.0507^{14}	.0736^{17}	.0068^{14}	.0107^{14}	**.2599**	**.4199**
14	HVF (LBP + Attr.)	.0493^{16}	.0728^{17}	.0064^{15}	.0103^{16}	**.2599**	**.4199**
15	HVF (Attractiveness)	.0407^{16}	.0628^{17}	.0059^{16}	.0095^{17}	**.2599**	**.4199**
16	MPCAV(Medium)	.0337^{17}	.0658^{17}	.0045^{17}	.0081^{17}	**.9996**	**.9998**
17	Random	.0097	.0200	.0015	.0025	**1.0000**	**1.0000**

Statistical significance was obtained using multiple pairwise t-tests with Bonferroni correction, $\alpha_{bonf} = \alpha/n$ = 0.05/136 = .00037

5.4.4. Hybrid recommendations (RQ3)

The Hybrid recommenders, summarized in Table 5.5, show a clear tendency: when features are combined into hybrids, they tend to perform better than the features used individually. Some of these improvements are statistically significant, as in the case of hybrid 1 which is significantly better than FA, CNN(All) and PMPCAV(All) individually and in fact significantly better than all other methods in (almost) all accuracy metrics. In other cases, as in hybrids 2 and 3, we observe improvements over individual features as well but in relation to FA these are not always statistically significant.

In order to get more insights into the performance of methods for different users, we also plotted the Precision@20 achieved by several methods at different profile size ranges.

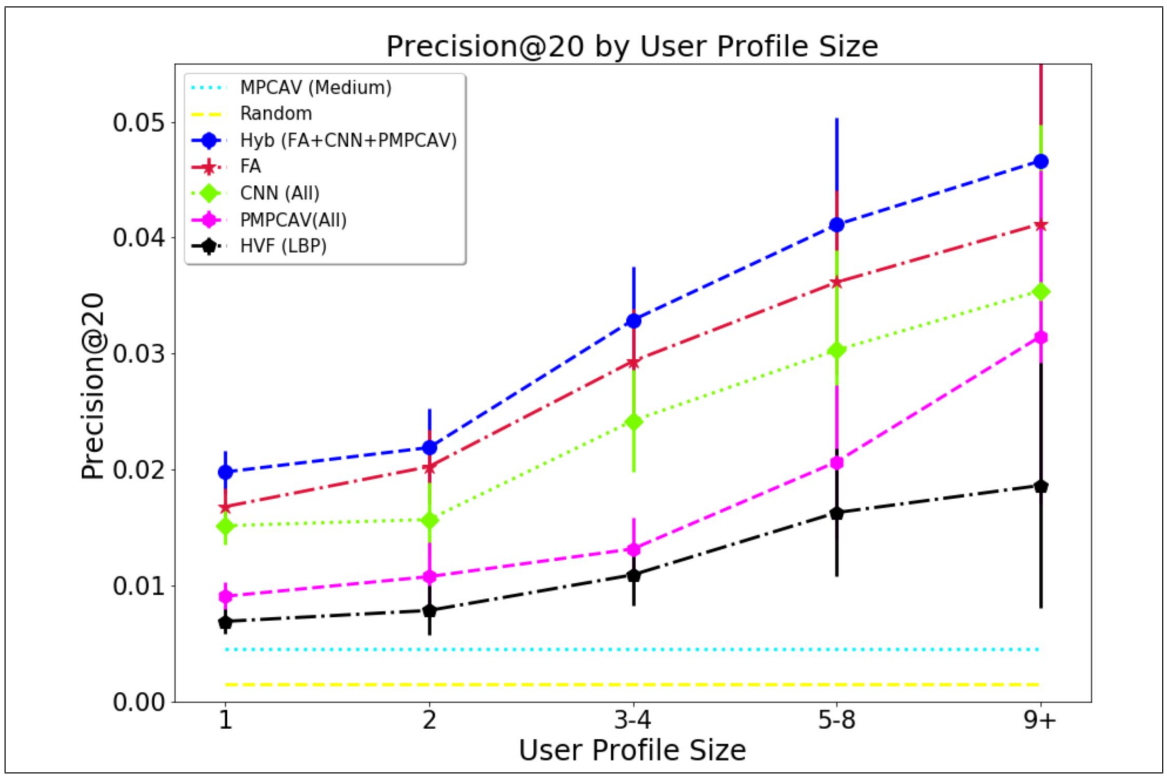

Figure 5.5. Precision@20 of different methods at different user profile sizes.

As shown in Figure 5.5, an important observation to highlight is the fact that the ranking of methods remains stable for all profile size ranges, with Hybrid 1 always at the top of the ranking. We can also observe a trend for all methods to improve their performance with bigger profile sizes. This seems to indicate that the different methods are able to better capture user tastes as they get more user feedback.

5.4.5. Effect on Diversity (RQ2 and RQ3)

Table 5.6 presents the results of several features and combinations of them upon six metrics of diversity. F1@20 is also reported in the table as a reminder of the ranking performance of each method. The results can be summarized as follows: In terms of visual diversity, we can clearly see the effect of CNN: all methods that use CNN features show lower visual diversities than those that do not. This is an expectable result, as pre-trained CNNs are powerful off-the-shelf tools that can map similar images to locally close vectors

Table 5.6. Diversity metrics for all content-based methods. F1score@20 (F1@20) is also included as a reminder of the accuracy of each method. The three smallest (underline) and largest (bold) results for each diversity metric are highlighted.

ID	Method	F1@20	D@10 visual cluster	D@10 visual pairwise	D@10 artist	D@10 jaccard pairwise	D@10 color	D@10 medium
1	Hybrid_1 (FA+CNN+PMPCAV)	**.0333**^{2}	10.0697^{4}	.3952^{2}	8.4375^{3}	.7433^{2}	11.7362^{12}	2.2719^{3}
2	Hybrid_2 (FA+CNN)	**.0325**^{5}	9.1883	.3803	7.6165^{4}	.7730^{1}	12.0959^{3}	2.7902^{4}
3	Hybrid_3 (FA+PMPCAV)	**.0312**^{4}	11.8327^{9}	.4297^{9}	7.8472^{2}	.7214^{4}	11.8309^{12}	2.0459^{12}
4	FA	.0295^{5}	9.7124^{2}	.4092^{8}	2.8809	.7068	11.9983^{3}	2.3864^{1}
5	CNN (All)	.0248^{6}	9.6688^{2}	.3913^{2}	12.6822^{1}	.8488^{16}	12.6514^{7}	3.3951^{2}
6	CNN (ResNet50)	.0236^{8}	10.1429^{4}	.3968^{7}	12.6804^{1}	.8524^{5}	12.7164^{7}	3.4399^{2}
7	CNN (AlexNet)	.0216^{8}	10.1732^{4}	.3923^{2}	13.0314^{5}	.8502^{16}	12.4317^{2}	3.5119^{5}
8	CNN (VGG19)	.0205^{9}	10.6845^{7}	.4016^{6}	14.3341^{11}	.8648^{6}	13.0546^{15}	3.5386^{6}
9	CNN (InceptionV3)	.0201^{10}	11.2208^{8}	.4195^{11}	13.8768^{7}	.8712^{8}	13.1360^{15}	3.6926^{11}
10	CNN (NASNet Large)	.0197^{12}	11.0767^{8}	.4144^{8}	14.0180^{16}	.8697^{8}	13.1435^{15}	3.6827^{8}
11	CNN (InceptionResNetV2)	.0192^{13}	11.1313^{8}	.4151^{4}	14.0232^{16}	.8703^{8}	13.1871^{15}	3.6072^{6}
12	PMPCAV(All)	.0156^{13}	13.6607^{3}	.4498^{3}	14.4608^{11}	.7429^{3}	11.0691	1.8303^{16}
13	HVF (LBP)	.0107^{14}	14.6874^{12}	.4667^{12}	15.8733^{12}	**.8949**^{15}	13.9820^{11}	**4.1296**^{9}
14	HVF (LBP + Attr.)	.0103^{16}	15.3969^{13}	.4732^{13}	**16.3359**^{13}	**.8961**^{15}	**14.0628**^{11}	**4.0633**^{9}
15	HVF (Attractiveness)	.0095^{17}	**15.4358**^{13}	**.4743**^{13}	**16.5584**^{13}	.8850^{9}	12.8210^{7}	4.0569^{9}
16	MPCAV(Medium)	.0081^{17}	**15.4375**^{13}	**.4829**^{15}	13.7440^{7}	.7844^{2}	**14.3841**^{14}	1.0017
17	Random	.0025	**17.4006**^{16}	**.4972**^{16}	**18.4069**^{15}	**.9123**^{14}	**14.2869**^{14}	**4.5804**^{13}

Statistical significance was obtained using multiple pairwise t-tests with Bonferroni correction, $\alpha_{bonf} = \alpha/n = 0.05/136 = .00037$.

in an image embedding space, which leads to more visually similar (and therefore less diverse) recommendations. There is a notable exception nonetheless. FA, which recommends by sampling artworks from the user's favorite artists, shows either lower or comparable values of both visual cluster diversity and visual pairwise diversity with respect to CNN based methods. This result indicates that artists in our dataset paint visually similar artworks, making recommendation lists based on the same artist less visually diverse compared to other methods.

Moreover, when FA and CNN(All) are combined, as in Hybrid_2, the resulting recommender achieves the lowest visual diversities ($\overset{\text{D@20}}{\underset{\text{cluster}}{\text{visual}}} = 9.1883$, $\overset{\text{D@20}}{\underset{\text{pairwise}}{\text{visual}}} = .3803$) and better predictive accuracy than each individual method in isolation, providing evidence that users are more likely to purchase similar-looking artworks from artists they are familiar with, although recommending based on this heuristic can lead to the lowest visual diversity.

Regarding diversity metrics based on metadata, the most informative metric is Artist Diversity ($^{\text{D@20}}_{\text{artist}}$). From this metric we can notice a very interesting trend: the fewer artists used in a recommendation, the more accurate the recommendation becomes. This trend holds until we get to FA, with the lowest artist diversity ($^{\text{D@20}}_{\text{artist}} = 2.9$ approx.). However, the trend gets reversed when we get to the top 3 hybrid recommenders, all of them recommending from about 7-8 artists on average. This seems to indicate the existence of an optimal level of artist diversity in order to achieve higher recommendation accuracy. This result is also good news from a business standpoint: the top hybrid recommenders can achieve higher accuracy while still being able to promote paintings from a reasonably diverse group of artists. With respect to Color ($^{\text{D@10}}_{\text{color}}$) and Medium ($^{\text{D@10}}_{\text{medium}}$) diversities, these metrics do not reveal very insightful patterns, besides the fact that the lowest values are reached by MPCAV(Medium), PMPCAV(All) and hybrids that include the latter. On the contrary, Jaccard Pairwise Diversity ($^{\text{D@10}}_{\substack{\text{jaccard}\\\text{pairwise}}}$) do seem to show a pattern similar to Artist Diversity, although the apparent correlation is probably due to the influence of the artist in the bag of attributes used for Jaccard Index calculations.

5.4.6. Validation with Expert Users (RQ4)

Table 5.7 presents the results of the online evaluation with 8 expert curators from *UGallery*, showing the mean over four metrics: nDCG@5, nDCG@10, Precision@5 and Precision@10. As explained in Section 5.3.2, each curator had to rate 10 recommended items from each of the five methods shown in Table 5.7 (i.e., they rated 50 items in total). The most important aspect to highlight is that combining FA with visual features in a single hybrid (FA+CNN+HVF) outperforms all the other features, either hybrid or single, in all four metrics, which is consistent with the offline results. Another interesting result is that CNN shows better performance than FA, which is the opposite to the offline evaluation. We think that this might be due to the lack of diversity that FA promotes, but also to the potential noise present when sampling artworks from artists to fit a top-n recommendation without awareness of the visual content. It is also remarkable that the isolated features show smaller differences between them in this user experiment than in

Table 5.7. nD@5, nD@10, P@5 and P@10 for algorithms tested with 8 UGallery experts. For nD@k, all ratings $<= 3$ were set to 0. For P@k, only ratings ≥ 4 were regarded as relevant.

Name	**nD@5**	**nD@10**	**P@5**	**P@10**
Hybrid(FA+CNN+HVF)	**0.9042**	**0.8913**	**0.7500**	**0.6750**
Hybrid(CNN+HVF)	0.6747	0.6638	0.5000	0.4250
CNN	0.7176	0.6947	0.5000	0.4000
FA	0.4276	0.5662	0.3000	0.4000
HVF	0.5498	0.5314	0.3500	0.2625

the offline evaluation. In terms of nDCG@5, nDCG@10 and Precision@5, CNN seems to outperform both FA and HVF, while it has the same performance as FA in terms of Precision@10. Given the small sample size, we cannot report tests of statistical significance, but the trend of results points toward implementing a hybrid recommender with FA and visual features for the best performance without hindering diversity.

5.5. Summary & Discussion

The main findings with respect to our RQs can be summarized as follows:

- **RQ1. Metadata**: In general, using the most popular curated attribute values (MPCAV) brings a significant improvement over random prediction, although the improvement is rather small. The personalized version PMPCAV, specially the one using all attributes, performed much better, confirming that personalizing is key. But most notably, just recommending based on a user's favorite artists produced very high accuracy metrics, the best among metadata-based methods.
- **RQ1. Visual Features**: The features automatically obtained from pre-trained convolutional neural networks (CNN) significantly outperformed handcrafted visual features (HVF). This result is consistent with the fact that CNNs are the

current state of the art in computer vision. This also supports the use of transfer learning, i.e. leveraging CNNs pre-trained for a different task, e.g. object classification on ImageNet, for the task of artwork recommendation.

- **RQ2. Visual Features vs. metadata**: Visual features obtained with CNNs performed better than curated attributes. This is an important result, since it points towards using features extracted directly from the images rather than spending resources in manually tagging the images. However, the best predictive single feature overall was Favorite Artist, so combining the strengths of both visual features and FA seems like a promising approach.
- **RQ3. Hybrids**: Hybrid methods combining multiple features outperformed individual features. The hybrid method which combined FA, CNN and PMPCAV produced the best offline results (a variant that only combined FA and CNN performed almost equally well).
- **RQ4. Expert online evaluation**: In the online evaluation, the best performance was achieved by the hybrid (FA+CNN+HVF), confirming that enhancing FA with visual information can produce good personalized recommendations. Also noteworthy were the narrower differences among isolated features (CNN, HVF, FA) compared to their offline results and the fact that CNN outperformed FA in the online setting.

Taken together, the results presented in this section show that a recommender system which utilizes several types of content could indeed support people who buy artworks online based on their personal taste. In particular, we found that the two most important content properties to take into account when building a content-based artwork recommender are 1) the visual information from images, which can be better exploited by deep convolutional neural networks, and 2) the artists that created the artworks. In contrast, expert-annotated metadata can be useful to a certain extent, but it would require a constant effort by humans to get all artworks properly and consistently tagged, which can be very

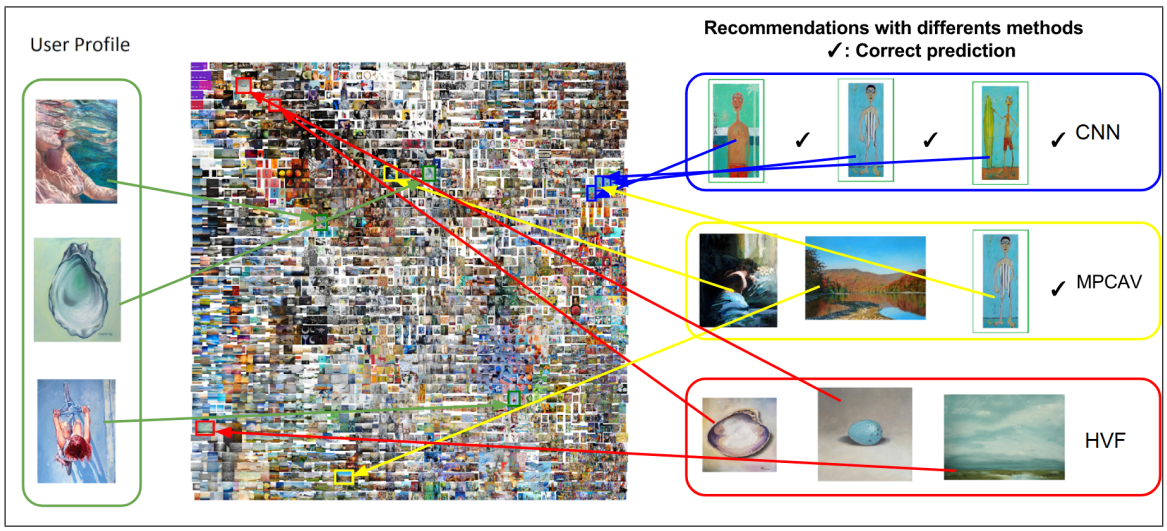

Figure 5.6. t-SNE map of the CNN image embedding displaying paintings of an anonymized user profile (green), and recommendations contextualized with three methods: CNN (AlexNet) (blue), MPCAV (yellow) and HVF (Attractiveness) (red). Check marks indicate correct predictions.

exhausting, and as our results have shown expert-annotated metadata did not perform as well as CNN features and artists, which are always available.

In this section we have presented evidence that deep convolutional neural networks can be of great value in the field of personalized artwork recommendations, since they decrease the cost of domain expert knowledge to identify the visual features which can be most successful, with a small compromise on diversity. However, in order to make recommendations really useful and not only persuasive (Tintarev & Masthoff, 2015), researchers and developers need to make sure that people can inspect and have a sense of control (Knijnenburg, Bostandjiev, O'Donovan, & Kobsa, 2012; Parra & Brusilovsky, 2015). One way to provide users with such control is to use techniques such as t-SNE with an interactive interface. t-SNE (Maaten & Hinton, 2008) is a dimensionality reduction technique commonly used to visualize what CNN embeddings might encode (R. He et al., 2016; R. He & McAuley, 2016; Nguyen, Yosinski, & Clune, 2016). This technique could be used to help users visualize high-dimensional data in a lower-dimensional space in order to understand recommendations, explore them and inspect them, features associated with

improved user satisfaction (Knijnenburg et al., 2012; Verbert et al., 2013). For instance, Figure 5.6 uses t-SNE to reduce CNN embeddings and then display an anonymized user profile and the images predicted by three different methods: CNN (AlexNet), MPCAV and HVF (Attractiveness). We foresee building rich visual applications providing user control, transparency and explainability, important characteristics to build user trust and acceptance in recommendations (Tintarev & Masthoff, 2015; Ekstrand, Kluver, Harper, & Konstan, 2015).

An important aspect to bear in mind regarding the methods presented in this section is the fact that they are entirely content-based and heuristic, in the sense that we build local user profiles based on the contents of items consumed by each individual user, and we use heuristics to aggregate the information and make personalized recommendations, such as e.g. favorite artist, maximum cosine similarity, combining scores through simple linear combinations, etc. However, it's very likely that an optimal recommendation algorithm would exploit non-obvious patterns that would be very hard to capture through human-designed heuristics. These patterns are usually present in the implicit feedback of users. Therefore, one possible avenue for improvement is to learn a recommendation algorithm from the purchase history of users directly. In effect, this is what collaborative filtering usually does: it leverages the implicit wisdom of the crowd by learning a model that considers the collective feedback of all users in the system (expressed in e.g. likes, purchases, comments, watches, etc.) so that the learned model captures implicit, non-obvious consumption patterns, which then can be used to make novel recommendations to users. However, in the introduction we already mentioned that the *one-of-a-kind* setting we are dealing with is not suitable for the application of traditional collaborative filtering techniques that depend on multiple user-item co-occurrences for their success. Fortunately, we can work around this difficulty by using items' content to embed items into a latent content space, in which we can exploit co-occurrence patterns and apply collaborative filtering. We will elaborate this idea in the next section.

6. SECOND APPROACH: HYBRID CONTENT-COLLABORATIVE RECOMMENDATION

6.1. Previous relevant work on hybrid content-collaborative recommendation

Before introducing our proposed model (section 6.2), we will first review two previous works that have served us as inspiration.

VBPR: Visual Bayesian Personalized Ranking. VBPR (R. He & McAuley, 2016) is a recommendation method that incorporates visual information obtained from deep CNNs (content) into the traditional Matrix Factorization model for collaborative filtering (Koren, Bell, & Volinsky, 2009). Concretely, given a user u and an item i, VBPR proposes to model the preference of user u for item i as:

$$\hat{x}_{u,i} = \beta_i + \gamma_u^T \gamma_i + \theta_u^T (E f_i) + \beta'^T f_i \tag{6.1}$$

, where β_i is item i's bias term (1×1), γ_u and γ_i are the latent factors of user u and item i ($K \times 1$), θ_u is user u's visual factors ($D \times 1$), E is a visual projection matrix ($D \times F$), f_i is the deep CNN visual feature vector of item i ($F \times 1$) and β' is a global visual bias vector ($F \times 1$). The term $\gamma_u^T \gamma_i$ models user u's preference for item i's latent factors, the term $\theta_u^T (E f_i)$ models user u's preference for item i's projected visual features $E f_i$, and the term $\beta'^T f_i$ models users' overall opinion toward the visual appearance of item i.

For model learning, VBPR uses Bayesian Personalized Ranking (BPR), a pairwise ranking optimization framework (Rendle et al., 2009). In BPR, a training set D_S consisting of triples (u, i, j) is defined, where each triple means that user u (very likely) prefers item i over item j. In positive-only feedback settings, where users only provide positive feedback for items they probably like (e.g. purchases, views) but no explicit negative feedback for items they don't like, BPR suggests defining the training set D_S as follows:

$$D_S = \{(u, i, j)|u \in U \wedge i \in I_u^+ \wedge j \in I \setminus I_u^+\} \tag{6.2}$$

, where U stands for the set of all users, I for the set of all items and I_u^+ for the set of items for which user u has shown evidence of positive interest (e.g. purchased items). The intuition is that any user u would probably prefer items in I_u^+ over items in $I \setminus I_u^+$. Given this dataset D_S, the following optimization criterion is used for personalized ranking (BPR-OPT):

$$\sum_{(u,i,j) \in D_S} ln(\sigma(\hat{x}_{uij}(\Theta))) - \lambda_\Theta ||\Theta||^2 \tag{6.3}$$

where Θ is the model's parameter vector, $\hat{x}_{uij}(\Theta)$ denotes a model specific function of Θ that estimates how much user u prefers item i over item j, σ is the logistic (sigmoid) function and λ_Θ is a model specific regularization hyperparameter.

When using Matrix Factorization as the preference predictor (i.e. BPR-MF), which is the case of VBPR, $\hat{x}_{uij}(\Theta)$ is commonly defined as:

$$\hat{x}_{uij}(\Theta) = \hat{x}_{u,i} - \hat{x}_{u,j} \tag{6.4}$$

where $\hat{x}_{u,i}$ and $\hat{x}_{u,j}$ are defined by Eq. 6.1. Since the training set D_S can be intractably large, the usual approach is to learn BPR-MF efficiently via stochastic gradient descent, meaning that the gradient of BPR-OPT (Eq. 6.3) can be approximated by computing the gradient for a mini-batch of triples randomly sampled from D_S at each gradient descent step.

Deep Neural Networks for Youtube Recommendations. As stated by Covington et al. (2016), "Youtube represents one of the largest scale and most sophisticated industrial recommendation systems in existence". In their RecSys 2016 paper, the authors revealed the overall architecture of Youtube's state-of-the-art recommender system based

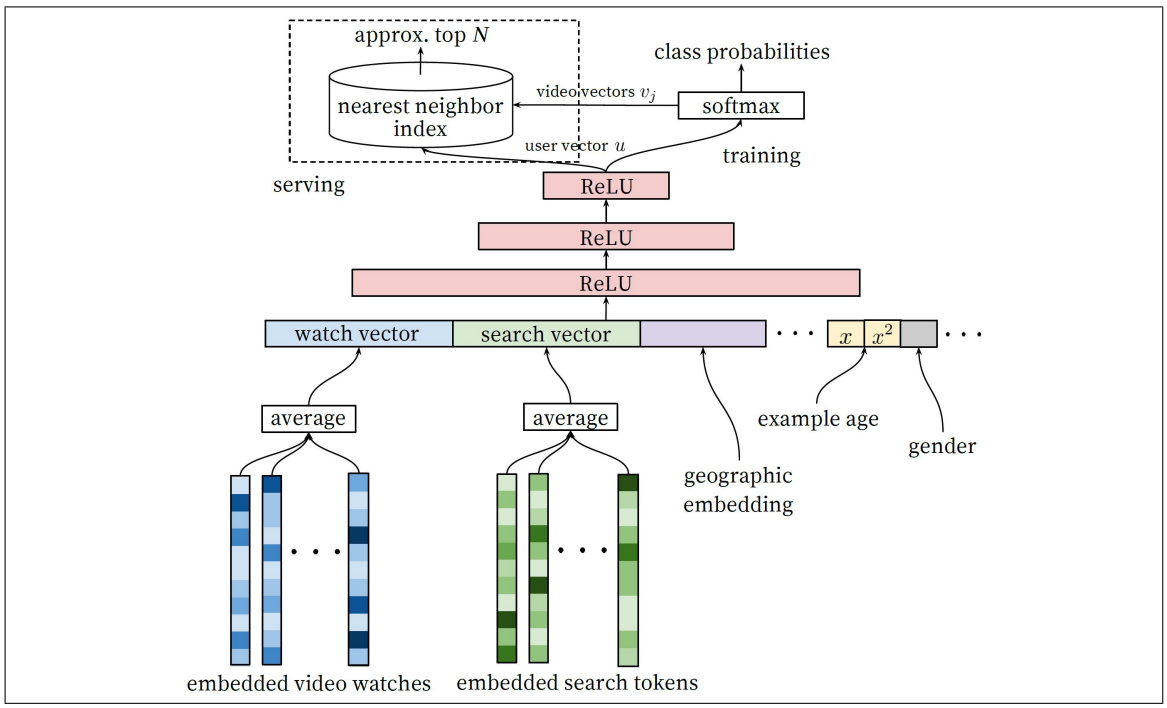

Figure 6.1. Architecture of Youtube's candidate generation neural network according to Covington et al. (2016)

on Deep Neural Networks, which brought dramatic performance improvements over past versions. Their recommendation architecture consists of two stages: deep candidate generation (which filters hundreds of candidate videos from a corpus of millions) followed by a deep ranking model. The model we propose in section 6.2 is mainly inspired by the deep candidate generation neural network of the first stage, although the deep ranking neural network has a similar architecture.

As shown in Figure 6.1, the candidate generation network displays several interesting characteristics: 1) It follows the traditional approach of Matrix Factorization, in the sense that users and items are modeled as vectors in a latent embedding space and user-item preferences are computed via dot product, i.e. $\hat{x}_{u,i} = \vec{u} \cdot \vec{i}$. 2) Instead of learning a separate latent vector for each user, the network generates the user vector by aggregating information about the watch and search history of the user along with contextual information such as geographic location, age, gender, etc. The information is processed by multiple fully

connected layers with ReLU (Nair & Hinton, 2010) activations until the final user vector is produced. This allows the network to generalize and make personalized recommendations to any user very quickly as soon as he or she starts interacting with the website, avoiding altogether the need to define and train a separate set of variables per user —which can be both memory and computation inefficient, especially when dealing with hundreds of millions of users.

Youtube's Recommender System is a clear example of combining content and collaborative information. The neural networks in Youtube are trained considering the collective feedback of millions of users over time, but also incorporating context metadata about users (age, location, gender, etc.) as well as latent embedding vectors of video watches and search queries in the user history. The neural network used in the ranking stage is even more exhaustive and fine-grained in terms of the features it uses, but the candidate generation network (Fig. 6.1) should suffice to illustrate the point.

6.2. Proposed model: YT-VBPR

Inspired by the works reviewed in the previous section, we developed a new deep neural network model that combines content and collaborative information for artwork recommendation. We call it Youtube-like Visual Bayesian Personalized Ranking, or in short, YT-VBPR. The model architecture is illustrated in Figure 6.2. The core idea behind the network design was the goal of building a single neural network with a fixed number of parameters capable of embedding any user and any item into a latent space in which dot-product user-item preferences can be computed, based entirely on visual content. In that sense, once trained YT-VBPR is entirely content-based because it only needs images as input: the network can output 1) an item vector if given the item's image and 2) a user vector if given the images of items with positive feedback in the user's history.

This design 1) allows easy generalization to new items and users without further training and 2) dramatically alleviates the cold-start problem present in *one-of-a-kind* artwork

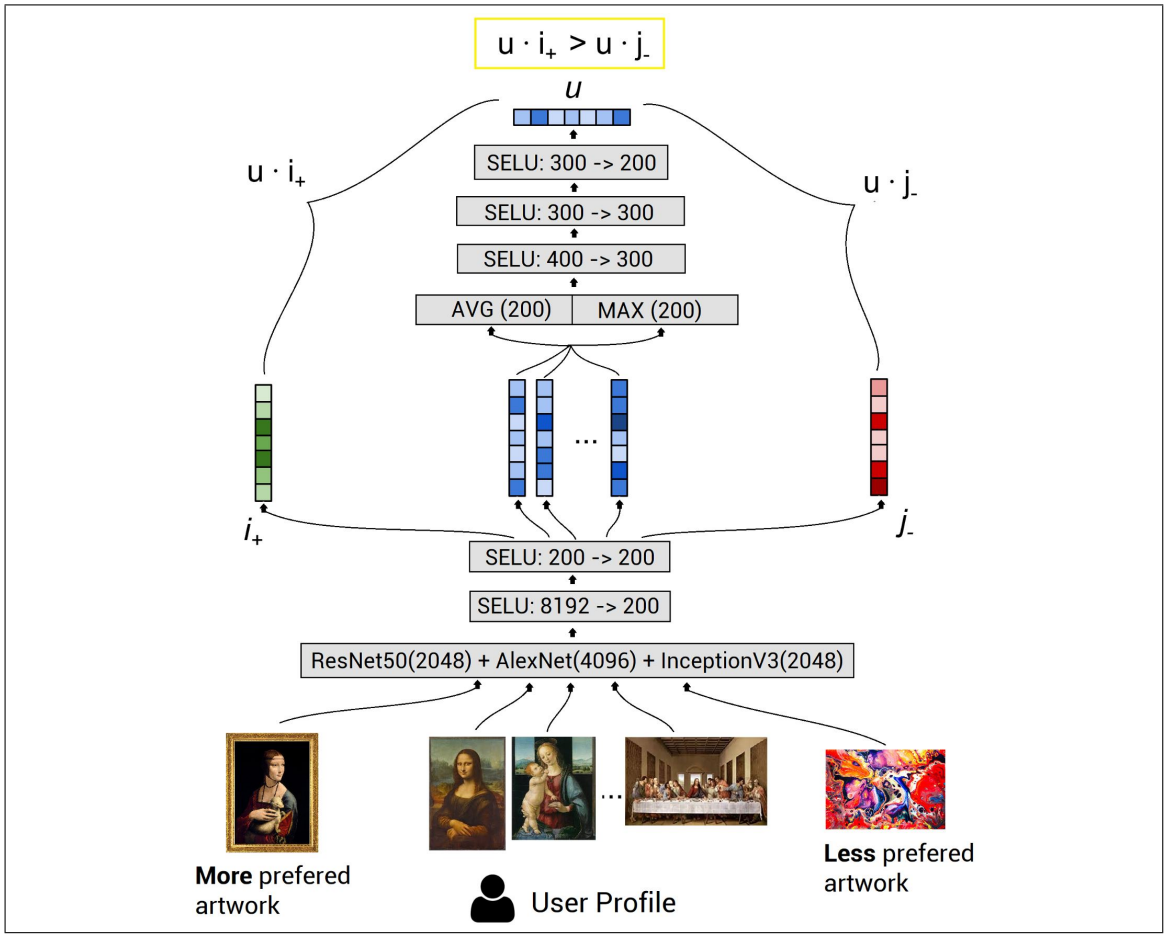

Figure 6.2. Architecture of YT-VBPR

galleries. As soon as an item gets uploaded to the system, its image can immediately be processed, indexed and become available for recommendation to users. Likewise, since the neural network only needs images to produce a user vector, it's possible to make responsive in-session recommendations for cold-start users based on viewed images as well as based on the purchase history for warm-start users (a property that comes directly from the network's Youtube-like design). Furthermore, since the network's input is just images, during training it can learn patterns in the image embedding space even in a *one-of-a-kind* setting where each image is purchased by a single user at most.

The first step to accomplish all of this is to extract visual features from images. In our experiments we combined 3 CNNs: ResNet50, AlexNet and InceptionV3, since these

networks achieved top performance among all CNNs we tested, as already seen in table 5.4[1]. We extract features from each CNN and concatenate them into a $\mathbb{R}^{8192}$ vector. Then 2 fully-connected layers with SELU[2] activations reduce the item embedding dimension to $\mathbb{R}^{200}$. This is illustrated in the lower half of Fig. 6.2. The second step is to compute the user embedding vector. This is done by aggregating the $\mathbb{R}^{200}$ embedding vectors of the user's purchased items. We do this aggregation by computing and concatenating the average pooling and maximum pooling, and then apply 3 additional fully-connected layers with SELU activations to ouput a final $\mathbb{R}^{200}$ user embedding vector. This second step is illustrated in the upper half of Fig. 6.2. Finally, the dot product between user and item vectors models user-item preference scores.

6.3. YT-VBPR Model Training

We showed in section 5.4 that the favorite artist (FA) was a very good heuristic to find relevant artworks for users. Moreover, the hybrid combining FA + CNN(All) + PMP-CAV(All) achieved top performance among content-based methods. However, based on the description of YT-VBPR we have presented so far it is not clear how FA or PMP-CAV(All) are used or whether metadata features are used at all. These questions should become clear once we understand how YT-VBPR is trained. Inspired by the Bayesian Personalized Ranking (BPR) optimization framework (Rendle et al., 2009), we define a training set D_S of triples (p, i, j), meaning that a user with profile (history of items consumed) p should (most likely) prefer item i over item j, i.e. $\vec{u_p} \cdot \vec{i} > \vec{u_p} \cdot \vec{j}$. Notice that instead of concrete users u, we use profiles (set of items) p. This is because YT-VBPR does not learn user-specific variables, instead it learns the parameters of a general network that outputs user vectors from item consumption histories. This formulation adds more flexibility to the process of generating training triples (p, i, j). For example, given the whole purchase history of a user u, one can sample many profiles p from the same user,

[1]Strictly speaking VGG19 did slightly better than InceptionV3, but we chose InceptionV3 because its output (2048) is half the size of VGG19's (4096)

[2]SELU stands for Scaled Exponential Linear Unit, a new activation function with self-normalizing properties (Klambauer, Unterthiner, Mayr, & Hochreiter, 2017)

Table 6.1. Symbols used in the definition of YT-VBPR training sets

Symbol	Description
U, I	user set, item set
u, i, j	a specific user, positive item and negative item (resp.)
T_u	set of all timestamps in which user u made a purchase
N_u	total number of purchase baskets of user u in the dataset
I_u^+	set of all items purchased by user u in total (full history)
$I_{u,t}^+$	set of all items purchased by user u up to instant t
$I_{u,k}^+$	set of all items purchased by user u up to his k-th purchase basket (inclusive)
$P_{u,k}$	set of all items purchased by user u in his k-th purchase basket
A_u	set of all artists user u has purchased an item from (full history)
$A_{u,t}$	set of all artists user u has purchased an item from up to instant t
VC_u	set of all visual clusters user u has purchased an item from (full history)
$VC_{u,t}$	set of all visual clusters user u has purchased an item from up to instant t
a_i	the artist (creator) of item i
vc_i	the visual cluster of item i

representing e.g. the different states of user u's history over time. It is even possible to generate fictitious user profiles if one wants to.

Concretely, we generate the training set D_S as the union of multiple (almost) disjoint[3] training sets, each one generated with a different strategy in mind. These strategies and their corresponding training sets are described below. Table 6.1 summarizes the notations used in this section.

(i) **Rank profile above all.** Given a user u who has purchased the items I_u^+, each item $i \in I_u^+$ should be ranked above any item $j \notin I_u^+$ (outside u's full history). Formally:

$$D_S^1 = \{(I_u^+, i, j) \mid u \in U \wedge i \in I_u^+ \wedge j \in I \setminus I_u^+\} \tag{6.5}$$

[3]Theoretically these training sets are not perfectly disjoint, but in practice we hash all training triples and make sure no two training triples have the same hash. This prevents duplicate training triples from being added to the final training set.

The intuition is that if user u has purchased items I_u^+, then we can assume with strong confidence that he likes items in I_u^+, and moreover, we can conservatively assume that he is more likely to prefer them over everything else.

(ii) **Rank profile above all (artificial single-item profiles)**. Given an artificial profile of a single item i, that same item i should be ranked above anything else. Formally:

$$D_S^2 = \{(\{i\}, i, j) \mid i \in I \ \wedge \ j \in I \setminus \{i\}\} \tag{6.6}$$

Since our dataset is not very large, we realized we could make the training more robust if we include a wide spectrum of easy artificial training cases to make sure the network learns parameters that generalize well (so they do not overfit to user histories).

(iii) **Recommending visually similar artworks from favorite artists.** Given a user u who has purchased the items I_u^+, a non-purchased item $i \notin I_u^+$ that shares artist a_i and visual cluster vc_i with items in I_u^+ should be ranked above any item $j \notin I_u^+$ as long as j does not belong to a visual cluster or artist that user u likes. Formally:

$$\begin{aligned} D_S^3 = \{(I_u^+, i, j) \mid u \in U \wedge \\ i \in I \setminus I_u^+ \ \wedge \ a_i \in A_u \ \wedge \ vc_i \in VC_u \wedge \\ j \in I \setminus I_u^+ \ \wedge a_j \notin A_u \ \wedge vc_j \notin VC_u\} \end{aligned} \tag{6.7}$$

The intuition in this case is that if an non-purchased item belongs to both an artist and visual cluster user u has liked in the past, then there is a very high chance that he will like such an item too. In other words, we are telling the network to pay attention to artworks visually similar from favorite artists, because these artworks can be very good recommendations for user u —a heuristic inspired by the hybrid FA + CNN(All) that achieved good results in the content-based approach (Table 5.5).

(iv) **Recommending visually similar artworks from favorite artists (artificial single-item profile)**. Given an artificial profile of a single item i', an item i sharing artist with i' should be ranked above any item j not sharing artist with i' as long as i is also more similar to i' than j by a minimum margin. Formally:

$$\begin{aligned} D_S^4 = \{(\{i'\}, i, j) \mid i' \in I \wedge \\ i \in I \setminus \{i'\} \wedge a_i = a_{i'} \wedge \\ j \in I \setminus \{i'\} \wedge a_j \neq a_{i'} \wedge \\ vsim(i', i) > vsim(i', j) + m\} \end{aligned} \tag{6.8}$$

This last training set is just the application of the FA+CNN heuristic to artificial single-item profiles.

(v) **Predicting next purchase basket.** Given a user u who has purchased the items $I_{u,k}^+$ up to his k-th purchase basket, an item i in u's next purchase basket $P_{u,k+1}$ should be ranked above any item $j \notin I_u^+$ as long as j does not belong to a visual cluster or artist that user u likes. Formally:

$$\begin{aligned} D_S^5 = \{(I_{u,k}^+, i, j) \mid u \in U \wedge 0 \leq k < N_u - 1 \wedge \\ i \in P_{u,k+1} \wedge \\ j \in I \setminus I_u^+ \wedge a_j \notin A_u \wedge vc_j \notin VC_u\} \end{aligned} \tag{6.9}$$

The goal of D_S^5 is to teach the network to predict future purchases, i.e. given items purchased in the past, it should try to push up in the ranking items that will be purchased next. Notice that only items of the immediately next purchase basket are predicted, but not the items of purchase baskets further in the future. We do this because intuitively predicting future baskets given the past becomes increasingly harder if the elapsed time is longer, to the point that the prediction can be almost impossible. Therefore, forcing the network to blindly learn future predictions can potentially introduce noise to the training. We minimize this risk by only predicting the immediately next purchase basket.

(vi) **Predicting hidden item in k-th purchase basket**. Given a user u who has purchased items $I^+_{u,k}$ up to his k-th purchase basket (inclusive), with $|P^+_{u,k}| \geq 2$, if we hide an item $i \in P_{u,k}$ and use the rest as profile, then i should be ranked above any item $j \notin I^+_u$ as long as j does not belong to a visual cluster or artist that user u likes. Formally:

$$D^6_S = \{(I^+_{u,k} \setminus \{i\}, i, j) \mid u \in U \ \wedge \ 0 \leq k < N_u \ \wedge |P_{u,k}| \geq 2 \wedge \\ i \in P_{u,k} \wedge \\ j \in I \setminus I^+_u \ \wedge a_j \notin A_u \ \wedge vc_j \notin VC_u\} \tag{6.10}$$

This training set is different kind of next purchase prediction, because the network has access to the user's full purchase history (excepting the item i which is hidden) instead of the previous purchase basket only. It gives more context to predict the hidden item i.

Finally, the training set D_S is formally defined as:

$$D_S = \bigcup_{i=1}^{6} D^i_S \tag{6.11}$$

In practice, we uniformly sample about 10 million training triples, evenly distributed among the six training sets (about 1.667 million triples per training set). Likewise, we sample about 400,000 validation triples (about 66,667 triples per training set). No two triples are identical. For experimental purposes, we consider 2 versions of the training set: 1) using all transactions and 2) hiding the last transaction (the last purchase basket) of each user. The second version is intended to evaluate the generalization of the network to new profiles not seen in training. Notice that we use visual clusters and artist together with user transactions to generate the training set D_S, but we do not use curated attributes. We made this decision because expert-annotated metadata depends on manual work, it's not always available and as we saw in the results of content-based methods (Table 5.5) they do not perform as well as FA and CNN. We don't rule out the possibility of exploiting

Table 6.2. nDCG (nD), Recall (R), Precision (P) and F1 Score (F1), User (UC) and Session (SC) coverages for YT-VBPR and all content-based methods. The best three absolute results of each metric are highlighted. The superindex indicates the ID of the method with the closest but still significantly smaller result. For instance, YT-VBPR nD@20 = .6495[1] tells that YT-VBPR has a significantly larger nD@20 than (1)Hybrid$_1$ (nD@20 = .1790) and all the other methods with nD@20 $<$ .1790

ID	Method	nD@20	R@20	P@20	F1@20	UC	SC
0	YT-VBPR (all)	**.6495**[1]	**.8337**[1]	**.0745**[1]	**.1259**[1]	**.2599**	**.4199**
1	Hybrid$_1$ (FA+CNN+PMPCAV)	**.1790**[2]	**.2362**[3]	**.0201**[2]	**.0333**[2]	**.2599**	**.4199**
2	Hybrid$_2$ (FA+CNN)	**.1759**[5]	**.2309**[5]	**.0197**[5]	**.0325**[5]	**.2599**	**.4199**
3	Hybrid$_3$ (FA+PMPCAV)	.1731[5]	.2228[5]	.0190[5]	.0312[4]	**.2599**	**.4199**
4	FA	.1743[5]	.2113[5]	.0180[5]	.0295[5]	**.2599**	**.4199**
5	CNN (All)	.1295[8]	.1702[8]	.0151[6]	.0248[6]	**.2599**	**.4199**
6	CNN (ResNet50)	.1247[7]	.1628[8]	.0145[9]	.0236[8]	**.2599**	**.4199**
7	CNN (AlexNet)	.1081[8]	.1461[8]	.0135[9]	.0216[8]	**.2599**	**.4199**
8	CNN (VGG19)	.1008[9]	.1398[13]	.0124[10]	.0205[9]	**.2599**	**.4199**
9	CNN (InceptionV3)	.1007[10]	.1332[13]	.0125[8]	.0201[10]	**.2599**	**.4199**
10	CNN (NASNet Large)	.0998[13]	.1379[13]	.0120[12]	.0197[12]	**.2599**	**.4199**
11	CNN (InceptionResNetV2)	.0932[13]	.1300[13]	.0119[13]	.0192[13]	**.2599**	**.4199**
12	PMPCAV(All)	.0681[13]	.1051[13]	.0099[13]	.0156[13]	**.2599**	**.4199**
13	HVF (LBP)	.0507[14]	.0736[17]	.0068[14]	.0107[14]	**.2599**	**.4199**
14	HVF (LBP + Attr.)	.0493[16]	.0728[17]	.0064[15]	.0103[16]	**.2599**	**.4199**
15	HVF (Attractiveness)	.0407[16]	.0628[17]	.0059[16]	.0095[17]	**.2599**	**.4199**
16	MPCAV(Medium)	.0337[17]	.0658[17]	.0045[17]	.0081[17]	**.9996**	**.9998**
17	Random	.0097	.0200	.0015	.0025	**1.0000**	**1.0000**

Statistical significance was obtained using multiple pairwise t-tests with Bonferroni correction, $\alpha_{bonf} = \alpha/n$ = 0.05/136 = .00037

curated attributes for sampling higher quality training triplets, but for simplicity we limited ourselves to FA and CNN as sampling heuristics, leaving the use of curated attributes for triplet sampling as an option for future work.

6.4. Evaluation and Results (RQ5)

In Table 6.2 we can observe the results of YT-VBPR (all) along with all previous content-based methods, over all transactions. The *all* in YT-VBPR (all) stands for *trained over all transactions*, meaning that the training set was sampled according to the sampling strategies described in the previous section, but assuming that all transactions from all users are available for sampling. As can be observed, YT-VBPR (all) outperforms all

Table 6.3. Recall (R), Precision (P), F1 Score (F1) and nDCG (nD) of several methods evaluated at the last purchase basket of each user. The best three absolute results of each metric are highlighted.

Method	**R@20**^{last}	**P@20**^{last}	**F1@20**^{last}	**nD@20**^{last}
YT-VBPR (all)	**.9605**	**.0610**	**.1126**	**.6637**
Hybrid$_1$ (FA+CNN+PMPCAV)	**.2781**	**.0164**	**.0307**	**.1688**
Hybrid$_3$ (FA+PMPCAV)	**.2690**	**.0162**	**.0301**	**.1696**
Hybrid$_2$ (FA+CNN)	.2629	.0158	.0294	.1615
FA	.2543	.0151	.0282	.1655
YT-VBPR (last hidden)	.2541	.0155	.0288	.1494
CNN (All)	.1996	.0125	.0231	.1152
VBPR (last hidden)	.1913	.0118	.0219	.1163
PMPCAV(All)	.1370	.0083	.0155	.0715
Random	.0193	.0012	.0022	.0082

content-based methods from the first part by a wide margin. But more importantly, the network achieves such a performance without having user or item specific variables – let's remember that YT-VBPR has a fixed set of trainable parameters independent of the number of users or items. Thus the network has no choice but to learn general parameters to be able to rank items correctly for a huge number of possible profiles. YT-VBPR has to integrate the visual features of items in each user profile in order to generate user vectors on the fly, as we simulate purchases and update user profiles over time during the offline experiment.

Having said that, in order to measure generalization to unseen user profiles and to make a more fair comparison with other baselines, we conducted a second experiment in which we trained YT-VBPR hiding the last purchase basket of each user during training, and used these hidden purchase baskets as the ground truth for evaluation. The results of this second experiment are shown in Table 6.3. We include the results of YT-VBPR (all) as a reference. We also report results for VBPR (R. He & McAuley, 2016) as a baseline. Like YT-VBPR, VBPR also learns its ranking function from the data. For VBPR we defined the training set using sampling strategies similar to strategies (i) and (iii) of YT-VBPR, namely, 1) items purchased by user u should be ranked at the top and 2) non-purchased but visually similar items from u's favorite artists should be ranked at the top

Table 6.4. Recall (R) and Area Under the ROC Curve (AUC) of several methods evaluated at the last purchase basket of each user. The best three absolute results of each metric are highlighted.

Method	**R@20**^{last}	**AUC**^{last}
YT-VBPR (all)	**.9605**	**.9960**
Hybrid$_1$ (FA+CNN+PMPCAV)	**.2781**	**.7335**
CNN (All)	.1996	**.7166**
YT-VBPR (last hidden)	.2541	.7144
VBPR (last hidden)	.1913	.6512
FA	**.2543**	.5949
Random	.0193	.4998

as well. We were not able to implement other sampling strategies for VBPR since VBPR requires both user- and item-specific variables and hence it's not compatible with arbitrary or dynamic user profiles like YT-VBPR does. As shown in Table 6.3, YT-VBPR (last hidden)'s performance decreased considerably with respect to YT-VBPR (all), achieving metrics similar to that of FA (except for nD@20^{last} where FA had a bigger advantage). FA in general performed relatively well, only being outperformed by the hybrids and YT-VBPR (all). It's worth to note that VBPR performed worse than YT-VBPR and CNN (All), which seems to indicate that the use of user- and item-specific parameters in a relatively small dataset yields worse results than YT-VBPR – which shares the same fixed set of parameters across all users and items (forcing it to generalize more by design).

Futhermore, in order to get a better picture of the full ranking quality of each method, we calculated AUC[4] as in (Rendle et al., 2009), reported in Table 6.4. AUC calculates a method's probability of ranking a randomly sampled relevant item higher than a randomly sampled non-relevant item, by looking at the full inventory and counting how many *(relevant, non-relevant)* pairs were correctly ranked (as opposed to other metrics that only look at the top K recommended items). Namely:

$$AUC(u) := \frac{1}{|I_u^+||I \setminus I_u^+|} \sum_{i \in I_u^+} \sum_{j \in I \setminus I_u^+} \delta(\hat{x}_{ui} > \hat{x}_{uj})$$

[4]**A**rea **U**nder the ROC (Receiver Operating Characteristic) **C**urve

where $\hat{x}_{ui}$ is the model's score for user u and item i. Note that Random achieves AUC = 0.5 approx., i.e. the performance of "flipping a coin". We can see that FA performs very bad in terms of AUC. This indicates that when users are going to buy items from previous artists, FA can trivially push relevant items to the top, but when users decide to explore new artists, FA is no different from random chance. Other methods such as the Hybrids, YT-VBPR and even CNN (All) are able to give more reasonable recommendations when users are looking for new artists, and thus they achieve much better AUC than FA on average.

Finally, it would be interesting to know why the performance drops so much from YT-VBPR (all) to YT-VBPR (last hidden). The most obvious reason is the fact that YT-VBPR (all) has access to the last purchase basket of each user during training. But another important reason is the fact that the dataset itself is already relatively small, so hiding all last purchase baskets makes the training set even smaller, which in turn makes it even harder for the network to learn valuable purchase patterns. If this is the case, it would be interesting to test this hypothesis by evaluating YT-VBPR in a larger dataset so as to verify if the gap between YT-VBPR (all) and YT-VBPR (last hidden) narrows when more training data is available.

7. CONCLUSIONS, LIMITATIONS & FUTURE WORK

In this thesis we have studied different approaches to the problem of *one-of-a-kind* artwork recommendation, where the information available comprises artwork images, metadata (artist + curated attributes) and users' purchase history. We studied two general recommendation strategies: 1) content-based methods and 2) hybrid content-collaborative methods. From a purely content-based perspective, we saw that deep convolutional neural networks (CNN) are better visual feature extractors than handcrafted methods, which is consistent with the current state of the art in computer vision. We also saw that the artist of an artwork is a very valuable signal: it captures to a reasonable extent a user's taste, most probably because artists tend to be more or less consistent in terms of style, theme and/or content, increasing the chances of a user liking a new artwork from a previously liked artist. Moreover, when different content features are used simultaneously by a single content-based recommender, the performance generally improves over individual features used in isolation.

Furthermore, when considering both content and collaborative information, our results show that our new proposed hybrid content-collaborative recommendation model called YT-VBPR —a deep neural network that combines ideas from Youtube's Recommender System (Covington et al., 2016) and VBPR (R. He & McAuley, 2016) together with insights learned from content-based methods— achieves the best performance overall, outperforming by a wide margin the best content-based method tested and even VBPR, a strong hybrid content-collaborative baseline. This shows the benefits of training deep neural networks for artwork recommendation (1) with an architecture that leverages content to allow generalization to any user and item and (2) with a training scheme that incorporates collaborative information as well as domain-specific heuristics and insights.

A limitation in our results is the fact that they relate to only one single artwork retailer website – albeit one of the most popular on the Web. This might hinder the generalizability of our results. In addition, other forms of user evaluation are needed in order to test

whether a user evaluation correlates with our offline results, such as a large controlled laboratory study as well as a field online study using A/B testing.

With respect to YT-VBPR, our proposed model, there are ablation studies that still need to be done in order to assess the impact of each design decision, such as the impact of each sampling strategy, the number of layers, using other activation functions, number of neurons per layer, among others. We also want to explore ways to improve YT-VBPR's performance even more. For instance, instead of aggregating the user profile with average and maximum, we could explore more sophisticated aggregation techniques that take into account the temporal dimension, such as binning purchases into discrete temporal categories (e.g. short-term profile, mid-term profile and long-term profile), using recurrent neural networks to process the sequence of purchases, etc. Another aspect to explore is the effect of fine-tuning: in our experiments we always used CNNs with their weights pre-trained on ImageNet, so it would be interesting to see the effect of fine-tuning the CNN's weights along with the rest of the network in a end-to-end fashion. Another possible avenue for improvement is the use of style embedding features extracted from artwork images, motivated by recent work on real-time neural artistic style transfer (Ghiasi, Lee, Kudlur, Dumoulin, & Shlens, 2017). We could combine the higher-level features of traditional CNNs with artistic style embedding features extracted by a neural network such as the one proposed by Ghiasi et al. to enrich the input of YT-VBPR, which we believe should greatly enhance the network's capacity to capture users' taste and therefore improve its performance.

REFERENCES

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, *17*(6), 734–749.

Akçay, S., Kundegorski, M. E., Devereux, M., & Breckon, T. P. (2016). Transfer learning using convolutional neural networks for object classification within x-ray baggage security imagery. In *Proceedings of the ieee international conference on image processing (icip)* (p. 1057-1061).

Albanese, M., d'Acierno, A., Moscato, V., Persia, F., & Picariello, A. (2011). A multimedia semantic recommender system for cultural heritage applications. In *Proceedings of the fifth ieee international conference on semantic computing (icsc)* (pp. 403–410).

Amatriain, X. (2013). Mining large streams of user data for personalized recommendations. *ACM SIGKDD Explorations Newsletter*, *14*(2), 37–48.

Aroyo, L., Wang, Y., Brussee, R., Gorgels, P., Rutledge, L., & Stash, N. (2007). Personalized museum experience: The rijksmuseum use case. In *Proceedings of museums and the web*.

Bennett, J., Lanning, S., et al. (2007). The netflix prize. In *Proceedings of kdd cup and workshop* (Vol. 2007, p. 35).

Benouaret, I., & Lenne, D. (2015). Personalizing the museum experience through context-aware recommendations. In *Proceedings of the ieee international conference on systems, man, and cybernetics (smc)* (pp. 743–748).

Celma, O. (2010). Music recommendation. In *Music recommendation and discovery* (pp. 43–85). Springer.

Covington, P., Adams, J., & Sargin, E. (2016). Deep neural networks for youtube recommendations. In *Proceedings of the 10th acm conference on recommender systems* (pp. 191–198).

Cremonesi, P., Koren, Y., & Turrin, R. (2010). Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth acm conference on recommender systems* (pp. 39–46).

Deldjoo, Y., Elahi, M., Cremonesi, P., Garzotto, F., Piazzolla, P., & Quadrana, M. (2016). Content-based video recommendation system based on stylistic visual features. *Journal on Data Semantics*, *5*(2), 99–113.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Proceedings of the ieee conference on computer vision and pattern recognition (cvpr)* (pp. 248–255).

Ekstrand, M. D., Kluver, D., Harper, F. M., & Konstan, J. A. (2015). Letting users choose recommender algorithms: An experimental study. In *Proceedings of the 9th acm conference on recommender systems* (pp. 11–18).

Elahi, M., Deldjoo, Y., Bakhshandegan Moghaddam, F., Cella, L., Cereda, S., & Cremonesi, P. (2017). Exploring the semantic gap for movie recommendations. In *Proceedings of the eleventh acm conference on recommender systems* (pp. 326–330).

Esman, A. R. (2012). *The World's Strongest Economy? The Global Art Market.* `https://www.forbes.com/sites/abigailesman/2012/02/29/the-worlds-strongest-economy-the-global-art-market/`. ([Online; accessed 21-March-2017])

Ghiasi, G., Lee, H., Kudlur, M., Dumoulin, V., & Shlens, J. (2017). Exploring the structure of a real-time, arbitrary neural artistic stylization network. *arXiv preprint arXiv:1705.06830*.

Gomez-Uribe, C. A., & Hunt, N. (2016). The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)*, *6*(4), 13.

Gonzalez, R. C., Eddins, S. L., & Woods, R. E. (2004). *Digital image publishing using matlab*. Prentice Hall.

Harper, F. M., & Konstan, J. A. (2015, December). The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, *5*(4), 19:1–19:19.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 770–778).

He, R., Fang, C., Wang, Z., & McAuley, J. (2016). Vista: A visually, socially, and temporally-aware model for artistic recommendation. In *Proceedings of the 10th acm conference on recommender systems* (pp. 309–316).

He, R., & McAuley, J. (2016). VBPR: Visual Bayesian Personalized Ranking from implicit feedback. In *Proceedings of the thirtieth aaai conference on artificial intelligence* (pp. 144–150).

Karnowski, J. (2015). *AlexNet + SVM*. `https://jeremykarnowski.files.wordpress.com/2015/07/alexnet2.png`. ([Online; accessed 1-December-2017])

Klambauer, G., Unterthiner, T., Mayr, A., & Hochreiter, S. (2017). Self-normalizing neural networks. In *Advances in neural information processing systems* (pp. 971–980).

Knijnenburg, B. P., Bostandjiev, S., O'Donovan, J., & Kobsa, A. (2012). Inspectability and control in social recommenders. In *Proceedings of the sixth acm conference on recommender systems* (pp. 43–50).

Konstan, J. A., & Riedl, J. (2012). Recommender systems: from algorithms to user experience. *User Modeling and User-Adapted Interaction*, *22*(1-2), 101–123.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*(8).

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Proceedings of advances in neural information processing systems 25 (nips)* (pp. 1097–1105).

La Cascia, M., Sethi, S., & Sclaroff, S. (1998). Combining textual and visual cues for content-based image retrieval on the world wide web. In *Proceedings of the ieee workshop on content-based access of image and video libraries* (pp. 24–28).

Lacic, E., Kowald, D., Eberhard, L., Trattner, C., Parra, D., & Marinho, L. B. (2015). Utilizing online social network and location-based data to recommend products and categories in online marketplaces. In *Mining, modeling, and recommending'things' in social media* (pp. 96–115). Springer.

Lei, C., Liu, D., Li, W., Zha, Z.-J., & Li, H. (2016). Comparative deep learning of hybrid representations for image recommendations. In *Proceedings of the ieee conference on computer vision and pattern recognition (cvpr)* (pp. 2545–2553).

Maaten, L. v. d., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, *9*(Nov), 2579–2605.

Macedo, A. Q., Marinho, L. B., & Santos, R. L. (2015). Context-aware event recommendation in event-based social networks. In *Proceedings of the 9th acm conference on recommender systems* (pp. 123–130).

Maes, P., et al. (1994). Agents that reduce work and information overload. *Communications of the ACM*, *37*(7), 30–40.

Manning, C. D., Raghavan, P., Schütze, H., et al. (2008). *Introduction to information retrieval* (Vol. 1) (No. 1). Cambridge university press Cambridge.

McAuley, J., Targett, C., Shi, Q., & Van Den Hengel, A. (2015). Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international acm sigir conference on research and development in information retrieval* (pp. 43–52).

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (icml-10)* (pp. 807–814).

Nguyen, A., Yosinski, J., & Clune, J. (2016). Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *arXiv preprint arXiv:1602.03616*.

Nunes, I., & Jannach, D. (2017). A systematic review and taxonomy of explanations in decision support and recommender systems. *User Modeling and User-Adapted Interaction*, *27*(3), 393–444.

Ojala, T., Pietikäinen, M., & Harwood, D. (1996). A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, *29*(1), 51–59.

Olah, C., Mordvintsev, A., & Schubert, L. (2017). Feature visualization. *Distill*.

Parra, D., & Brusilovsky, P. (2015). User-controllable personalization: A case study with setfusion. *International Journal of Human-Computer Studies*, *78*, 43–67.

Parra, D., & Sahebi, S. (2013). Recommender systems: Sources of knowledge and evaluation metrics. In *Advanced techniques in web intelligence-2* (pp. 149–175). Springer.

Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2009). BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference*

on uncertainty in artificial intelligence (pp. 452–461).

Rui, Y., Huang, T. S., Ortega, M., & Mehrotra, S. (1998). Relevance feedback: a power tool for interactive content-based image retrieval. *IEEE Transactions on circuits and systems for video technology*, *8*(5), 644–655.

San Pedro, J., & Siersdorfer, S. (2009). Ranking and classifying attractiveness of photos in folksonomies. In *Proceedings of the 18th international conference on world wide web* (pp. 771–780).

Semeraro, G., Lops, P., De Gemmis, M., Musto, C., & Narducci, F. (2012). A folksonomy-based recommender system for personalized access to digital artworks. *Journal on Computing and Cultural Heritage (JOCCH)*, *5*(3), 11.

Sharif Razavian, A., Azizpour, H., Sullivan, J., & Carlsson, S. (2014). CNN features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition workshops* (pp. 806–813).

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of aaai* (Vol. 4, p. 12).

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 2818–2826).

Tintarev, N., & Masthoff, J. (2015). Explaining recommendations: Design and evaluation. In *Recommender systems handbook* (pp. 353–382). Springer.

Trattner, C., & Elsweiler, D. (2017). Investigating the healthiness of internet-sourced

recipes: implications for meal planning and recommender systems. In *Proceedings of the 26th international conference on world wide web* (pp. 489–498).

Verbert, K., Parra, D., Brusilovsky, P., & Duval, E. (2013). Visualizing recommendations to support exploration, transparency and controllability. In *Proceedings of the 2013 international conference on intelligent user interfaces* (pp. 351–362).

Weinswig, D. (2016). *Art Market Cooling, But Online Sales Booming.* `https://www.forbes.com/sites/deborahweinswig/2016/05/13/art-market-cooling-but-online-sales-booming/`. ([Online; accessed 21-March-2017])

Zoph, B., Vasudevan, V., Shlens, J., & Le, Q. V. (2017). Learning transferable architectures for scalable image recognition. *arXiv preprint arXiv:1707.07012*, *2*(6).