UNIVERSIDADE DE LISBOA

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE BIOLOGIA VEGETAL

Emotion-based Image Recommendation System

Maria Leonor da Silva Lopes Pereira de Miranda

Mestrado em Bioinformática e Biologia Computacional

Dissertação orientada por:
Professora Doutora Márcia Cristina Afonso Barros
Professora Doutora Soraia Vanessa Meneses Alarcão Castelo de Almeida Pires

2025

Acknowledgments

First, I would like to thank my advisors, Márcia Barros and Soraia Meneses Alarcão, for their support and guidance, especially their patience with all my doubts. They were an essential part of this dissertation. Thank you. I would also like to thank LASIGE for the opportunity to work in this laboratory. I would also like to give special recognition to all the fantastic people I have met during this academic journey, Benedita Vasconcelos and Mariana Antunes. I would particularly like to mention the incredible people I was lucky enough to meet during this master's degree: David Correia, Joana, João Inácio, Pedro Sequeira, Marta Ramos, Madalena Girão, Pedro Travessa, Rafaela Lopes, Márcia Vital and Miguel Graça.

To my friends Carlota Fernandes, Joana Santos, Joana Costa, Joana Oliveira, Sofia Moreira, Marta Teixeira, Maria Carreira, and Madalena Gil, for always being a shoulder to lean on and having a caring word, even when I was most discouraged. I would also like to mention the teaching team at Lisbon Zoo for always welcoming me with a smile.

I want to thank my family, both those who are here and those who have already left, for making me a person who isn't afraid to follow her dreams. Thanks to my parents, Clara and Paulo Miranda, for encouraging me to be a better person. Thank you, Dad, for being a haven, listening, motivating, and advising me. Thank you, Mom, for being my haven and teaching me that sometimes you must sit on a rock and listen to your heart. I'd also like to thank my grandmother Arminda Miranda and my aunt Maria Helena Tomás for showing me the unconditional affection of a grandmother.

My most enormous thanks to my boyfriend, João Pedro Silva, for always being there, for showing me unconditional support and unlimited patience, for the theoretical discussions, and for all the help, affection, and motivation you gave me. You were a fundamental part of this dissertation.

Abstract

Recommendation systems play a crucial role in today's digital society by helping users make decisions when so much information is available. Among other types of content, images are ingrained in people's daily lives and can significantly influence their emotional states. Because of this intense emotional impact, images have become important tools in therapeutic contexts for the treatment of dementia and post-traumatic stress. By considering emotional responses, an image recommendation system can personalize and make recommendations according to emotions, increasing relevance and effectiveness. This work focuses on developing an image recommendation system where emotions felt by the user are a feature for the recommendation. This study was done with a new dataset of the EmoRecSys Project that is still under construction. To do that, we developed an emotion-based recommendation system based on the user's declared explicit emotions from the dataset. To evaluate the performance of the proposed recommendation system, two more recommendation systems that are more widely studied in the literature were developed: pixel and metadata, as well as a random recommendation system that works as a baseline. The systems' performance was evaluated using precision, recall, f1-score, and normalized discounted cumulative gain. The results at top@10 for precision were 0.119, 0.113, and 0.111 for the emotion-, pixel-, and metadata-based, respectively. Regarding recall, at top@10, the recalls are 0.542, 0.505, and 0.492 for the emotion-, pixel-, and metadata-based, respectively. In the normalized discounted cumulative gain, emotions, pixels, and annotations have top@10 of 0.302, 0.299, and 0.285. The results show that the recommendation of images based on emotions achieve better results than using state of the art features, such as pixel and metadata similarities

Keywords: Content-based, Recommendation System, Emotions, Images

Resumo

Os sistemas de recomendação têm um papel crucial na nossa sociedade, na era digital, permitindo facilitar o processo de tomada de decisão e busca quando existe muita informação disponível. Estes encontram-se presentes no nosso dia-a-dia sob várias formas desde as redes sociais aos sites e plataformas de compras online, de forma que consigamos encontrar com maior facilidade os vídeos, imagens ou produtos em que temos interesse. O principal objetivo destes algoritmos é tornar a experiência de pesquisa de informação mais agradável e personalizável. Para isto existem 3 tipos principais de sistemas de recomendação: a filtragem colaborativa, que usa a semelhança entre os gostos dos utilizadores para realizar recomendações; o sistema de recomendação baseado em conteúdo que usa a semelhança entre as características intrínsecas dos próprios itens a tarefa de recomendação; e ainda o sistema híbrido que agrega diferentes tipos de sistema de recomendação de forma a colmatar as desvantagens de cada tipo de sistema, como a falta de diversidade ou quando temos utilizadores novos. Em todos os aspetos nas nossas vidas, o conteúdo multimédia encontra-se presente, seja em esquemas e gráficos que nos permitem interpretar melhor os dados e informações que teremos de analisar, seja em vídeos e imagens nas redes sociais. Além disso, as imagens também têm o propósito de comunicar pensamentos, ideias e sentimentos. Desde o exemplo mais simples como os emojis, que transmitem emoção às nossas mensagens de texto, até aos vídeos e imagens que se encontram nas mais diversas plataformas das redes sociais como o Instagram ou o Youtube. Uma outra componente importante é o uso de imagens em diversas terapias para diferentes tipos de síndromes, como a demência e o stress pós-traumático. Estas terapias tiram partido do facto de as imagens nos transmitirem emoções, permitindo melhorar as vidas de milhares de pessoas que sofrem destas síndromes. Tendo em conta o carácter pessoal deste tipo de terapias, é importante garantir que as imagens selecionadas sejam adequadas a cada pessoa e a cada situação, de modo a tornar a terapia mais eficaz no seu tratamento. Apesar de existir muita investigação na área dos algoritmos de recomendação, particularmente de imagens, não existem muitos sistemas que tenham em conta as emoções elicitadas pelas imagens em cada pessoa.

Numa abordagem baseada no conteúdo (sem ter em conta relações entre os diferentes utilizadores), existem muitos sistemas de recomendação de imagens. Existem algoritmos baseados nos pixéis em que, utilizando técnicas de aprendizagem automática, utilizam os pixéis de cada imagem como características para avaliar a semelhança entre cada imagem, levando a recomendar as mais semelhantes às que o utilizador gosta. Existem outros algoritmos baseados em meta dados, como anotações às imagens que, através de diferentes algoritmos de semelhança de palavras, procuram recomendar imagens mais semelhantes às do gosto do utilizador. No entanto, não existem muitos sistemas capazes de utilizar as emoções

como características para procurar semelhanças entre as imagens. O objetivo deste trabalho será realizar um sistema de recomendação baseado em conteúdo para a recomendação de imagens servindo-se das emoções que as mesmas evocam. Assim foi utilizado um conjunto de dados pertencente ao grupo de investigação no qual este trabalho está inserido, onde estão representadas imagens, utilizadores e as emoções que as imagens evocam reportadas explicitamente pelos utilizadores. O estudo foi realizado offline havendo uma divisão do conjunto de dados. Das 15 imagens que cada utilizador avaliou, 12 imagens constituíram o perfil do utilizador, e as restantes 3 imagens avaliadas em conjunto com mais 20 imagens pertencentes a uma amostragem negativa formaram o teste. Para a tarefa de recomendação foram utilizadas as emoções explícitas que cada imagem continha para realizar a comparação entre as imagens, servindo-nos de métricas de similaridade como a distância euclidiana e a similaridade do cosseno. De seguida para fazer uma comparação com as imagens que o utilizador já tinha previamente gostado e desgostado (pertencentes ao perfil do utilizador) foram calculadas três pontuações diferentes: a primeira que, ao comparar as imagens e classificações do perfil do utilizador com as que queríamos recomendar, realizava uma média simples das classificações; a segunda que realizava uma média ponderada das classificações em conjunto com o valor da semelhança entre as imagens; finalmente a terceira pontuação que tem em consideração uma média ponderada dos quadrados dos valores das semelhanças.

Para compreender se as emoções tem de facto valor na recomendação de imagens foram feitos mais três sistemas de recomendação: um com base nos pixéis das imagens, outro com base nos meta dados (anotações descritivas) das mesmas e por último um sistema de recomendação aleatório para servir como base para a avaliação dos outros sistemas referidos. Para o sistema de recomendação que tem por base os pixéis das imagens for feito um vetor com o auxílio de diferentes arquiteturas de redes neuronais convolucionais. Para os meta dados foram utilizados modelos de processamento de linguagem natural, como a frequência do termo e a frequência inversa do documento, entre outros modelos do estado de arte da literatura. No resto do processo de recomendação, foram usados os mesmos métodos descritos no parágrafo anterior (distância euclidiana e similaridade do cosseno) em conjunto com as três pontuações.

Para a avaliação dos modelos foram utilizadas métricas como a precisão@k, evocação@k, pontuação F1@k e o ganho cumulativo com desconto normalizado (nDCG@k). A precisão@k mede a proporção de itens que foram identificados como relevantes entre os k itens no topo da lista de resultados. A evocação@k mede a proporção de itens relevantes recuperados nos top k, em relação ao número total de itens relevantes disponíveis. A pontuação F1@k é a média harmónica entre a precisão@k e a evocação@k. Por último, ganho cumulativo com desconto normalizado (nDCG@k), tem em conta não só a relevância dos itens, mas também a sua posição na lista, aplicando um desconto aos itens que aparecem pior posicionados. Os resultados que obtivemos foram satisfatórios, ao colocar a precisão como métrica de avaliação, uma vez que esta reflete diretamente a proporção de recomendações relevantes entre os itens sugeridos. Os resultados foram apresentados numa lista de k itens recomendados. Para k igual a 10 temos que a precisão@k apresenta os seguintes resultados para o sistema baseado em emoções, pixéis e meta dados, respetivamente: 0.119, 0.113, e 0.111. Para a evocação@k temos 0.542, 0.505 e 0.492, e para o nDCG@k temos os seguintes valores 0.302, 0.299, e 0.285, para as emoções, pixéis e meta dados, respetivamente. Estes resultados demonstram que, apesar de a diferença entre os modelos não ser pronunciada, as emoções podem ser utilizadas enquanto característica no que diz respeito à recomendação

de imagens. Os resultados apresentaram ser superiores ao sistema de recomendação aleatório provando assim, a sua validade enquanto recomendadores. Este trabalho evidenciou a relevância dos sistemas de recomendação baseados em conteúdo de pixéis e meta dados, com particular destaque para os que têm as emoções como característica primária dos itens. Foi demonstrado que, embora os modelos com base em pixéis se destaquem para a maioria dos ks, o facto de para $k = 10$ as emoções ultrapassarem estes sistemas, embora por pouco, demonstram que o papel das emoções na tarefa de recomendação não é indispensável. Ao combinar as características das imagens, os pixéis, anotações e emoções num sistema de recomendação baseado em conteúdo futuro, poderá melhorar os valores obtidos assim como a redução de dimensionalidade dos vetores de características. Um sistema de recomendação híbrido com um sistema baseado em conteúdo e de filtragem colaborativa poderá também ser de interesse para a recomendação das imagens.

Palavras Chave: Filtragem baseada em Conteúdo, Sistemas de Recomendação, Emoções, Imagens

Contents

List of Figures

List of Tables

Acronyms

PTSD Post-Traumatic Stress Disorder

CNN Convolutional Neural Networks

RMSE Root Mean Squared Error

MAE Mean Absolute Error

nDCG Normalized Discounted Cumulative Gain

PCA Principal Component Analysis

t-SNE t-Distributed Stochastic Neighbor Embedding

UMAP Uniform Manifold Approximation and Projection

LSH Locality-Sensitive Hashing

IFT Information Foraging Theory

HTML HyperText Markup Language

SIFT Scale-Invariant Feature Transform

SURF Speeded-Up Robust Features

LBP Local Binary Patterns

SVM Support Vector Machine

NN Neural Network

VRS Visually-aware Recommender Systems

VBPR Visual Bayesian Personalized Ranking

DeepStyle Deep Learning for Style-based Recommendations

VNPR Visual Neural Personalized Ranking

ACF Adaptive Collaborative Filtering

IFE Image Feature Extractor

DNN Deep Neural Network

AVS Audio-Visual System

RGB Red, Green, Blue

PHOG Pyramid Histogram of Oriented Gradients

EM Expectation Maximization

PAD Pleasure-Arousal-Dominance

VGG Visual Geometry Group

EARS Emotion Aware Recommender System

EMERS Emotion-aware Music Recommender System

IAPS International Affective Picture System

VAD Valence, Arousal, Dominance

URL Uniform Resource Locator

TF-IDF Term Frequency-Inverse Document Frequency

SNS Social Network Sites

MAP Mean Average Precision

PwD People with Dementia

SAM Self-Assessment Manikin

BLIP Bootstrapped Language-Image Pre-training

ITC Image-Text Construal

CIDEr Consensus-based Image Description Evaluation

SPICE Semantic Propositional Image Caption Evaluation

USE Universal Sentence Encoder

BERT Bidirectional Encoder Representations from Transformers

CLS Classification token

SEP Separator token

PTB Penn Treebank

DCG Discounte Cumulative Gain

MRR Mean Reciprocal Rank

AUC Area Under the Curve

SVD Singular Value Decomposition

ResNet Residual Network

CLIP Contrastive Language-Image Pretraining

ReLu Rectified Linear Unit

Chapter 1

Introduction

1.1 Motivation and Problems

Multimedia is a central part of everyday life in today's society. Images, mainly, are highly relevant because they reveal a real-world perspective. When coming into contact with different images, humans are involved in various emotions that directly and indirectly influence our well-being. At the same time, the interpretation of each image depends on the individual who observes it, their experiences, their context, and their emotional state. Therefore, selecting images to present to people must be done carefully and personally. This care is even more relevant when the images are applied to specific treatments, such as reminiscence therapy for people with dementia or Post-Traumatic Stress Disorder (PTSD).

A tool for selecting images is recommendation systems, which have been widely used in other domains, such as social networks, online shopping sites, or even search engines. These tools enable a more personalized experience and show strong potential for tailoring therapeutic interventions through digital images. This personalization is even more relevant when it comes to people with dementia or other medical conditions such as depression and PTSD because their emotional responses may be more pronounced, and sensitivity to certain visual stimuli may be different when compared to people without a condition. For all these reasons, it could be advantageous to use the emotions provoked by images as features for a recommendation algorithm, which could be used for these therapies.

1.2 Objectives

With the aim of tackling existing problems, our main objective is to develop an image recommendation system based on users' emotional responses — specifically, emotions reported by users concerning the images they view. This approach is especially relevant in contexts where the objective is to generate greater emotional closeness to users. Its personalized and in-depth dimension focuses on the individual experience.

Four recommendation systems (Emotion-based, Pixel-based, Metadata-based, and Random) were designed. The development of the emotion-based recommendation system is the main focus of this work and the other three were developed for comparison to serve as a baseline.

1. **Emotion-based (proposed system):** the system leverages the emotions reported by multiple users for each image to generate recommendations, using emotional similarity as the basis for comparing images.

2. **Pixel-based:** explores the visual characteristics of images by using image similarity considering pixels.

3. **Metadata-based:** in this system, the similarity was explored according to the textual descriptive information associated with the images.

4. **Random Recommendations System:** it is a simple approach that suggests images completely at random.

1.3 Research Questions

In order to delimit the scope and better define the ambit of this work, the following research questions were defined:

- **RQ1:** Can emotions reported by different users be used as a single feature for recommending images?

- **RQ2:** Is an emotion-based recommendation system more effective than traditional approaches such as pixel-based, metadata-based, and random baselines?

1.4 Contributions

The main contributions of this work are:

- The design and implementation of a novel emotion-based image recommendation system;

- A comparative analysis with three baseline systems: pixel-based, metadata-based, and random;

- An empirical evaluation using standard recommendation metrics (precision, recall, F1-score, and nDCG);

- The presentation of this work in the form of a poster at the 9th LASIGE Workshop in 2024.

1.5 Document Structure

The remainder of this document is organized into four chapters. The Background and Related Work chapter reviews the theoretical foundations, important key concepts, and past research relevant to the project. Methodology chapter explores the methodology used to develop and evaluate the proposed recommendation system. It also provides details of the dataset, preprocessing steps, and implementation of the four recommendation systems developed. Results and Discussion chapter presents the results obtained for each of the recommendation systems developed. The comparison between them is emphasized. Conclusion and Future Work chapter summarizes the entire study and the impact of the work and suggests how it can be improved.

Chapter 2

Background and Related Work

In this chapter, the key concepts are presented to help understand the key concepts of this dissertation: emotions, recommendation systems, the different types of systems that exist, how they work, and how we can evaluate them. We will also summarize different methodologies used in content-based recommendation systems.

2.1 Emotions

Emotions play a fundamental role in our lives and in the way we think, understand the world, and interact with our peers. Emotions are complex psychological states that combine subjective feelings, psychological changes, and expressive behaviors, often triggered by events that have a personal meaning [6].

In addition to facilitating interpersonal communication, emotions guide us in recognizing what truly matters and help us respond effectively to challenges in our surroundings [7]. Rather than being an outdated relic of human evolution, emotions are integral to decision-making and cognitive processes, demonstrating their enduring importance to our well-being and everyday experiences [8].

Various theoretical perspectives have emerged that give us more insight into the complexity of human emotions. These include discrete theories that emphasize different emotional states, dimensional theories that position emotions in a continuous space, and cognitive theories that highlight the role of cognitive processes in shaping emotional experiences [9].

Theories of discrete emotions are based on the idea that there are fundamental emotions that cross cultures and are supported by biology [10]. Thus, there is a distinction between "basic" emotions, according to Ekman [11], which are six (Anger, Contempt, Disgust, Enjoyment, Fear, Sadness, Surprise), and all other emotions. Basic emotions are expressed through facial expressions, which are unique and universal. The other emotions are a mix of basic emotions (see Figure 2.1). According to Plutchik [12], there are eight basic emotions (Anger, Disgust, Sadness, Surprise, Fear, Trust, Joy, and Anticipation), each with several related terms (see Figure 2.2). Although the theory of discrete emotions does not cover all derived emotions, it is still an important tool for investigating basic emotional processes.

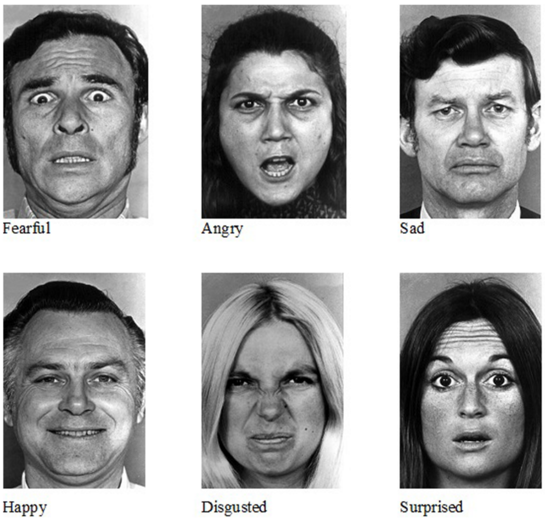


Figure 2.1: Ekman's six emotions represented by facial expressions [1].

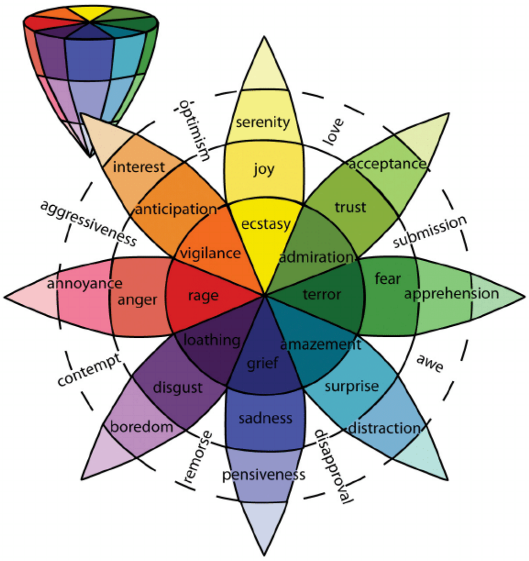


Figure 2.2: Plutchik's model describes the relations among emotion concepts, which are analogous to the colors on a color wheel [2] (best seen in color).

From a dimensional perspective, emotions are constructed as outcomes of a fixed number of concepts within a dimensional space. According to Barrett's conceptualization, emotions are mapped onto specific dimensions, namely valence, arousal, and dominance [13]. Valence delineates the spectrum from negative to positive feelings, while arousal means the human body's activation level, ranging from drowsiness to excitement. Dominance corresponds to the intensity of the emotion experienced. This perspective aligns with the two-dimensional Circumplex Model of Affect, which posits that all affective states result from cognitive interpretations of core neural sensations [14]. In addition, Russell mapped the central coordinates of particular discrete emotions within the structure of the dimensional model. He argued that these coordinates are fluid, shifting over time as individuals grow and differ among people and groups based on their unique personality traits [15]. The dimensional perspective captures the connections between emotions, mapping them in space.

Cognitive theories of emotion are based on cognitive processes' role in regulating emotions. According to cognitive theorists, there can be a panoply of emotional responses to the same event, which are influenced by the individual's evaluation of the situation. The theory of Cognitive Theories of Emotion states that emotional experiences are characterized by the objective attributes of the situation and the interpretation of the individual who perceives them. Smith and Lazarus' appraisal theory states that emotions are elicited when an individual processes or evaluates a situation relevant to their goals, concerns, and well-being [16]. Another model is Ortony, Clore, and Collins, who state that emotions result from the interaction between an individual's processing of an event and their beliefs, desires, and intentions [9].

2.2 Machine Learning

For context, we will provide an overview of artificial intelligence, machine learning, and deep learning according to [17].

In computer science, artificial intelligence is defined as the study of any device that perceives itself and takes actions that maximize the success of any objective. Machine learning is a subfield of computer science that explores the construction of algorithms that can learn and make predictions from data. This area of study can be summarized in three parts:

- **Supervised Learning:** this category uses labeled data (data for which both the input features and the corresponding correct outputs are provided) to train the model. It is usually used in applications that rely on historical data to predict the future.
- **Unsupervised Learning:** which is used on data without labels. This category must discover underlying patterns or groupings in the data. The goal is to explore the dataset and identify its internal structure.
- **Semi-supervised Learning:** which is used in the same applications as supervised learning but uses models with and without labels for training.

Deep Learning is the study of neural networks (a model inspired by the workings of the human brain) that contain more than one hidden layer.

2.2.1 Convolution Neural Networks

A Convolutional Neural Networks (CNN) is a type of architecture of a neural network learning [18], specifically a feedforward neural network capable of extracting features.

The architecture of CNNs is inspired by visual perception. A biological neuron corresponds to an artificial neuron; the kernels of the CNNs represent different receptors that can respond to different features, and the activation functions simulate the function that only neuronal electrical signals that exceed a certain threshold can be transmitted to the next neuron [19].

CNNs are used for multiple tasks in various domains because of their performance, such as image classification, object and face detection, and speech and vehicle recognition, among many others.

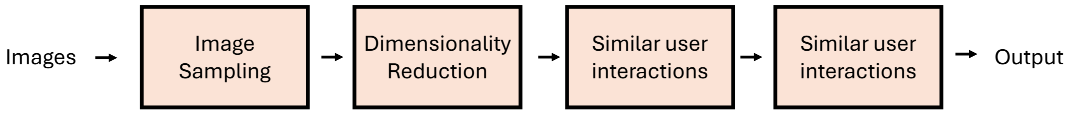


Figure 2.3: Elementary Components of a CNN [3].

Figure 2.3 shows a general model of a CNN, composed of the following components: the convolutional network (Image Sampling), the polling layer (Dimensionality Reduction), the activation function (Feature Extraction) and a fully connected layer (Classifier) [3]. Various CNNs architectures have been developed over the years, each with its characteristics and innovations. Some of the most widely used are:

- **AlexNet:** one of the first CNN architectures to achieve significant advancements in image recognition and classification, using dropout (which is a regularization technique that prevents overfitting by randomly disabling neurons during training) and Rectified Linear Unit (ReLu) (which is an activation function widely used for its simplicity and effectiveness in avoiding vanishing gradients).
- **VGG:** a multi-layered architecture that demonstrates that the use of small filters, such as $3x3$, can improve CNNs performance.
- **ResNet:** uses skip connections (concept that help mitigate the vanishing gradient problem (is a challenge that occurs during the training of deep neural networks, particularly those with many layers) by allowing gradients to flow more directly through deeper networks) to avoid the vanishing gradient problem and allow much deeper networks to be trained.
- **GoogleNet:**, also known as Inception-V1, introduced the concept of blocks and used different filter sizes to capture spatial and channel information at various resolutions.

CNNs continue to evolve with increasingly powerful emerging architectures to improve their performance and respond to more specific problems.

2.2.2 Text mining

Text mining is a term that refers to the process of extracting information or patterns from textual documents. The main objective of text mining is to discover unknown information, whether explicitly written or implied in the text [4].

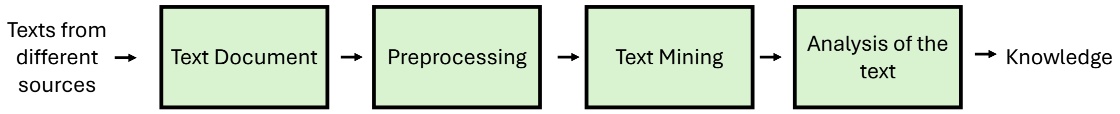


Figure 2.4: Text mining process inspired by [4].

The overall text mining process can be seen in Figure 2.4. This process begins by collecting pieces of text from different sources into a document, then pre-processing the text document, and then applying a text mining technique to analyze the text [4]. Text mining techniques are divided into the following categories according to the result they aim to achieve, according to [4; 20]. They are as follows:

- **Information extraction:** aims to extract meaningful information from a large amount of text. Domain experts specify the attributes and relationships according to the domain.
- **Information retrieval:** this process can be described by extracting relevant patterns and patterns associated with a given set of words or phrases.
- **Natural Language Processing:** focuses on the automatic processing and analysis of unstructured textual information.
- **Clustering:** is an unsupervised process to classify textual documents by applying different clustering algorithms.
- **Text Summarization:** collects and produces a concise representation of original text documents. Pre-processing and processing operations are carried out to summarize the raw text.

Text mining techniques are valuable for efficiently analyzing meaningful information from large amounts of data.

2.3 Recommendation Systems

Decision-making is the cognitive process of identifying and choosing alternatives based on preferences, beliefs, and the degree of importance the decision-maker gives to objects or actions. Recommendation systems help people make this choice, usually by computing a short list of suggestions that reduces the space of possible options [21]. There are a few key concepts for understanding what recommendation systems are. The recommendation system is a tool that uses user data, such as preferences, to suggest items (see Figure 2.5). A user is the person interacting with the system, and an item is an object (or content) on which the user interacts with the system.

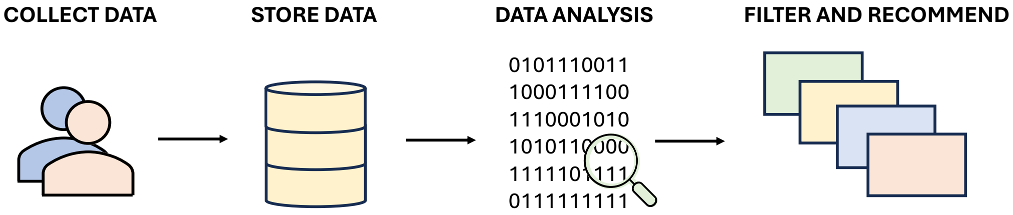


Figure 2.5: Step-by-step process to how recommendation systems work.

The basic principle of recommendations is that there is a dependency between the user's activity and the item. For example, if a user has enjoyed a wildlife documentary, they are probably also interested in other wildlife documentaries or an educational program rather than a comedy film. Different categories often show significant correlations, which can be balanced with more precise recommendations. If we go even deeper, correlations can exist at the level of the intrinsic characteristics of the items. These dependencies can be learned in a data-driven way through a classification matrix, and the resulting model is used to make predictions for target users. The greater the number of evaluated items available to the user, the easier it becomes to make robust predictions about the user's future behavior. One problem that can arise from this point is the lack of novelty in the recommendations given. Various learning models can be used to accomplish this task. However, the objectives of recommendation systems are transversal. Some of these objectives, according to [22], are as follows:

- **Relevance**: the most obvious goal of a recommendation system is to recommend items that are relevant to the user in question since users are more likely to like and consume items that they find relevant.

- **Novelty**: Recommendation Systems are only advantageous if they recommend an item the user has never seen. The repeated recommendation of a specific type of item would rarely be new to the user.

- **Serendipity**: refers to discovering new and joyful things by chance. Chance differs from novelty because the recommendations are truly surprising to the user rather than simply something they did not know before. It can happen that a user is consuming items of a single type because they do not know all the different types of items that exist and have a pleasant surprise with that type of item.

- **Diversity**: usually, the Recommendation Systems suggests a list of top k items. When all the items in this list are very similar to each other, the risk of the user not liking any of these items increases. However, if the items on this list are from different types, the user is more likely to like at least one of them. The goal is to ensure the user is not bored with repeated item recommendations.

When developing a recommendation system, in addition to the objectives mentioned above, we have to take other factors into account, such as the amount of data available and the feedback mechanisms, so that we can develop a model capable of identifying patterns and then making better recommendations. How these challenges are addressed has raised great interest in both academia and industry, leading to the rapid development of new recommendation and evaluation methodologies.

One of the key aspects of recommendation systems is the nature of the feedback used to train the model. Feedback may be explicit or implicit. Explicit feedback is a user-initiated action (e.g., ratings or likes/dislikes), where users explicitly define their preferences. Implicit feedback is induced from user action, (e.g., view duration, clicks, or navigation behavior), without user intervention. While direct feedback is a strong signal, it tends to be in short supply, while indirect feedback is abundant but noisy. Acknowledging the difference between the two types of feedback is an important step while designing recommendation systems.

2.3.1 Fundamental Approaches to Recommendation Systems

There are several types of recommendation systems, such a) knowledge-based, which uses information from items based on user needs and is defined by rules and ontologies; b) demographic-based, which uses demographic information about users, such as age and gender, to compare and provide recommendations, c) content-based, which use intrinsic characteristics of items and users' past preferences, d) collaborative filtering systems that are based on the information they have from other users, the rating they have given to different items and the similarity of these ratings with other users, and finally e) hybrid systems that combine within several systems, taking advantage of the combinations of different recommendation systems.

Following, we present three types of recommendation systems: collaborative filtering, content-based, and hybrid approaches, which are the most usual approaches.

2.3.1.1 Collaborative Filtering Models

Collaborative filtering models use their knowledge of other users' item ratings to make recommendations. One of the challenges of this method is that many of the items in a universe of items have not been evaluated, leading to data with high sparsity. For example, on a website about books, not all the existing books have been categorized, meaning that many books remain uncategorized. The idea behind collaborative filtering is that these missing ratings can be predicted based on the patterns of the available ratings, since ratings between users and items usually show significant correlations. This method compares the ratings that different users have given (user u_i and user u_j). If they are similar, we can infer that even if one of the users u_i has not seen a particular item, their rating will be identical to the user u_j, who has the same preferences and has seen that exact item. The filtering model exploits the correlations between users (inter-users) and the correlations between items (inter-items) or both. There are also two main types of techniques used in this type of model:

1. **Memory-based:** These methods focus on predicting user-item combinations according to the two neighborhoods. These neighborhoods can be defined as user-based or item-based. In the first case, the ratings of other users similar to the user u_i are used to make predictions for this user. Thus, the idea is to distinguish the users most similar to the user u_i to use the weighted averages of the ratings they have given to predict the rating of an item that user u_i has not yet viewed. In the second case, to indicate whether user u_j might like a specific item 1, i_1, we look at the items most similar to item i_1 and see whether or not user u_j liked the identical items to recommend item i_1. The main advantage of this technique is that it is a straightforward implementation.

2. **Model-based:** These techniques are based on machine learning models and data mining, which are used to build predictive models. Decision trees, Bayesian models, latent factor models, matrix factorization, and graph methods are commonly used in this model type. Typically, these techniques perform better than memory-based ones even with sparse classification matrices (with several elements absent) [23].

The biggest challenges with this type of system are: i) the cold-start problem, where we do not have enough interactions between users and items to draw meaningful conclusions or make accurate recommendations, and ii) the data sparsity problem, where with the increase in users and items, it is challenging to predict classifications for all items since the similarities between users and items can be scarce. This last problem may be even more evident nowadays since we have so much information available on the internet [24].

2.3.1.2 Content-based Filtering Models

Content-based recommendation systems use the intrinsic characteristics of items to make recommendations. The term content refers to these same characteristics. These methods combine the ratings with the information available about the items. For example, if the user u_i likes an item i_1, then it is likely that this same user will like items similar to the item i_1. In this case, other users' ratings are not considered. In content-based models, the items' descriptions and ratings are used to create classification or regression models specific to each user. Thus, the training of these models will correspond to the descriptions of the items with which the user in question has already interacted. The trained model is then used to predict the rating the user might give to an item they have never seen. This type of filtering is beneficial for avoiding the cold-start problem for new items (items without ratings) since we do not need many interactions from different users. The likelihood that the target user has already rated an item similar to the one we want to recommend is high. However, this method lacks the diversity of recommended items since it is unlikely to be recommended if a user has never interacted with a specific category of items. For example, if user u_i has never interacted with mystery books, they are unlikely to be recommended. On the other hand, we need a reasonable history of interactions from each user so that the model can be more accurate since it is trained with the items that a particular user has already classified [22; 25]. Figure 2.6 presents the difference between Content-based and Collaborative Filtering models.

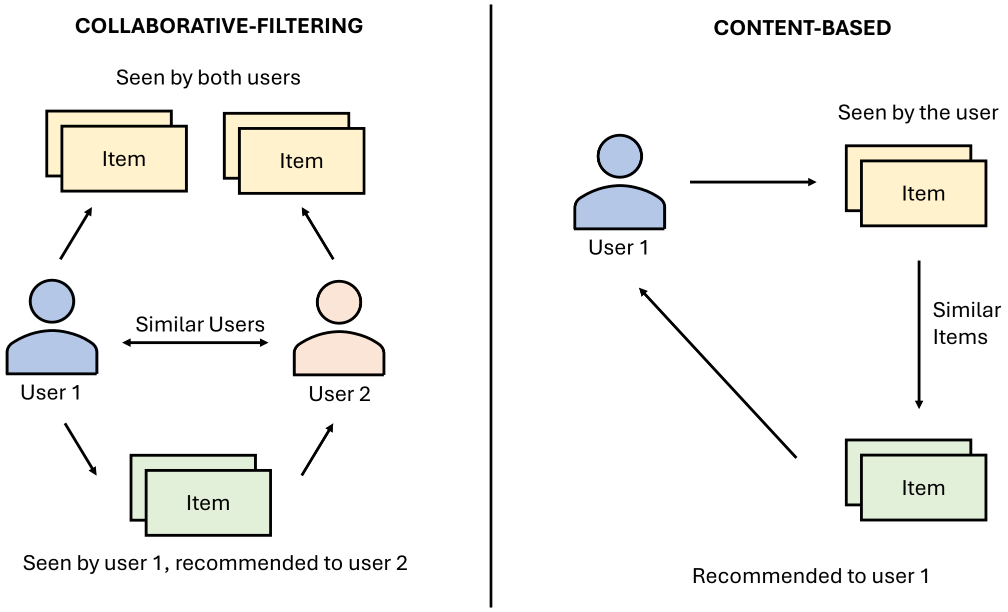


Figure 2.6: Recommendation Systems. On the left is a representation of a collaborative filtering recommendation system. On the right is a representation of a content-based recommendation system. Image adapted from [5] (best seen in color).

2.3.1.3 Hybrid Filtering Models

Hybrid Recommendation Systems can be a solution when we have a wide variety of available inputs and can use different recommendation systems for the same task [22]. The hybrid filtering technique combines different recommendation techniques to optimize the system to avoid some of the limitations and problems of recommendation systems. This concept encompasses different ways of combining models. The idea behind hybrid techniques is that combining algorithms will provide more accurate and effective recommendations than a single algorithm, as one algorithm can overcome the disadvantages of the other [25]. Various types of hybridization techniques combine recommendation models; this taxonomy (means of organizing the different types of hybridization) was suggested by [26], such as:

- **Weighted**: this model type computes the results of different recommendation methods into a single score using weighted linear functions. For example, the first meaningful recommendation system was P-Tango [27], which initially gave equal weights to the collaborative filtering and content-based models for recommending newspapers and then adapted the weights according to user feedback.

- **Switching**: depending on the situation, the system changes its recommendation technique to tackle the problem inherent to the data type. For example, the system can switch from the collaborative filtering recommendation method to a content-based one when there are few similar user interactions.

- **Mixed**: Recommendations from different recommendation systems are presented simultaneously. In other words, we have several recommendation systems making recommendations simultaneously. These recommendations are combined to make a final list of recommended items.
- **Feature combination**: In this hybrid type, a single recommendation algorithm combines characteristics from different data sources. For example, in the case of a Collaborative filtering, Content-based hybrid, the system does not depend exclusively on the output of collaborative filtering data. This output is considered as additional content-based data, generating the final list of recommendations. This reduces sensitivity to the scarcity of initial data.
- **Cascade**: One recommendation system refines the results of the recommendations given by another. In other words, a first recommendation technique is used, which is then provided as input to another recommendation model to refine these recommendations. This technique is sensitive to the order in which the systems are shown first by content-based, then collaborative filtering, or the other way round.
- **Feature augmentation**: The output of one system is used as input for a feature of the other. These systems are also order-sensitive as they are characterized by the second technique being based on the output of the other. However, it focuses on enhancing the richness of features provided to the second algorithm instead of directly altering the recommendation list
- **Meta-level**: The model learned by one recommendation system is used as input for another. It is another type of order-sensitive recommendation since it uses the entire model produced by the first as input for the other. It is typical to use content-based models first and then build collaborative filtering recommendation models.

Hybrid recommendation systems may alleviate some of the challenges of single recommendation methods. However, for this type of system to be successful, it is necessary to critically analyze the data and recommendation systems, as well as the objectives (prioritize recommendation time and precision of recommendations, among others) and computational resources available (such as memory or processing limitations).

2.3.2 Problems and Challenges

Following, we present some of the problems and challenges that recommendation systems face, both ethically and in terms of implementation [22; 28; 29]. We summarize where these problems fit into the content-based and collaborative filtering recommendation systems (see Table 2.1).

- **Cold Start Problem:** this problem occurs when new users or items are added. In these cases, we cannot predict the new user's liking or know how other users have rated the items, so the recommendations may not be accurate. This problem can be alleviated by asking the user to explicitly define their liking, suggesting items based on demographic information, or asking them to rate some items from the start.

- **Synonymy:** this problem arises when an item is represented more than once with different names or entries having identical meanings. In these cases, the recommendation system cannot identify whether the term is applied to other items or the same item.

- **Shilling Attacks:** this problem occurs when a user who wants to harm the system or a competitor starts giving false ratings on certain items to increase or decrease their popularity. Such attacks can break down trust in recommendation systems and lower the recommendation system's performance and quality.

- **Privacy:** this problem arises because giving out personal information usually leads to the system producing better results. However, storing this data can lead to data privacy and security problems. Users are reluctant to give personal data to recommendation systems with privacy problems.

- **Overspecialization and Limited Content Analysis:** this problem arises from the fact that an item's selection is based mainly on the characteristics of the items' attributes, meaning that characteristics that could be more interesting to the user are not considered. Content is sometimes scarce or challenging to represent, and in these situations, relevant items may not be considered unless the items have characteristics that make them stand out. Limited content analysis also leads to over-specialization as they suggest items that are too close to each other and cannot recommend new items.

- **Grey Sheep or Generalization:** this problem occurs when a user's opinions are not a match for any group, and so the system is unable to benefit from the recommendations. This problem can be solved with a pure content-based system since it suggests based on the active user's profile.

- **Sparsity:** this problem stems from the availability of a large amount of data on items in the set of available items, and the lack of interest on the part of users in giving ratings to items can lead to the registration of sparse profiles leading to less precise recommendations.

- **Scalability:** this problem emerges as the growth of nearest-neighbor algorithms shows a linear relationship with the number of items and users. It becomes difficult for a recommendation system to process large data. Different techniques have been used to deal with this problem, including clustering and dimensionality reduction.

- **Latency Problem:** this problem occurs when new items are added frequently to the database and the system only suggests items that have already been categorized, so new items are not suggested. Content-based can reduce waiting times but introduces overspecialization.

- **Evaluation and the Availability of Online Datasets:** this problem arises because evaluating a recommendation system determines, among other things, its quality. The design of evaluation criteria and the selection of suitable evaluation metrics is a key problem in recommendation systems. Another problem is the lack of benchmark datasets to evaluate recommendation systems in a specific domain.

- **Context-Awareness:** this problem, from an operational point of view, encompasses all the categories required for a recommendation system to be deployed, such as location, time, and current activity. It is envisaged that future recommendation systems will be able to have a history that encompasses the user's history in the long and short term so that the recommendation system is the best it can be.

Table 2.1: Comparison of existing problems in Content-Based and Collaborative Filtering Recommendation Systems

Problems			
		Content-Based	**Collaborative Filtering**
Cold Start Problem	Items		X
	Users	X	X
Synonymy			X
Privacy		X	X
Overspecialization and Limited Content Analysis		X	
Grey Sheep			X
Sparsity			X
Scalability		X	X
Latency		X	X
Evaluation and Availability of Online Datasets		X	X
Context-awareness		X	X

These multifaceted problems present a real challenge for recommendation systems to prove their worth by making ever-better recommendations and trying to mitigate all the listed challenges. However, these challenges also make recommendation systems an interesting object of study, and there are still many technological advances to be made.

2.3.3 Types of ratings

Recommendation algorithms are based on ratings that users give to items. This feedback can be given explicitly or implicitly. Explicit ratings are those where users expressly give their opinion on a particular item. For example, giving five stars to a film on a streaming platform. Implicit ratings are those in which we can indirectly infer what users think about a particular item. For example, an implicit rating can be based on the number of clicks a user makes on an item on a shopping/online platform or the time they spend looking at a particular item on the same platform.

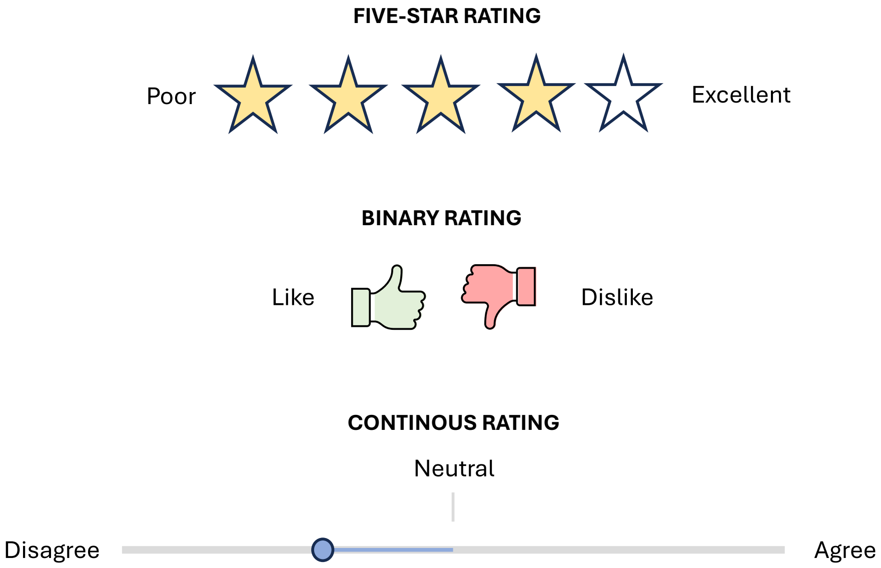


Figure 2.7: Examples of Types of Ratings

Ratings can also be continuous or discrete values (see Figure 2.7). However, continuous feedback is rare. Ratings are usually within a range, given by a discrete set of ordered or binary numbers, such as like or dislike. Examples of such ratings are the 5-point rating scale, where the set can be -2 to 2 or 1 to 5. Ratings can also have categorical values such as Disagree, Neutral, or Agree. There are also unary ratings where the user can only specify their liking. This type of rating is very common in implicit feedback. Sometimes, the ratings scale is also unbalanced, with more positive ratings than negative ones or the other way around.

Both of the feedbacks have limitations. According to [30], in implicit feedback, the user is not deliberately giving their opinion but rather clicking on an article or video, which is then exploited for the system to infer the importance of these propositions for the user. It is necessary to interpret the user's behavior, which can lead to a bias in the recommendation system since a user may look longer at an item because they are interested or simply because they could be distracted. Explicit feedback also has limitations because the information we have can be scarce and often requires extra work on the user's part to specify preferences. It also proves to be a cognitive effort as users must put their preferences into the rating system on a pre-defined scale (e.g., 5-point Likert scale or binary).

2.3.4 Similarity Functions

Similarity functions are of the utmost importance for a content-based recommendation system because they define how similar two items are. We can use various similarity techniques to compute the similarity between two objects. The most commonly used are cosine similarity and Euclidean distance.

- **Cosine Similarity:** in the equation 2.1 , $\mathbf{x}$ and $\mathbf{y}$ are two vectors, and $||\mathbf{x}||$ is the Euclidean norm of the $\mathbf{x} = (x_1, x_2, \ldots, x_p)$ vector, defined as $\sqrt{x_1^2 + x_2^2 + \cdots + x_p^2}$. Conceptually, it represents the length of the vector. Similarly, $||\mathbf{y}||$ is the Euclidean norm of the vector $\mathbf{y}$.

  The cosine similarity measures the similarity between two vectors in the inner product space. It is measured by the cosine of the angle between the two vectors and determines whether they are pointing in approximately the same direction [31].

  This measure varies between 0 and 1. If the value is 0, it means that the two vectors are at $90°$ to each other, i.e., they are not a match. The closer the value of this formula is to 1, the smaller the angle and the greater the match between the two vectors.

$$\text{sim}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{||\mathbf{x}||\,||\mathbf{y}||} \tag{2.1}$$

- **Euclidean Distance:** the Euclidean distance is a metric based on distance and is used to see how similar two vectors are in space (see Equation 2.2). It assesses how close two vectors are by measuring their geometric distance, considering the difference in their coordinates by scale and direction. If the Euclidean distance is 0, it means that the two points are identical vectors. The greater this distance, the further apart the similarity vectors are. This distance tends towards infinity.

$$\text{dist}(\mathbf{x}, \mathbf{y}) = \sqrt{\mathbf{x} \cdot \mathbf{x} - 2 \cdot \mathbf{x} \cdot \mathbf{y} + \mathbf{y} \cdot \mathbf{y}} \tag{2.2}$$

2.3.5 Evaluation

To assess the quality of recommendation systems, it is possible to carry out three different types of experiments: online, offline, and user studies [32].

- **Offline**: This type of experiment is carried out using data collected *a priori* from users classifying items. In this case, it is assumed that the data collected is similar to the user's behavior when the recommendation system is deployed in order to make reliable decisions based on simulation. This type of experiment can, therefore, answer fewer questions about the algorithm's predictive power. It is impossible to measure the direct influence of user behavior in this setting. Normally, this type of experiment is used to test a larger number of algorithms so that only a few candidates can then go on to user studies or online experiments, which are more demanding.

- **User Studies**: User studies are conducted by recruiting several test subjects and asking them to carry out tasks that require interaction with the recommendation system. While the users carry out the tasks, their behavior is collected through quantitative measures, such as the time taken, the accuracy of the results of the task, as well as qualitative questions before and after the task is completed, such as whether the subject found the recommendations relevant or not.

- **Online**: In many recommendation system applications, the system's design wants to influence user behavior. The aim is to measure the change in user behavior when interacting with different recommendation systems. They serve to assess whether users of one system follow recommendations more often than another in order to define which system is best.

Regardless of the type of experiment carried out for evaluation, one of the central objectives of recommendation systems is to predict user preferences accurately. Predicting recommendation systems is by far the most discussed property in the literature. The vast majority of recommendation systems have a predictive engine at the base. This engine can predict users' opinions of items. A basic assumption is that the user favors recommendation systems that provide more accurate predictions. In fact, most applications want to predict the rating a user would give an item (e.g., from 1 to 5 stars). In these cases, we want to measure the accuracy of the system. There are two popular metrics for this evaluation: Root Mean Squared Error (RMSE) and the Mean Absolute Error (MAE) and their normalized and average variations [33].

The RMSE between the predicted and actual ratings are calculated using the formula in Equation 2.3. In this equation, we have the ratings predicted ($\hat{r}_{ui}$) by the system for the test set $\mathcal{T}$ of user-item pairs (u, i) for which the actual ratings (r_{ui}) are known. Typically, these ratings are known because they are hidden in the offline experience or they are obtained from user studies or the offline experience. The MAE is given by the equation in 2.4 and is a way of penalizing errors with the same weight, in contrast to the RMSE, which penalizes larger errors more heavily.

$$\text{RMSE} = \sqrt{\frac{1}{|\mathcal{T}|} \sum_{(u,i)\in\mathcal{T}} (\hat{r}_{ui} - r_{ui})^2} \tag{2.3}$$

$$\text{MAE} = \frac{1}{|\mathcal{T}|} \sum_{(u,i)\in\mathcal{T}} |\hat{r}_{ui} - r_{ui}| \tag{2.4}$$

Many system applications do not cater to user preferences but rather recommend items to users that they can use, such as their likelihood of interaction or engagement. For example, Netflix [1] suggests a list of films that have not been watched before by the active user (user interacting with the system). In this case, we are not interested in whether the system predicts well or poorly the ratings the user will give to this list of films but whether the system can predict whether the user will add these recommended films to their list.

While some metrics such as RMSE and MAE are crucial for predicting the model's accuracy. Recommendation systems also have to take into account user engagement and interaction. This is particularly important in offline evaluations, where assumptions about user interest can introduce biases, such as overestimating false positives.

In the case of an offline evaluation, the assumption is that items that have not been watched are not of interest to the user. This may not be entirely correct since we cannot know for sure, so the number of false positives may be overestimated, jeopardizing the recommendation system. However, systems are evaluated using the following metrics, among others: precision, recall, and f-measures [32; 25].

- **Precision**: measures the number of recommended relevant items (True Positives) over recommended relevant items plus recommended items that are not relevant (False Positives) (see Equation 2.5).
$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{2.5}$$
- **Recall**: measures the number of relevant items recommended (True Positives) over all relevant items, those that were recommended and those that were not (False Negatives), (see Equation 2.6).
$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{2.6}$$
- **F-measure**: measures the balance between precision and recall since a system that maximizes precision can recommend only a small number of items, ensuring that they are all relevant, and a system that maximizes recall can recommend many items, including irrelevant ones, decreasing precision, (see Equation 2.7).
$$\text{F-measure} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}) \tag{2.7}$$

When the number of recommendations that can be presented to the user is predetermined, the most relevant measures can be evaluated at a specific top-k.

- **Precision@k**: it measures the proportion of relevant items among the k recommended items (see Equation 2.8).
$$\text{Precision@k} = \frac{\text{relevant items@}k}{k} \tag{2.8}$$

[1] https://www.netflix.com/

- **Recall@k**: Measures the proportion of relevant items recommended in the top k in relation to the number of relevant items available (see Equation 2.9).

$$\text{Recall@k} = \frac{\text{relevant items@}k}{\text{total relevant items}} \tag{2.9}$$

- **F-measure@k**: it is the harmonic mean between **Precision@k** and **Recall@k**, balancing both metrics (see Equation 2.10).

$$\text{F-measure@k} = \frac{2 \cdot \text{Precision@k} \cdot \text{Recall@k}}{\text{Precision@k} + \text{Recall@k}} \tag{2.10}$$

- **Mean Reciprocal Rank (MRR)**: this metric evaluates the quality of the classification of the most relevant item in the list of recommendations. It is calculated as the average of the position values of the first relevant item for each user. In other words, this metric shows that the higher the Mean Reciprocal Rank (MRR) value, the better positioned the most relevant items are (see Equation 2.11).

$$\text{MRR} = \frac{1}{n_{\text{users}}} \sum_{i=1}^{n_{\text{users}}} \frac{1}{\text{rank}_i} \tag{2.11}$$

- **Discounted Cumulative Gain (DCG)**: this metric takes into account the position and relevance of the items in the recommended list, giving decreasing weights as the items appear in lower positions in the list. This prioritizes the most relevant items at the top of the list (see Equation 2.12).

$$\text{DCG@k} = \sum_{i=1}^{k} \frac{rel_i}{\log_2(i+1)} \text{rank}_i \tag{2.12}$$

- **Normalized Discounted Cumulative Gain (nDCG)**: This metric normalizes the Discounte Cumulative Gain (DCG) divided by the Ideal Discounted Cumulative Gain, which is obtained when the relevant items are ranked perfectly at the top of the list (see Equation 2.13).

$$\text{nDCG@k} = \frac{\text{DCG@k}}{\text{IDCG@k}} \tag{2.13}$$

In conclusion, the evaluation of recommendation systems can be anchored in a diverse set of metrics that have been created to measure different things according to the objectives and challenges of each problem. The RMSE and MAE metrics are crucial for predicting accuracy, and the precision, recall, and f-measure metrics shed light on user engagement and interaction. The inclusion of rank-based metrics such as precision, recall, and Normalized Discounted Cumulative Gain (nDCG) emphasizes the importance of prioritizing the relevance and position of recommendations. Thus, by evaluating these forms of evaluation ,it is possible to gain a comprehensive understanding of the system's performance.

2.4 Related Work

Following, we provide an overview of the state of the art regarding image-based recommendation systems. In particular, we will focus our analysis on the ones using pixels, emotions, or annotations.

2.4.1 Recommendation systems for images

The following section will focus on exploring image-based recommendation systems, in order to understand the different approaches, and the conclusions that have been reached.

The aim of the study [34] is to build an image recommendation system that suggests items according to visual similarity. To do this, the methodology used by the authors was a pre-trained CNNs with Imagenet, in which the feature is retrieved in the fifth layer (the lower the layer, the more accurate are the features) resulting in a high dimensional vector. To reduce the dimensionality of the vectors, the authors used linear Principal Component Analysis (PCA) and non-linear t-Distributed Stochastic Neighbor Embedding (t-SNE) and Uniform Manifold Approximation and Projection (UMAP) techniques. They then used clustering, the k-means algorithm, to group the images according to their visual characteristics. Finally, similarity measurement is used with Euclidean and cosine distances, and the Locality-Sensitive Hashing (LSH) is present in Spotify's Annoy library to find the nearest neighbors. Finally, a recommendation is made of the k closest images, i.e., those similar to the image seen. The recommendations are evaluated using precision, recall, accuracy, and time taken for different k. The results for the precision were made for different ks (10, 20, and 30) and for different categories (shoes, jewelry, dress, and menswear). The highest precision value was observed in the PCA + Cosine and LSH systems for the Jewellery category with $k = 10$, while the lowest precision value was recorded with PCA + Euclidean for the Shoe category with $k = 20$, reaching a value below 0.2. The authors concluded that the methodology applied to the datasets used (UTZappos50k, the jewelry dataset created by the authors, and the DeepFashion "In-shop Retrieval Dataset") produced sufficient results and good quality recommendations were obtained. The dimensionality reduction techniques made the data more amenable to pass to the distance functions, and it has also been proven that Spotify's LSH is applicable to both music and images.

The aim of the work [35] was to investigate how Information Foraging Theory (IFT) could be applied to a content-based recommendation system. IFT states that users are more likely to focus on pieces of information with a more substantial 'scent' (this scent can be estimated with visual and textual clues from the information). The model used to extract the content of the images was the ResNet50 model, pre-trained with ImageNet, which has more than 1000 image categories. A color classifier was also applied, using a clustering algorithm, k-means, which aimed to aggregate the basic HyperText Markup Language (HTML) colors with the predominant colors in the image. For this work, the authors used a dataset comprising images taken from Pinterest [2] of spaghetti bolognese and zoodles (which are noodles made from zucchini). The recommendation process consists of the user entering a name in their folder for

[2] https://www.pinterest.com/

a keyword-based search, and in the second phase, the user chooses a preference, after which the system retrieves similar items. These images were labeled with a title and description, and these annotations are important for improving the system as they help to strengthen the predominant visual characteristics. The study found that the scent of information influences user preferences, with more relevant images attracting more attention and interaction (such as bolognese and zoodles). The study concluded that the dataset (1116 images) was too small to draw further conclusions but that applying IFT to recommendation systems could be viable.

The aim of the study in [36] is to implement a content-based recommendation system based on features retrieved by pre-trained CNNs. The methodology of this work was divided into two phases: an online phase and an offline phase. In the offline phase, the features of the images were extracted and stored using two CNN models pre-trained on ImageNet, SqueezeNet, and ResNet18. In addition to the CNNs, the study also used traditional feature extraction methods, such as color and texture, to serve as a comparison for CNNs. Once extracted, the features were stored for use in the online phase. In this phase, the user interacted with the images, and the system calculated the most similar image using the features extracted by the CNN and the Euclidean distance or applied the color and texture models to recommend the most similar image. The system then collected the quality assessment of the search. The method proposed by the authors performed highly in most categories. The average accuracy in the Dinosaur category stood out at 0.971, Bus at 0.957, and Horse at 0.928. The results are also consistent for the Building category (0.936) and Flower (0.914). The CNN-based approach outperformed other methods, such as Rasho, which achieved lower performance in categories like Airplane (0.489) and Mountain (0.398). In some categories, such as bus and flower, Singh's method was more accurate (1.0) than the opposite method. This study concluded that pre-trained CNN are a capable tool for content-based recommendation systems.

The goal of the work [37] is to develop an image-based recommendation system that uses common image annotation techniques to address sparsity and accuracy problems. The recommendation process begins with feature extraction using Scale-Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF), and Local Binary Patterns (LBP) to extract visual information from the images uploaded by the user. Bag of Words is then used to represent the images as a set of visual words. Next, k-means clustering is performed on these sets, and a histogram is generated with the features of each image. The system recommends the images with the most similar characteristics using k-NN to find the 10 closest images to be recommended. Finally, the user can indicate which image interests them the most. The study was conducted on a dataset of shoes of various types and brands, and it was concluded that the system achieved an average recommendation accuracy for shoe classes of 69% for the SIFT descriptor, 71% for the SURF descriptor and 62% for the LBP descriptor. The authors considered these results to be good, given the variety of the dataset.

The focus of this study [38] is the implementation of a content-based product recommendation system based on images using CNN. For image recommendation, the Support Vector Machine (SVM) model recognizes the category of the input image using edges as features. This type of feature is used as input since each edge represents a sudden change in color. Similarity is then calculated using cosine distance;

although the study also tests Euclidean distance, it did not perform as well. Finally, a neural network is used to determine the probability of two images being similar. This network acts as a binary classifier. Ultimately, the image most similar to the input image is recommended. For this study, a Kaggle[3] dataset with 2,000 images of fashion products was used. The results compare the accuracy of three different models, CNN, Neural Network (NN), and CNN+Pre-Trained. The highest accuracy was achieved by the last model (CNN+Pre-Trained) with 71%, followed by CNN with 67% and finally NN with an accuracy of 66%.

The study presented in [39], the study aimed to give a better understanding of the representational power of CNN commonly adopted by the community when integrated into Visually-aware Recommender Systems (VRS). They used AlexNet, VGG19, and ResNet50 for the VRS and used different visual models: Visual Bayesian Personalized Ranking (VBPR), Deep Learning for Style-based Recommendations (DeepStyle), Visual Neural Personalized Ranking (VNPR), and Adaptive Collaborative Filtering (ACF). The fashion datasets were Amazon [4] Baby and Amazon Boys and Girls. The authors concluded that with a deeper Image Feature Extractor (IFE) such as ResNet50, there is a higher quality and quantitative ability to recommend more diverse products when considering both the recommendation perspective and the visual appearance. For the Amazon baby dataset, the models showing the best results were DeepStyle with ResNet50 with a recall of 0.2195 and an Area Under the Curve (AUC) of 0.6400. The VBPR model with ResNet50 also obtained high AUC values of 0.6475 but had a slightly lower recall of 0.2063. The collaborative filtering was the model with the worst results, with an AUC of 0.5544. For the Boys & Girls dataset, the VBPR model with ResNet50 again showed the best results with an AUC of 0.6606, followed by DeepStyLe with 0.6402. For both models, the recall was 0.1250, solidifying VBPR with ResNet50 as the best model.

Dominguez's study aims to research the effect of explaining suggestions in artistic images [40]. They conducted a user study on Amazon Mechanical Turk on three different interfaces and two different algorithms. The interfaces have explanations based on image similarity and explanations based on visual characteristics. The two algorithms used were Deep Neural Network (DNN) and Audio-Visual System (AVS). For the DNN model, image similarity was based on the features extracted. The output vector representing the image is usually called an image's visual embedding. The visual embedding in their experiment was a vector of features obtained from an AlexNet. For AVS, they obtained a vector of explicit visual features of attractiveness using the OpenCV software library3: brightness, saturation, sharpness, colorfulness, naturalness, entropy, and Red, Green, Blue (RGB) contrast. In the study, the images used were provided by the UGallery online shop. The study concluded that the recommendation algorithm DNN is perceived as better than ACF in most dimensions, demonstrating that the algorithm's accuracy significantly impacts user perception, even when the features are not explainable.

The article [41] describes a recommendation system for images on unsupervised probabilistic machine learning for e-commerce. The study was carried out on a dataset of fashion product images.

[3] https://www.kaggle.com

[4] https://www.amazon.com/

The proposed recommendation system uses characteristics of product images, including colors, textures, and shapes. These features are extracted from the images using techniques such as Fourier and Pyramid Histogram of Oriented Gradients (PHOG). The features are then processed using dimensionality reduction techniques, PCA and Expectation Maximization (EM) PCA, to transform them into a lower-dimensional space. Finally, the system uses clustering algorithms, such as K-means (the best being PSVD-K-means++), to group images with similar characteristics and calculate distance measures to recommend visually similar products. Dimensionality reduction with PCA-Singular Value Decomposition (SVD) transformed the dataset from 14,400 dimensions to 144 principal components, achieving 90.01% variance.

The studies presented show that machine learning techniques, such as pre-trained CNN and clustering algorithms, combined with similarity methods, are effective for implementing recommendation systems based on visual content. Despite the differences in the approaches and datasets used, all the works emphasize the importance of representing visual characteristics of images efficiently, either through embeddings, traditional characteristics (such as color and texture), or a combination of both. In addition, dimensionality reduction methods and the use of distances such as Euclidean and cosine are recurrent, standing out as critical tools for improving the efficiency and accuracy of recommendations. Finally, the studies reviewed indicate that incorporating emotional personalization may lead to improvements in recommendation accuracy and user engagement. While more research is needed to generalize these findings, the evidence points to the growing relevance of emotions in the design of recommendation systems.

2.4.2 Recommendation systems using emotions

The aim of this section is to review the state of the art regarding the use of emotions in recommendation systems.

The authors in [42] developed a music recommendation system based on identifying the personality traits, moods, and emotions of a single user. The approaches used include recognizing the user's personality based on the behavior on social networks (likes, status updates, photos, etc.) and mapping personality traits in the Mehrabian's Pleasure-Arousal-Dominance (PAD) emotional space. The mood analysis is based on the last objects accessed; for each song listened to, features of the song are extracted using the Mel-Spectrogram and Visual Geometry Group (VGG) models, and the detected mood (arousal and valence). For recommendation, a content-based recommendation system is used in which an emotional mapping of audio objects in PAD space is made. The dataset is then organized using a Ball-tree data structure, and items are suggested based on the minimum Euclidean distance concerning the user's mood. Recommendations take advantage of two types of information: user profile and the content-based strategy. The system's recommendations achieved a recall of 1 for a list of 85 or more recommended items.

In the study by [43], the authors developed an Emotion Aware Recommender System (EARS) for recommending online news. Users' self-assessed emotions are given explicitly through a widget at the end of the news, and these emotions are then incorporated into the system using PAD. The recommendation system developed is a hybrid based on three types of systems: content-based, collaborative filtering, and knowledge-based. It is concluded that incorporating pleasant emotions in collaborative filtering recommendations consistently outperforms other algorithms.

In [44], the goal is to understand which type of affective labeling works best for an already-made multimedia content-based recommendation system: an explicit approach, an implicit approach, or generic metadata. For the implicit approach, the authors used facial recognition for affective labeling, training the model with a subset of the Cohn-Kanade dataset and the LDOS-PerAff-l dataset. Both datasets consist of annotated video clips. For the explicit approach, the researchers asked the user to give an affective label to the image being observed explicitly. Thus, the labels are unequivocal, but their veracity can be questionable. In the case of the generic metadata, the item is characterized by its content (genre and visualization time). Explicit affective labeling showed the best results with a precision in SVM of 0.68, followed by implicit affective labeling with SVM with a precision of 0.64, and finally, generic metadata with a precision of 0.61 for SVM. The study concludes that the most effective approach is using explicit feedback. However, implicit feedback works better than using generic metadata.

This article [21] investigates ways of incorporating emotions into the content-based recommendation process. In addition to using emotions to generate personalized recommendations, the system also considers the user's affective state when the recommendation is made by analyzing text messages published on social networks (e.g., Facebook and Twitter). The IBM Bluemix Tone Analyzer is used to categorize these emotions into Ekman's six basic emotions for a song the user shares. The methodology involves creating an emotional profile for each user by collecting emotional data. Emotions are characterized in three ways: using the properties of the audio, lyrics, and music genre. These emotions are then converted into vectors to calculate similarity using cosine similarity for both items and affective states. Based on this, a list of recommendations is generated, with the songs having the highest scores being recommended. It is important to note, however, that the emotional content expressed in social media posts does not always correspond to the user's actual emotional state, but may instead reflect specific reactions to the subject matter of the post. This represents a potential source of noise in the emotion detection process. The article compares the performance of the proposed algorithm with classic content filtering, using the same algorithm but pre-fitting it based on the user's predominant emotion. It also tests a logistic regression method for predicting classifications (i.e., if the user likes or dislikes the music). The results show that the model in the article Emotion-aware Music Recommender System (EMERS) outperforms all of them for the parameters presented (alpha=0.5 and beta=0.5) and got a HitRate@10 of 0.090. The results demonstrate that the proposed model outperforms the other algorithms in terms of HitRate@n.

In [45], the aim of the study is to understand how a system based on affective metadata, which refers to data describing users' emotional responses to a particular item, improves the performance of a content-based system. In this study, the aim is to recommend images based on the International Affective Picture System (IAPS) and data (explicit classifications and time spent viewing the images as well as the answers to the questionnaire) collected from 52 participants. The methodology for content-based recommendation is based on the correspondence between item profiles and user profiles. This study compares two types of metadata for items: generic metadata, such as image category and viewing time, and affective metadata, which are emotional responses to the image presented, that are represented in Valence, Arousal, Dominance (VAD) space. The user's profile stores the user's preferences according to their emotions, and this user model was created using machine learning algorithms such as decision trees. For the recommendation, the algorithm compares the item profiles with the user profile using a similarity function and also uses a categorization algorithm that classifies each item as relevant or not relevant to the user according to their profile. The sets of recommended and non-recommended items are then generated based on the estimates of the binary classifications and probabilities. Four algorithms are used to predict the ratings: AdaBoost, C4.5, Naive Bayes, and SVM. The results presented are summarized in two datasets that illustrate the impact of incorporating affective metadata into the recommendation system. The COHN-KANADE dataset achieved a maximum accuracy of 0.95 for surprise emotion and a minimum of 0.90 for unknown. However, for the LDOS-PERAF dataset, we have a maximum accuracy of 0.65 pa and a minimum accuracy of 0.60. The researchers concluded that the use of affective data improves the performance of a content-based recommendation system.

The articles presented show how incorporating emotions can improve recommendation systems, both in the context of media such as music, images and in online news. Emotion analysis using explicit techniques, such as direct feedback, or implicit techniques, such as facial recognition and text analysis, is proving to be a promising strategy for personalizing recommendations according to users' emotional states. In addition, using models based on emotional spaces, such as PAD and VAD, makes it easier to map the emotions of users and items, allowing for more accurate similarity calculations. These approaches demonstrate the value of using affective metadata in content-based recommendation systems, often surpassing traditional strategies that rely exclusively on generic metadata. Finally, the results reinforce that emotional personalization not only improves the accuracy of recommendations but also offers a more engaging and satisfying experience for users, consolidating the role of emotions as a crucial component in developing modern recommendation systems.

2.4.3 Recommendation systems using metadata

Following, we present the studies that have been carried out that use metadata, specifically image annotations, to make recommendations.

In article [46] authors aimed to evaluate the accuracy of a tag-based recommendation system by comparing collaborative, content-based, and hybrid approaches to determine which yields the best results. The study used two types of data: bookmarks from the ECML-PKDD 2009 Discovery Challenge and BibTeX references. The collaborative approach was applied to identify which tags the user prefers, comparing these preferences with a user-compared social index to provide recommendations. In the content-based approach, the system analyzed different fields of the resource, such as title, Uniform Resource Locator (URL), and description, assessing the frequency of each term to compare tags and recommend the articles with the greatest similarity. The systems were evaluated based on accuracy, recall, and F1-score, the precisions of the best recommendation systems was the following the content-based recommendation system obtained a precision with the best parameters of 16.10% and collaborative filtering a precision of 11.20%. Their results were compared to those of other systems in the ECML-PKDD 2009 Discovery Challenge. Subsequently, a hybrid cascade and fusion approach, combining the two methods, was implemented. The fusion strategy achieved the best results, obtaining a precision of 18.59%.

This study [47] explores the possibility of making content-based recommendations on Twitter, enabling users to discover topics similar to those they are writing about. To achieve this, the authors used a natural language processing library to recognize noun phrases and search for similar ones. The phrases are also compared using Naive Bayes to calculate the probability of a tweet being associated with a particular topic for which the algorithm has been trained. The nominal detection approach demonstrates high accuracy (88%). Still, it produces fewer recommendations due to the infrequent exact match between noun phrases. At the same time, Naive Bayes achieves balanced accuracy (65% for the tokenization of unique words and 75% for the tokenization of nominal phrases) with a higher number of recommendations.

The aim of this article [48] is to develop a sports content-based recommendation system that helps users find the type of sport that best suits their preferences. The system collects data about the user, such as their motivations, exercise level, name, age, and gender, and then compares this data using Term Frequency-Inverse Document Frequency (TF-IDF) to generate recommendations based on the user's motivations and the sports magazine documents that have been collected. The results show that the method is viable for making recommendations, as the accuracy using this methodology is high (86.90%).

In conclusion, the studies presented show different approaches to recommendation systems applied to areas such as images, tags, and content. Each approach seeks to solve challenges such as data scarcity and accuracy. The image-based recommendation system, using techniques such as feature extraction and clustering, proved to be efficient for visual tasks. In the case of the tag-based system, a comparison was made between collaborative and content-based methods, with the aim of improving accuracy. The research on Twitter explored the use of natural language processing to recommend topics similar to the content published by users. Finally, the sports recommendation system demonstrated the effectiveness of personalized data and the application of TF-IDF and cosine similarity to provide more accurate suggestions. These studies highlight the advances in recommendation technologies, which are applicable to a variety of areas.

2.4.4 Recommendation systems in care and therapy

This section discusses recommendation systems with the aim of being used in care and therapy. This is relevant to this work since images have a great impact on emotional regulation, evoking emotions that can be used in therapy.

NowAndThen [49] is a photo recommendation tool designed to enhance user experience on Social Network Sites (SNS) and evoke emotions during reminiscing. NowAndThen employs an image retrieval system inspired by the winning approach at TRECVID Instance Search 2014. The system comprises offline indexing and online searching stages. The offline search used a Hessian-affine detector and one described by SIFT. The complexity is reduced by an approximate k-means algorithm, then the characteristics are quantified by making a cluster, then a TF-IDF weighting is made to give greater importance to less frequent words and an inverted index is made. For the online search, the same steps are taken up to the TF-IDF weighting, after which there is a distance calculation, a spatial check applied by the RANSAC algorithm to the 20 closest images, and a metadata match. This concept employs an algorithm to avoid triggering negative reminiscence, assuming that people only share on social media images they like, which does not prevent the image from being associated with negative emotions. The study lacked quantitative comparisons between the approach used and those that used randomized reminiscence triggers. In addition, future studies should have longer-lasting evaluations (days to weeks); this study only had a 30-minute assessment. The study only involved 20 participants with an average age of 20, which is limiting given that this is a very specific niche of users. The algorithm tested for image retrieval, tested with the Oxford Building Dataset, achieved a Mean Average Precision (MAP) of 82.4%; this result does not include the use of metadata.

I-CARE [50] is an activation system designed for both professional and informal caregivers to engage People with Dementia (PwD) in cognitive, physical, and social activities through activation contents (e.g., pictures, videos, games, and more). Following each activation, users provide explicit assessments using smiley ratings and optional voice comments. The system incorporates emotion recognition based on the user's facial expressions. The system uses a variety of sensors, including the camera, microphone, accelerometer of the tablet that users use, and also an Empatica E4 bracelet to measure movement, and a considerable corpus; however, recommendations are initially based on general biographical information, meaning that the cold-start phase is relatively long and users may give up before the personalized recommendations. The pilot study is limited by the small sample, 29 participants and their tandem partners, of which only 25 participants finished the study. The study has limited statistical power of analysis and should be supplemented with additional data from comparable contexts (e.g., users living in their own homes) and a control group. In terms of results, with regard to the average ratings of the content recommended by the system, a random baseline was compared, which selects content items randomly from the most popular content, and I-CARE, which is a content-based filtering algorithm, with the baseline obtaining values of 0.74 and I-CARE 0.8 of average evaluation, being zero the worst and one the better.

CAREGIVERSPR-MMD [51] is a project focused on building a digital platform catering to PwD and their caregivers. The platform aims to provide advanced, individually tailored services, enabling users to thrive in the community for an extended duration. Even though this study is currently being piloted by 600 dyads in 4 countries, the data is not available at the time, so we can not measure the efficiency and the value of the recommendation system. The HRS recommendation system in this study has four elements in its architecture: a manager that combines the predictions made by the two recommendation systems and applies filtering rules both before and after processing to ensure that the system works properly. There is also the recommendation system itself, where the content-based uses both the characteristics of the interventions and the feedback to predict the user's interest in new recommendations. In the content-based recommendation system, we have feature extraction, the creation of a user profile that collates feedback, feature matching, and rating prediction. The rule-based recommendation system uses a classification of health professionals' interventions to ensure that recommendations are also medically relevant. The adapters are an internal bridge between the models and the C-MMD platform, so the system is independent of changes to the data schemas. Finally, there is an endpoint where the C-MMD platform accesses the recommendations. The study does not have specific results as the data is in the pilot phase and, therefore, not yet available for analysis.

The article [52] focuses on the therapeutic decision to psyche out an autoimmune skin disease. The authors used a database from a clinic of patients with this condition in order to evaluate the different therapies and recommend the most appropriate one for each patient. A collaborative filtering recommendation system and a recommendation system based on demographic data were developed. The collaborative recommendation system shows a better performance in predicting recommendation results, however the sparsity of the data, namely the lack of therapy history, prevents the usability of this system for the entire database. The recommendation system based on demographic data performs less well on average, although it can embrace more queries and does not rely solely on history. To carry out the evaluation, the consultations were divided into two groups in which at least one of the recommended therapies was applied and those in which the recommended therapies were not applied. Therapies with an affinity equal to or greater than 0.5 were considered to have a good result. RMSE and accuracy were used to compare the two systems. The authors concluded that the data was limited and that a better solution would be to make a hybrid system with both systems tested, where an accuracy of 79.78% was estimated.

This article [53] proposes an intelligent health recommendation system using a restricted Boltzmann machine (a generative model that learns to represent data efficiently) and a CNN. The recommendation system in the article is hybrid, with a user profile (content-based) and integration of ratings and preferences from other users (collaborative filtering). However, the focus of this article is on improving accuracy by combining the restricted Boltzmann machine and CNN for feature extraction, which is why the type of recommendation System used is not explicitly mentioned. The study was conducted with a dataset of more than 10,000 different patient assessments, rated from 1 to 5, for more than 500 healthcare facilities. Thus, the authors concluded that the combination of the restricted Boltzmann machine with CNNs has an accuracy of 95% when compared to other feature extraction methods, such as matrix factorization, which has an accuracy of approximately 94%.

The paper [54] presents a systematic literature review of personalized recommendation systems for mental health interventions. Its aim is to provide guidance for future work. The paper concludes that data availability is often scarce or noisy and that there is a great need to protect the privacy and confidentiality of sensitive patient data. It also highlights the lack of research into the actual effectiveness of recommendation systems in mental health interventions. In addition, the article reports that the accuracy of recommendations is improved by incorporating contextual factors such as mental health history and emotional states. And the way to mitigate the problems of cold-star and sparsity is to incorporate data from external sources or hybrid systems. The exploitation of advanced techniques such as deep learning and reinforcement learning and the integration of multimodal data can further improve the accuracy and personalization of recommendations.

The studies show the potential of recommendation systems in specific domains, such as emotional memories and support for PwD. Tools such as NowAndThen and I-CARE demonstrate the ability to personalize experiences based on users' preferences, emotions, and needs. However, they face challenges such as limited samples, short-term evaluations, and difficulties in the initial personalization phase, which can affect user adherence. Despite these limitations, the results indicate that with greater methodological robustness and larger studies, these systems could offer significant contributions to improving users' quality of life and well-being.

2.5 Summary

This chapter provides an overview of emotions, recommendation systems, the binding between both, and the uses of this type of algorithm. Emotions are presented through different theories: discrete, where it is argued that there are basic and universal emotions, and dimensional, where emotions are in a continuous space using dimensions such as valence, dominance, and arousal. Recommendation systems are algorithms that suggest items to users based on their preferences, previous interactions, or item characteristics. The three main types are content-based recommendation systems, collaborative filtering, and hybrid systems. Emotions can be incorporated into these systems to improve the personalization of recommendations, and they can be used to analyze the user's emotional state to recommend the most appropriate items for their current mood. Recommendation systems have been used in various areas, from entertainment to e-commerce, and play a particularly important role when used for care and therapy roles. Despite the promising results presented in the literature, several gaps remain underexplored. Many studies rely on implicit emotion detection methods, which may not fully reflect users' actual emotional states, and few systems have been evaluated with real user interaction. Moreover, the integration of explicitly reported emotions into recommendation processes is still limited. This thesis aims to address these gaps by proposing a content-based recommendation system that uses users' self-reported emotions to personalize image suggestions. In doing so, it contributes to a deeper understanding of how emotions can be directly leveraged to improve recommendation quality and user experience.

Chapter 3

Methodology

Content-based recommendation systems rely solely on user interactions with items and on the features of the items. In this dissertation, the items are images. In Figure 3.1, we can see an overview of the methodology presented. The process begins with an item set and goes through feature extraction, which calculates similarity and scores according to user preferences, resulting in a recommendation list. Lastly, the system was evaluated. This methodology was chosen because it can exploit different image characteristics to make recommendations. The methods in question will be further analyzed in the following sections.

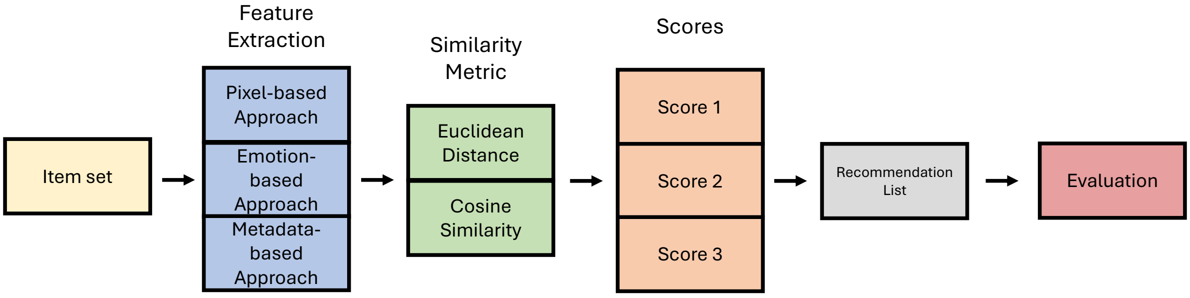


Figure 3.1: The diagram illustrates the methodology used to develop the multiple content-based recommendation systems, including the feature extraction approaches (pixel, emotion, and metadata), similarity metrics (euclidean distance and cosine similarity), scoring, generation of the recommendation list, and evaluation.

3.1 Overview - dataset

The dataset was developed by the research team of this work since, to the best of our knowledge, there were no image datasets publicly available that have been explicitly annotated by various users with emotional content. This dataset consists of images from different sources:

- Pexels [1]: Pexels is a website that offers high-quality, high-resolution photographs, many of them with captions so that they can be used for free under Pexels' terms and license.
- IAPS Mikels [55]: dataset annotated with categorical emotions comprising 330 photographs from IAPS, 133 of which are negative and 187 are positive. The IAPS [56] dataset contains 1182 images and was developed to evoke emotions through photographs.

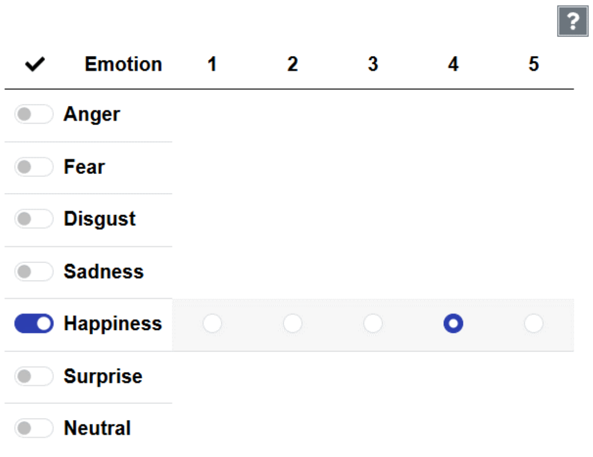
The Table 3.1 presents a summary of the dataset used for this dissertation. It includes the number of users who participated in the study, the total number of interactions recorded, the number of unique images interacted with, and the total number of images used.

Metric	**Value**
Number of users	163
Number of interactions	2445
Number of interacted images	553
Total number of images	2885

Table 3.1: Summary of the dataset used for this dissertation: number of users, interactions, and images.

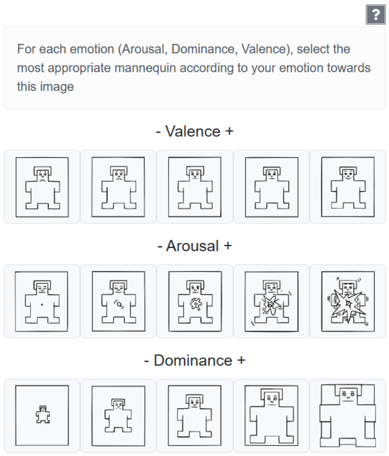
The dataset on which this dissertation is based was collected through an online survey to analyze user interactions with the images. This dataset has 163 users who participated voluntarily, interacting with 553 of the 2885 images present. So, 2500 images come from Pexels and 385 from Mikels. Before the questionnaire, users were asked for demographic information, such as gender, age, etc. Afterward, they were asked to rate the images from 1 to 5, where 1 represents a weaker emotion and five a stronger emotion, from the available emotions (anger, fear, disgust, sadness, happiness, surprise, and neutral) (see image 3.2). A binary rating of whether or not they liked the image in question was also collected. When there was no value for a given emotion in the questionnaire, its absence was assumed. In addition, the Self-Assessment Manikin (SAM) (see Figure 3.3) was also used to obtain dimensional emotions. An extensive analysis can be found in the appendix A.

[1] https://www.pexels.com/



✓	Emotion	1	2	3	4	5
○	Anger					
○	Fear					
○	Disgust					
○	Sadness					
●	Happiness	○	○	○	●	○
○	Surprise					
○	Neutral					

Figure 3.2: Categorical Emotions Survey



For each emotion (Arousal, Dominance, Valence), select the most appropriate mannequin according to your emotion towards this image

- Valence +

- Arousal +

- Dominance +

Figure 3.3: Dimensional Emotions Survey

Figure (3.4) shows a slight dataset bias towards positive ratings. Relevant items (items rated as 1 liked by the user) are important for recommending items similar to them, while non-relevant items (items rated as 0 disliked by the user) help avoid recommending similar items.

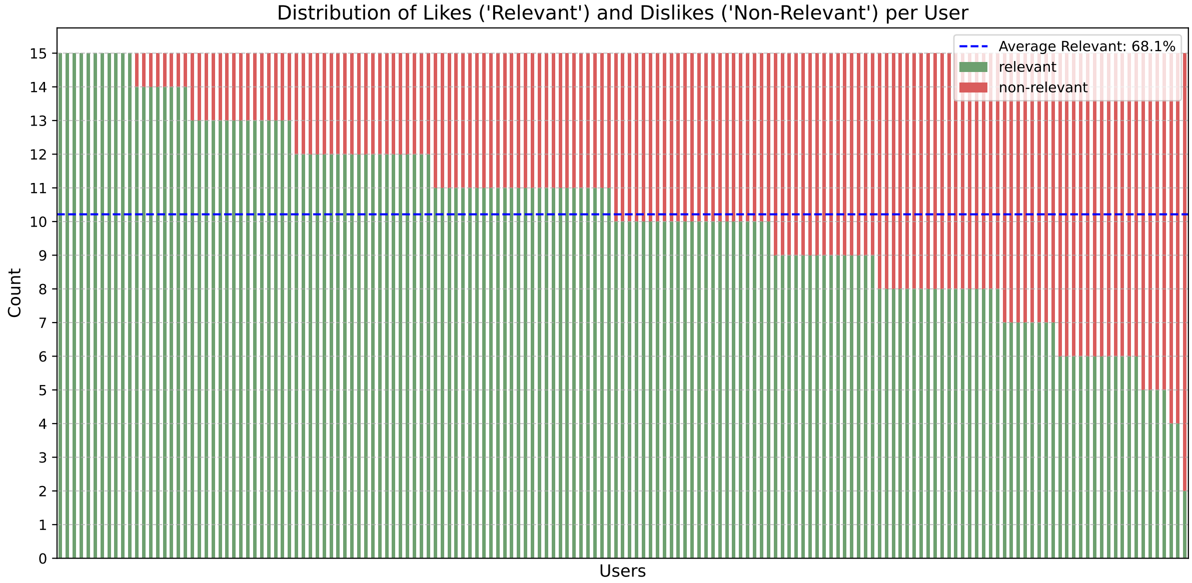


Figure 3.4: Distribution of likes ("relevant") and dislikes ("non-relevant") per user. Each bar in this graph corresponds to a user. The green part of the bar represents the items the user found relevant, while the red part represents the non-relevant items. The blue line indicates the average percentage of relevant items per user (68.1%), approximately ten items.

3.2 Feature Extraction

A content-based recommendation system is based only on a user and the quality of the characteristics of the items we want to recommend and the items already seen by the user.

The features of the three content-based recommendation systems were selected in various ways. For images, CNN were used; for emotions, we used the values provided by each user; and for annotations, the descriptions provided by Pexels were employed, with missing descriptions supplemented using Bootstrapped Language-Image Pre-training (BLIP). In the following subsections, we present the pixel-based, emotion-based, and metadata-based approaches.

3.2.1 Pixel-based Approach

We tested different models to extract the pixel-based features VGG, Residual Network (ResNet), Inceptionv3, and Xception). Visual information is essential for image recommendation because it contains details such as texture, color, and shapes often unavailable in textual descriptions or metadata. In addition, pixels are the fundamental unit of the image, capturing color, intensity, and spatial patterns. By analyzing pixels, we can identify the intrinsic characteristics of images, making recommendations more precise. The choice of these architectures was motivated by their unique advantages and distinct architectures, guaranteeing a complete representation of image characteristics. VGG is known for its simple and uniform architecture, based on sequential convolutional layers. This structure enables deep feature extraction and has demonstrated good performance in capturing fine-grained image details, such as textures and patterns. ResNet captures more complex features because of the residual connections that make it easier to capture them. InceptionV3 introduces innovative modules capable of processing features at different scales and different levels of complexity, balancing computational efficiency and feature quality. Xception builds on the principles of Inception by using depth-separable convolutional layers, offering an optimized architecture for feature extraction. By integrating all these models, this work allows for the comparison of different feature extraction strategies, contributing to a broader evaluation of their performance. All the neural network models used were trained with ImageNet ([57]), which is an image database containing over 3.2 million images annotated with more than 5,247 categories. Neural networks use this dataset to learn the general and complex characteristics of objects and contexts. The models in question will be described in the following paragraphs.

- **VGG**: network [58] developed by the VGG. This network is known for its straightforward and uniform architecture, which has proven effective for image classification tasks. The architecture of these CNNs is made up of:
  - **Convolutional layers**: where kernels are used (matrices of numerical values used in the convolution layers to extract the characteristics of the layer). In this case, the kernels are 3x3 and are used to extract edges, textures, and patterns from the image. These features are then encoded in feature maps.

  - **Spatial resolution reduction**: max-pooling layers are used to reduce the spatial resolution of the feature maps, preserving the most relevant information.
  - **Classification**: Dense, fully connected layers that receive the feature maps given by the above step to classify the image.

  The images were resized since the input for this model is RGB images with a fixed size of 224x224. The last layers of the model, the dense layers, were also removed since their purpose is classification, which is not relevant for extracting features. So, each image results in a one-dimensional vector of 512 values. The vectors are then used to make comparisons between them. The difference between the VGG16 and VGG19 networks is the number of convolutional layers used, with 16 convolutional layers and 19, respectively. Both networks perform well depending on what we want to achieve. According to the network's developers, VGG19 is more accurate and has a higher computational cost, while VGG16 has good accuracy and low computational costs. By using the two networks, we were able to compare the efficiency of both and determine whether increasing the depth of VGG19 significantly improves the recommendations or whether VGG16 has the same efficiency level at a lower computational cost.

- **ResNet**: the ResNet [59] architecture was built to overcome the degradation problem (this problem consists of very rapid degradation in accuracy when the training phase saturates) encountered when training very deep neural networks. This architecture consists of :

  - **Convolutional layers**: a sequence of convolutional layers inspired by VGG networks in which there are 3x3 filters (or kernels).
  - **Shortcut connections**: These consist of connections that ignore one or more layers by adding the layer's input to the output of the stacked layers. These connections implement residual learning. These connections generally perform an identity mapping without adding extra parameters or computational complexity.
  - **Linear projections**: When the input and output layers of the layer do not match, a linear projection is used in the shortcut links so that the dimensions match.
  - **Bottleneck architecture**: for the deepest networks (e.g. ResNet50, 101 and 152), a bottleneck design is used to reduce computational complexity. Each residual function comprises three layers: 1x1, 3x3, and 1x1, which correspond to filters where 1X1 only filters each pixel at a time, and 3x3 looks at a region of 9 pixels per window.

  ResNet networks do not use dense (fully connected) layers for classification. Instead, they use a technique called 'Networks on Conv feature maps' (NoC), which performs the classification directly from the convolutional feature maps. In our work, an image with a 224x224 input configuration will be used as input, although this is not expressly required. The model implemented does not use NoC convolutional layers since the objective is not classification but max pooling. So that the characteristics are summarised to keep the most important information and prepare the data to return a one-dimensional vector, therefore, it returns a vector with one dimension of 2048 values for each image. The three ResNet models were used to analyze the trade-off between the depth of the model, its accuracy, and its computational cost.

- **InceptionV3**: the inceptionV3 networks [60] are a new version of the inception architecture and the GoogLeNet, derived from research. The networks are used for image classification and are high-performance CNN for image classification tasks. Its computational cost is moderate. These networks are made up of the following layers:
  - **Traditional convulotional modules**: to process and extract low-level features.
  - **Inception modules**: use three types of modules, each designed to capture features of different scales and levels of complexity. The main feature of the inception modules is that they use different sizes of convolutional filters and a maximum pooling operation. These modules are used in 35×35 grids with 288 filters each.
  - **Factorized Inception Modules**: The convolutions are factorized on a 17×17 grid with 288 filters each. This technique is based on spatial aggregation and can be performed on smaller embeddings without much or any loss of representational power, reducing the computational cost and the number of parameters without sacrificing feature extraction.
  - **Inception Modules with Spatial Factorisation in Asymmetric Convolutions**: This layer consists of factorizing $n \times n$ convolutions into a $1 \times n$ convolution followed by an $n \times 1$ convolution. This scheme is applied to two Inception modules on an 8x8 grid with 1280 filters, reducing the computational cost and promoting high-dimensional representations.
  - **Dense layers**: an 8×8 pooling layer to reduce the dimensionality of the representation, a fully connected layer to combine the features extracted by the inception modules, and finally, a softmax layer, which is applied to produce the probability for each class and carry out the image classification.

  In this dissertation, the images were resized to the dimensions of 299×299, because this is the required input size for this pre-trained model. Subsequently, the dense layers were stripped, followed by max-pooling to transform the features into a one-dimensional vector comprising 2048 values.

- **Xception**: The Xception [61] architecture was developed based on the relationship between traditional convolutional, inception modules and depth-separable convolutions. Based on the observations of inception modules for depth-separable convolutions, it was postulated that it might be better to use depth-separable modules instead of inception modules. Their architecture is based on
  - **Linear stack**: this is a linear stack of depth-separable convolution layers with residual connections.

- **Logistic regression layer**: used to classify images.

The architecture is similar to the InceptionV3 architecture. However, the layers responsible for classification were removed. Thus, this architecture gives us vectors of 2048 values, which will be compared with each other.

3.2.2 Emotion-based Approach

Emotions were used because they are linked to user engagement and satisfaction. By recommending images based on emotional responses, we want to demonstrate personalized and meaningful recommendations for the user. The emotion vectors for each image were made from the users' responses on the surveys. For each image, an evaluation of the categorical and dimensional emotions is made. An average of all the users' emotional responses was made to make a global classification for each image. In this classification process, since the recommendation system is content-based, each recommendation is generated individually for each user. To avoid bias, the user's own ratings are removed from the data used to generate their recommendations — this prevents the system from ”learning” from the very information it is trying to predict. However, in some cases, an image was rated by only one user — an occurrence that happened 122 times in the dataset of 553 images. When this happens, and that user's ratings are removed for evaluation, the image ends up with no ratings at all. In those cases, the image is represented by a zero vector, which negatively affects the similarity calculation. Based on these classifications, we generated vectors representing emotions in two forms: categorical and dimensional. These vectors encode emotional information in a predefined order, as shown in Equations 3.1 and 3.2.

$$\mathbf{E}_{\text{categorical}} = \left[A_{\text{nger}}, F_{\text{ear}}, D_{\text{isgust}}, S_{\text{adness}}, H_{\text{appiness}}, S_{\text{urprise}}, N_{\text{eutral}}\right] \tag{3.1}$$

$$\mathbf{E}_{\text{dimensional}} = \left[V_{\text{alence}}, A_{\text{rousal}}, E_{\text{xcitement}}\right] \tag{3.2}$$

3.2.3 Metadata-based Approach

By using textual information about the images in the recommendation, we can capture semantic meanings that cannot be extracted directly from the pixels because they are abstract concepts. BLIP was used to caption the images that did not have annotations. BLIP [19] is a framework that aims to unite computer vision with natural language. This methodology aims to generate text from images and vice versa. The architecture of this model has three main components:

- **Unimodal Encoder**: this encoder processes images and text separately; it is trained to use the Image-Text Construal (ITC) loss minimizer; this encoder aligns the visual and textual representations, ensuring consistency between the two modalities.

- **Image-Grounded Text Encoder**: it incorporates additional cross-attention layers to capture interactions between vision and language. It is trained so that image-text matching loss is reduced, which differentiates positive image-text pairs from negative ones.
- **Image-Grounded Text Decoder**: this component replaces the bi-directional direction of the self-attention layers with a causal self-attention layer. This decoder is pre-trained in lossy language modeling to generate captions for the given images.

This model was pre-trained with several sets of images (COCO Captions, a dataset with images and their captions associated with Microsoft COCO; Visual Genome, which is a dataset rich in semantic annotations, including textual descriptions and relationships between objects; Flickr30k, which is a smaller dataset containing detailed textual descriptions for 30,000 images, web data was also used to complement the public datasets).

BLIP pre-trained with the set of images described above showed better image results when compared to LEMON-base (which is a vision-language model designed to learn rich multimodal representations by jointly training on image and text data) [62], with better results in both the Consensus-based Image Description Evaluation (CIDEr) and Semantic Propositional Image Caption Evaluation (SPICE). The metric CIDEr measures the similarity between the generated captions and the reference captions (created by humans) based on the n-gram frequency, a metric used to analyze and count the occurrence of words or characters in a text. This metric (CIDEr) is more sensitive to the grammatical accuracy and fluidity of the captions. The SPICE, on the other hand, evaluates captions based on their semantic similarity, converting captions into scene graphs that represent objects, attributes, and relationships present in the image; this scoring is done by comparing the scene graphs of the generated caption vs the scene graphs of the reference captions. The results are as follows (CIDEr 111.3 vs 104.5 and SPICE of 15 vs 14.6).

Three approaches (TF-IDF [63], Bidirectional Encoder Representations from Transformers (BERT) [64], and Universal Sentence Encoder (USE) [65] were employed to convert text (the original annotations + BLIP) into feature vectors. We used these approaches to capture different dimensions of textural information and explore different statistical and semantic aspects, resulting in a more robust and comprehensive analysis. TF-IDF is a technique based on word frequency that evaluates the relevance of each term in a document (image caption in this case) about a corpus (set of all concatenated captions). This model is essential for capturing statistical patterns and identifying the most representative words in a global context. This methodology offers a statistical baseline to build on with more complete models. The major limitation of this model is that it does not take into account the semantic context of sentences, treating words as independent. BERT uses deep learning techniques to capture the contextual meaning of words, modeling both what comes before and after each term. This technique is ideal for capturing complex texts' nuances and deep semantic relationships. It allows us to identify similarities between sentences that use different words but have similar meanings. The USE provides a more direct and efficient approach. It was designed to generate dense fixed vectors for complex sentences, optimizing semantic similarity. However, it is less detailed than BERT for highly complex contexts. Following, we briefly explain how each model works.

- **Term Frequency-Inverse Document Frequency (TD-IDF)**: evaluates how many times a given word appears in a given sentence. This model generates vectors for each sentence with the number of unique words in the vocabulary. These vectors will have 0 in the words that are not in the specific sentence and the TF-IDF value in the words in the sentence. The model indicates the relevance of a word in the corpus; it gives a lower weight to prevalent words and a higher weight to rarer words. By doing so, we can reduce the influence of the frequent terms and highlight the more informative terms to distinguish sentences and documents. The model is beneficial for capturing the relative importance of words in relation to the corpus, providing a basis for comparing textual analyses and similarity comparisons.
- **Bidirectional Encoder Representations from Transformers (BERT)**: is a transformer-based model that generates contextual representations for words and sentences. The input text is tokenized into a sequence of tokens representing the individual words of the text. This sequence is processed by adding special tokens such as Classification token (CLS) at the beginning and Separator token (SEP) at the end. In addition, position and segment embeddings are added to encode information about the position and segment of each token. BERT comprises multiple layers of stacked Transformer encoders. The number of layers varies between BERTBASE (12 layers) and BERTLARGE (24 layers). Each layer of the encoder is composed of a self-attention mechanism, which allows the model to weigh the importance of different parts of the input sequence when generating the representation of each token. The output of the model depends on the specific task to be performed for classification tasks, such as predicting the next sentence. To output a single vector per sentence, what is returned is the final hidden state of the CLS token that is extracted as a representation of the sentence. In this case, the vector that is returned is BERTBASE, which returns a vector with a dimension of 768 values for each phrase. The model was pre-trained with the same datasets as the original article, which form the Book Corpus (a set of English books) and the English Wikipedia (processed and cleaned). BERT shows superior results on several datasets when compared to previous models such as OpensAI GPT [66] and ELMo [67].
- **The Universal Sentence Encoder (USE)**: the article proposes two encoders with different objectives, one aiming for high accuracy at the cost of greater model complexity and resource consumption and the other aiming for efficient inference with slightly reduced accuracy. The USE was pre-trained, as in the article, using the following datasets (Wikipedia, Books, Forum Questions and Answers, and the Web Universal corpus). The model used was the transformer-based encoding model, which controls sentence embeddings using the transformer architecture encoding sub-graph. This sub-graph uses attention to calculate word representations in a sentence that take into account the order and identity of all the other words. The context-aware word representations are converted into a fixed-length phrase encoding vector by calculating the sum of the elements

of the representations at each word position. The encoder takes as input a tokenized Penn Treebank (PTB) string in lowercase and generates a 512-dimensional vector as the embedding of the sentence.

3.3 Similarity Functions

In each of the recommendation systems developed, and in order to compare the feature vectors that describe each image, we used two similarity measures (euclidean distance and cosine similarity). Both metrics were used because they measure the similarity in a different way. Cosine similarity (see Figure 3.5) measures the angle between two vectors in space and gives more importance to the orientation of the vectors. It is also scale-independent, i.e., it ignores the length of the vectors and focuses more on angular similarity. It is usually used for normalized embeddings. In this case, we used a sklearn function that already has L2 normalization built in (see Equations 3.3, 3.4, 3.5 and 3.6). For a vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]$, the L_2-normalization is defined as:

$$\mathbf{x}_{\text{normalized}} = \frac{\mathbf{x}}{\|\mathbf{x}\|_2} \tag{3.3}$$

Where $\|\mathbf{x}\|_2$ is the L_2-norm, calculated as:

$$\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2} \tag{3.4}$$

Each normalized component $x_{i_{normalized}}$ is given by:

$$x_{i_{normalized}} = \frac{x_i}{\sqrt{\sum_{j=1}^{n} x_j^2}} \tag{3.5}$$

After L_2-normalization, the sum of the squares of the normalized components equals 1:

$$\sum_{i=1}^{n} (x_{i_{normalized}})^2 = 1 \tag{3.6}$$

The Euclidean distance (see Figure 3.5), in turn, measures the proximity in space between two vectors. This metric is good for emphasizing the magnitude of vectors, i.e., to know how far apart points are in absolute space. By using these two metrics, it is possible to make a more comprehensive analysis of vectors since we can explore both orientation and magnitude and see the impact of each.

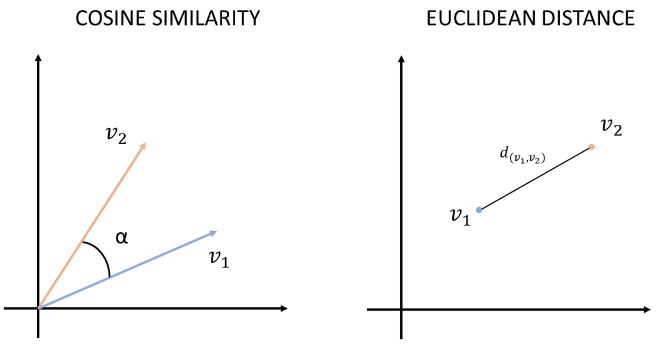


Figure 3.5: Comparison between cosine similarity and Euclidean distance. Cosine similarity (left) measures the angle α between vectors v_1 and v_2, evaluating their directional similarity. Euclidean distance (right) computes the straight-line distance $d(v_1, v_2)$ between the vectors, considering their absolute difference in magnitude and position.

3.4 Scores

The recommendation aims to rank the ten items most likely to be liked by the user, since it is a common standard in recommendation systems. The recommendation is based on the items in the trainset (further explained in Section 3.6), which will be the items that make up our user profile. The number of items in the user profile (from 1 to 15 because each user has a maximum of 15 images on their profile, the total number of images they have viewed) was tested to find the ideal number for each architecture, and three different scores were also applied to calculate the relevance of the recommended items.

The first score used the average of the ratings given by the user to the items most similar in their profile to the item we wanted to recommend. This score (see Equation 3.7) gives greater importance to historical relevance since it considers ratings as a relevance metric, so this metric suggests increasing the weight of the user's history since it gives greater importance to the ratings given by the user.

$$\text{Simple Average} = \frac{\sum_{i=1}^{n} r_i}{n} \tag{3.7}$$

- n: Number of the most similar items from the user's profile.
- r_i: Rating assigned by the user to item i in their profile.

Our second score uses a weighted average of the similarity value with the rating given by the user for the n items in their user profile (see Equation 3.8). This score balanced the user's profile with the similarity so that the similarity between the items has a greater weight than the first.

$$\text{Weighted Average} = \frac{\sum_{i=1}^{n} s_i \cdot r_i}{\sum_{i=1}^{n} s_i} \tag{3.8}$$

- n: Number of the most similar items from the user's profile.
- r_i: Rating assigned by the user to item i in their profile.
- s_i: Similarity between item i in the user's profile and the item being recommended.

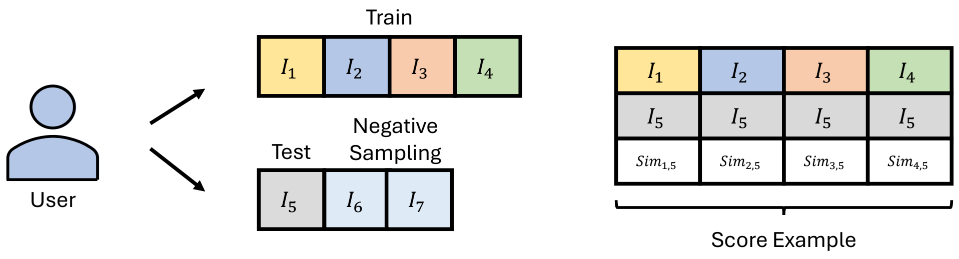
Our third score uses a weighted average similar to the second score, but squares the similarity between vectors. This score allows the weight to be increased for items that are more similar to the user's profile (see Equation 3.9).

$$\text{Squared Weighted Average} = \frac{\sum_{i=1}^{n} s_i^2 \cdot r_i}{\sum_{i=1}^{n} s_i^2} \tag{3.9}$$

- n: Number of the most similar items from the user's profile.
- r_i: Rating assigned by the user to item i in their profile.
- s_i: Similarity between item i in the user's profile and the recommended item.

When two images in the recommendation list have the same score, the similarity between the image to be recommended and the images in the user's profile was used to break the tie. By implementing different scores, complementary approaches to recommendation can be explored, and the best score can be assessed.

The figure 3.6 shows the workflow of a content-based recommendation system. The items are divided into two sets: the training set (I_1, I_2, I_3, I_4), used to build the user profile, and the test set (I_5, I_6, I_7), with items to be recommended. The image is a simplified version of the data split where 7 items are used as a symbolic representation. In reality, we have 12 images in the test set and 23 in the test, of which 3 are evaluated, and the remaining 20 belong to negative sampling. The relevance of each item in the test set is calculated based on its similarity to the items in the training set, weighted by the ratings given by the user. The items in the test set are then ordered by the calculated scores, resulting in a prioritized recommendation list (I_5, I_7, I_6).



$$\text{Score } I_5 = \frac{(Sim_{1,5} \times ratingI_1)+(Sim_{1,5} \times ratingI_1)+(Sim_{1,5} \times ratingI_1)+(Sim_{1,5} \times ratingI_1)}{4}$$

$$\text{Recommendation } List = Score\ I_5 > Score\ I_7 > Score\ I_6$$

$$\text{Recommendation } List = I_5,\ \ I_7,\ \ I_6$$

Figure 3.6: Recommendation workflow. The figure illustrates the process of generating user recommendations. The items are divided into a train set (I_1, I_2, I_3, I_4) to construct the user profile and a test set (I_5, I_6, I_7) for a recommendation. The relevance of test items is calculated based on their similarity to the train items, weighted by user-provided ratings. The test items are then ranked by their scores, producing a prioritized recommendation list (I_5, I_7, I_6).

3.5 Baseline Recommendation System

The Random recommendation system is used as a baseline in our work. In our implementation, the system selects 10 images from a sample of 23, including the test set images and 20 additional items the user has not seen. In this implementation, all images are equally likely to be included in any position in the list of recommendations.

3.6 Evaluation

Our evaluation was done using an offline approach because it allows us to simulate user interactions with the recommendation system using pre-existing data without requiring real-time feedback from users. A split was made to evaluate the system, dividing each user's items by 80% for the training set or user profile and the remaining 20% for the test set items. This split resulted in 12 trainset items and 3 test set items per user. The developed recommendation systems were evaluated using precision, recall, f1-score, and nDCG. These metrics were chosen because they evaluate different aspects of the recommendation system. The evaluations were made for different Ks (Ks, i.e., the number of items we are recommending). The items recommended for evaluating the system are the items in the test set plus 20 random items the user has never seen. For these 20 items, we assigned a rating of 0 since the user had never seen the images;

thus, we assumed that they did not like them (negative sampling). The split was made 5 times to have a fairer evaluation since the items in the trainset may not be at all similar to the items in the test and vice-versa. This way, we ensure we have an evaluation that depends less on the division's randomness to test the system without being so biased towards items that may be very close or very distant in terms of similarity.

The values of these metrics were compared for the dataset with 553 images for pixels, emotions, metadata, and the random model. The approaches (pixel-based, emotion-based, metadata-based, and random) using 553 images and 2885 images were also compared for pixels because there were no limitations due to the features. So we can analyze the generalization (that may occur with a bigger dataset).

3.7 Summary

This chapter presents the methodology for developing the content-based image recommendation systems proposed in this master's thesis. We presented the dataset we used to train and test our recommendation systems, followed by the three approaches we used to extract features (pixels, emotions, and textual annotations). Where CNN (VGG, ResNet, InceptionV3, Xcepiton). We used to extract the embeddings of the images; for the emotions, an average of the ratings given by the users in the survey was carried out to create the vectors representing the emotions for each item, and, finally, for the metadata, the annotations from Pexels were used. For the images that did not have them, a BLIP method was used, TF-IDF, BERT, and USE were used to capture the information of the feature vectors. Two similarity metrics were then used: cosine similarity and Euclidean distance. We used three types of scores to compute a simple average, a Weighted Average, and a Squared Weighted Average. Precision, recall, f1-score, and nDCG were used to evaluate the systems, and a split of 80% for training and 20% for testing was made. These evaluations were repeated 5 times in order to have greater confidence in the results of the metrics.

Chapter 4

Results and Discussion

This chapter presents the results of the three content-based recommendation systems (pixels, emotions, and metadata-based) developed. We also present the results of a random recommendation system as a baseline for comparison. We will then discuss the results, focusing on analyzing the impact of different factors such as the type of score used, the number of items in the user profile to search, the type of similarity, and the feature extraction model.

In our analysis, we focused on precision because, as seen in the Background and Related Work, it indicates that the system avoids recommending irrelevant items. This metric is favored over recall since recall is more sensitive to quantity because it measures the number of relevant items in the set of relevant items available, i.e., the more items that are recommended, the higher this value naturally becomes. Precision is also a good metric because it correlates with metrics that also value the relevance of the items at the top of the ranking, such as nDCG. In particular, we emphasize precision at $k = 1$, as our goal is to ensure that the very first recommended item is as relevant as possible. Other system configurations, such as variations in parameters, were also evaluated and are documented in Appendix A.

4.1 Emotion-based Recommendation System

In the Figures 4.1, 4.2, 4.3 and 4.4, we can see the results in different ks for each metric. It is important to note that the graphs were generated using the best-performing parameters for each emotion type, which can be seen in Table 4.1 in terms of precision at $k = 1$.

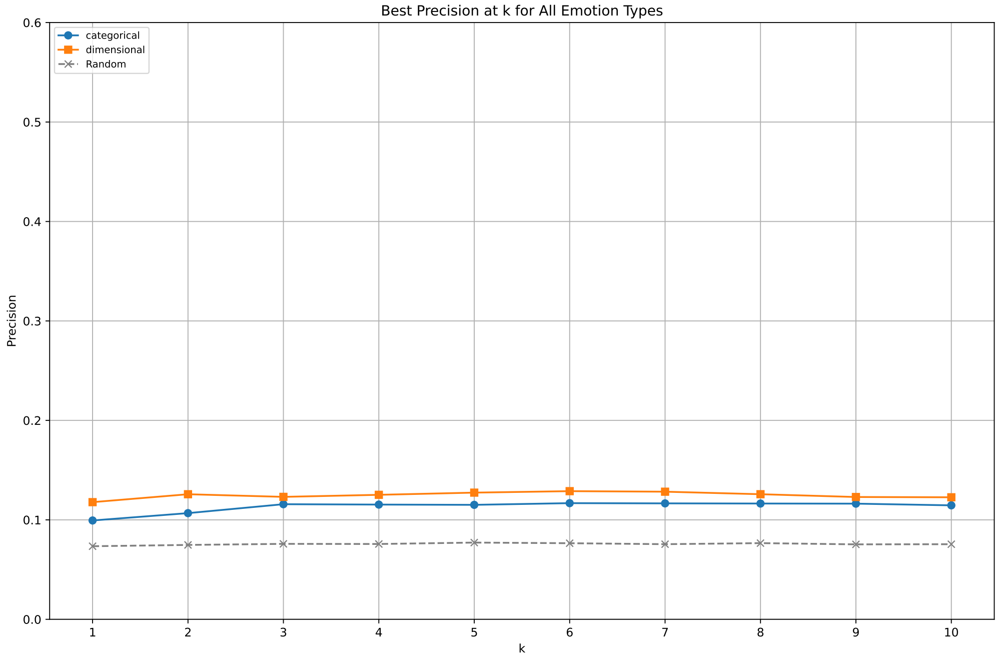


Figure 4.1: Results using precision@k for the emotion-based recommendation system. (best seen in color).

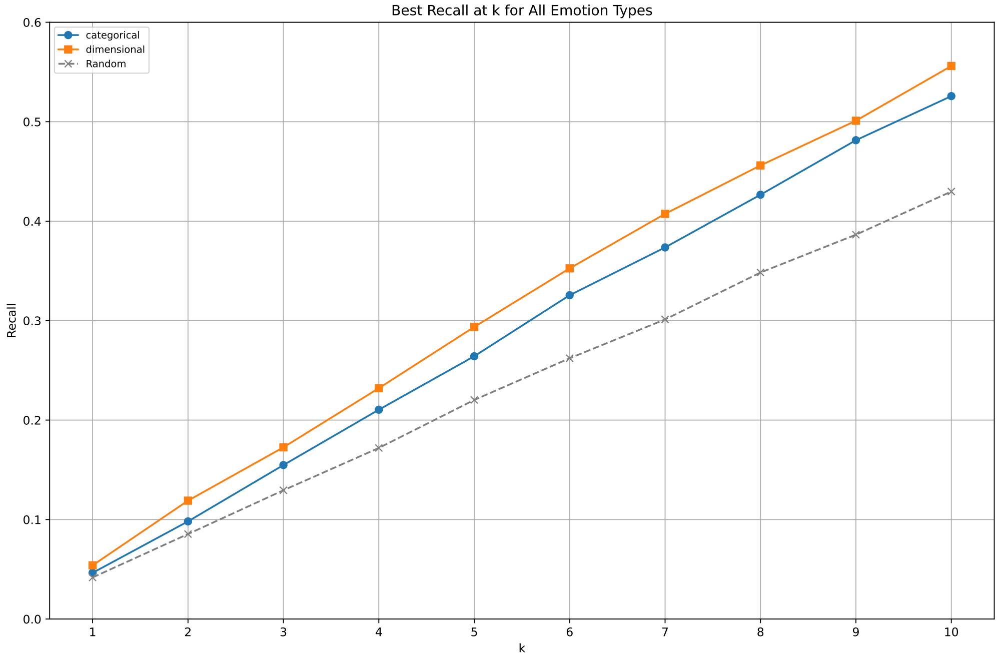


Figure 4.2: Results using recall@k for the emotion-based recommendation system. (best seen in color).

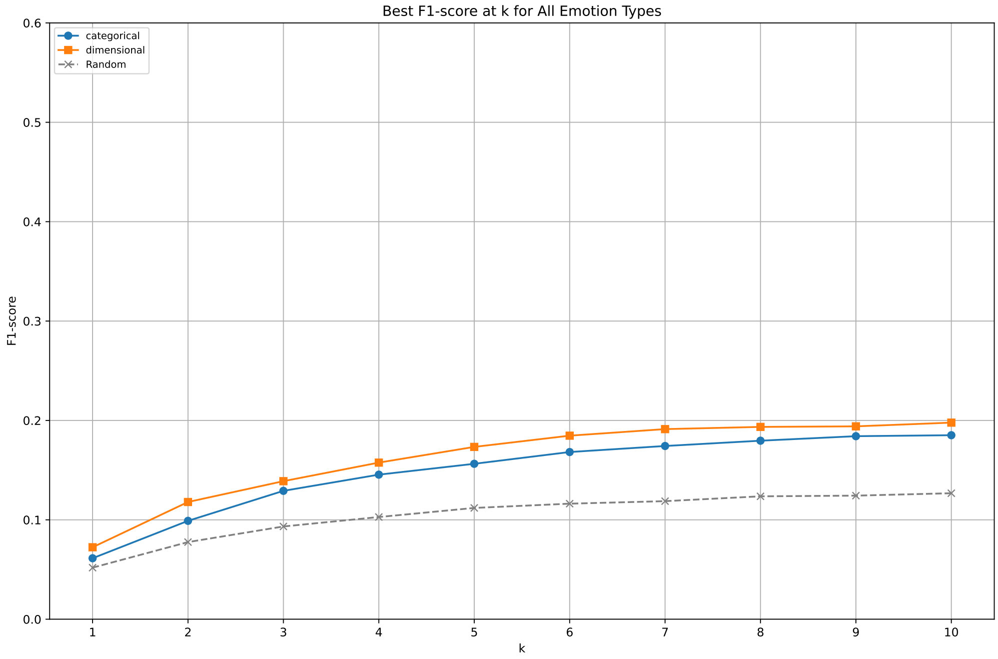


Figure 4.3: Results using f1-score@k for the emotion-based recommendation system. (best seen in color).

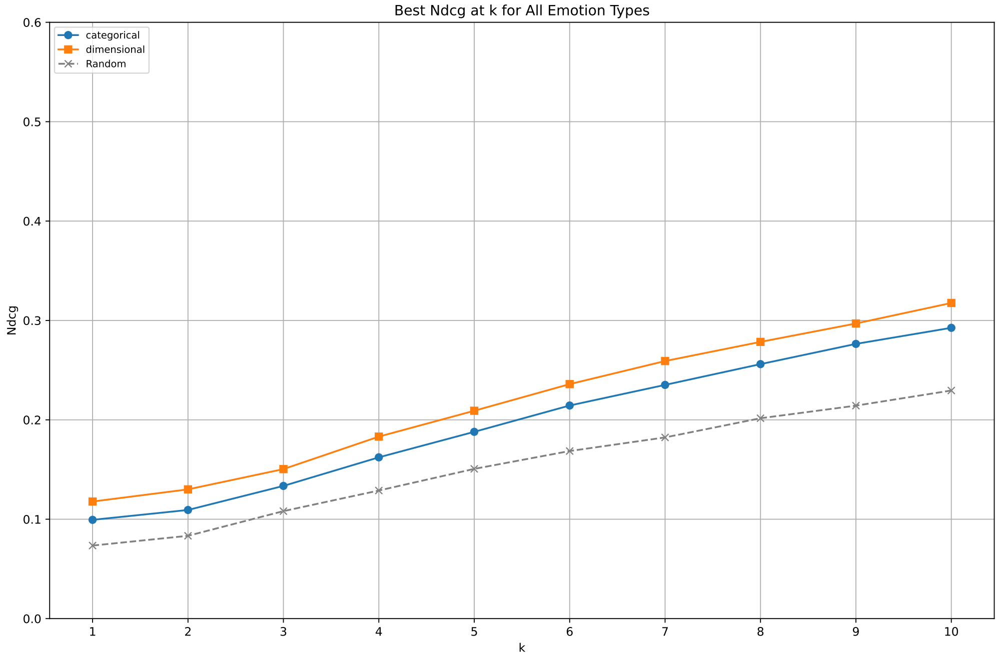
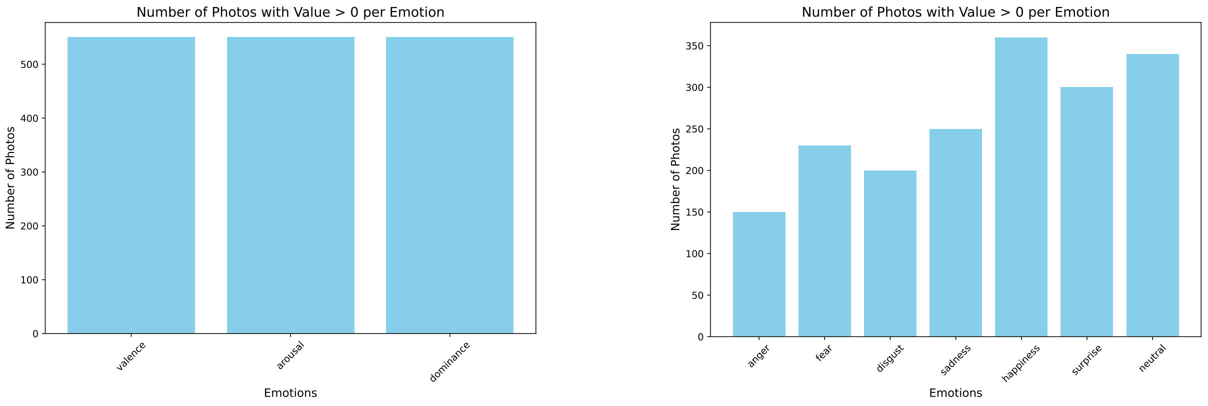


Figure 4.4: Results using nDCG@k for the emotion-based recommendation system. (best seen in color).

Table 4.1: Best configurations for each emotion type based on n (i.e., that is the number of items to fetch from the train set/user profile), similarity function, and score (i.e., the method used to rank the recommended items).

Emotion Type	n	Similarity Function	Scores
Dimensional	8	Euclidean	Weighted Average
Categorical	8	Euclidean	Simple Average

Thus, the graphs show that the dimensional model shows slightly higher results in all the metrics, particularly in nDCG. This increase may be related to the fact that all the images were classified in the three dimensions, which was not the case with the categorical emotions, as only one emotion needed to be provided to continue the survey. So, not all categorical emotions appear the same number of times. This data can be seen in Figure 4.5.



(a) Dimensional Emotions

(b) Categorical Emotions

Figure 4.5: Comparison between dimensional and categorical emotion distributions, focusing on the number of photos higher than 0, which are the images that were seen by a user.

One of the problems with emotions is when the rating of the user for whom the recommendation is being made is removed, so there is no bias in the recommendation. This way, some images are left without any rating because the only emotional rating they had was that from the user. In these cases, zero was considered. This case happens with the categorical emotions, and since a vector of zeros is not as similar as other vectors, scores 2 and 3 have a minor value. Of the 553 images that were evaluated, 112 were only evaluated once, which is one of the limitations of this methodology.

4.2 Pixel-based Recommendation System

Following, we will analyze the performance of each architecture used to extract the visual information of each image that was further used in the recommendation system developed. Table 4.2 presents the best n, similarity, and score combinations for each architecture.

As we can see in Figures 4.6, 4.7, 4.8, 4.9, all the architectures performed better than the random baseline, marked in gray with the discontinued line. It is important to note that the graphs were generated using the parameters that achieved the best precision at $k = 1$ for each architecture individually. The parameters are showed in Table 4.2. It also shows that the best architecture across all metrics is Xception, which is probably due to the structure of depthwise separable convolutions; the model is able to capture more complex patterns, as well as more details and subtle differences in textures, shapes, and colors. The worst-performing architecture is InceptionV3 this is probably due to the architecture capturing details and local patterns because of its smaller and more specific convolutions. This architecture shows good classification results but does not seem as good for a recommendation since it can not capture the global relationships between items.

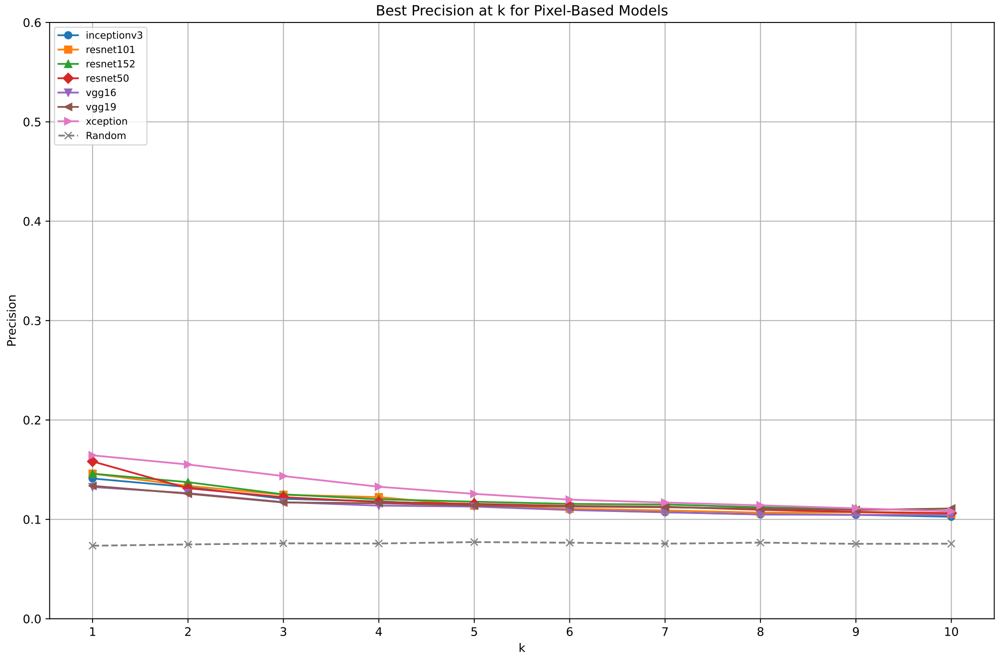


Figure 4.6: Results using precision@k for the pixel-based recommendation system. (best seen in color).

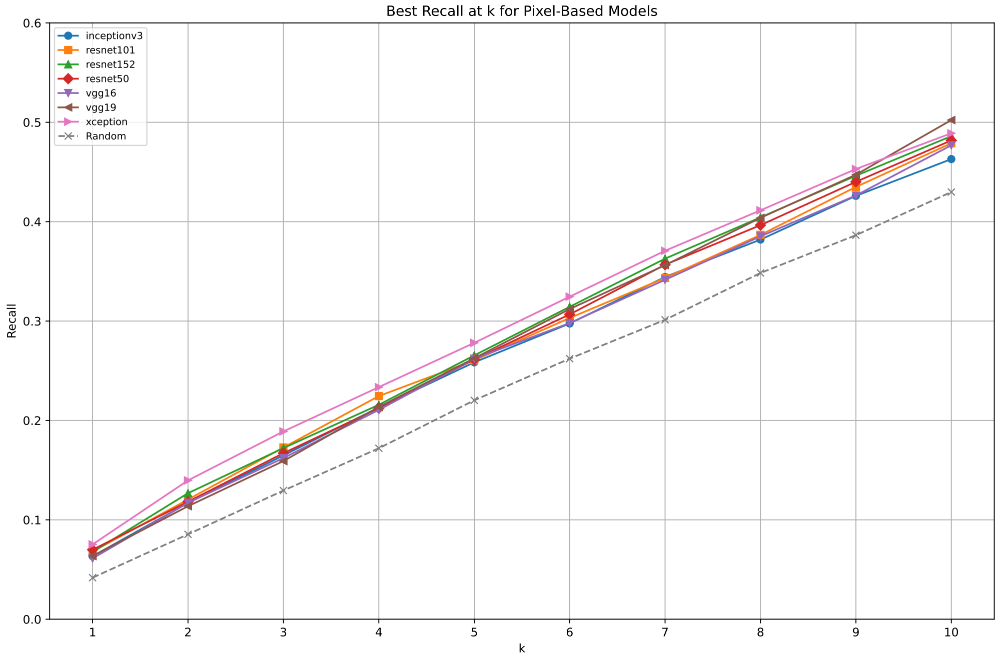


Figure 4.7: Results using recall@k for the pixel-based recommendation system. (best seen in color).

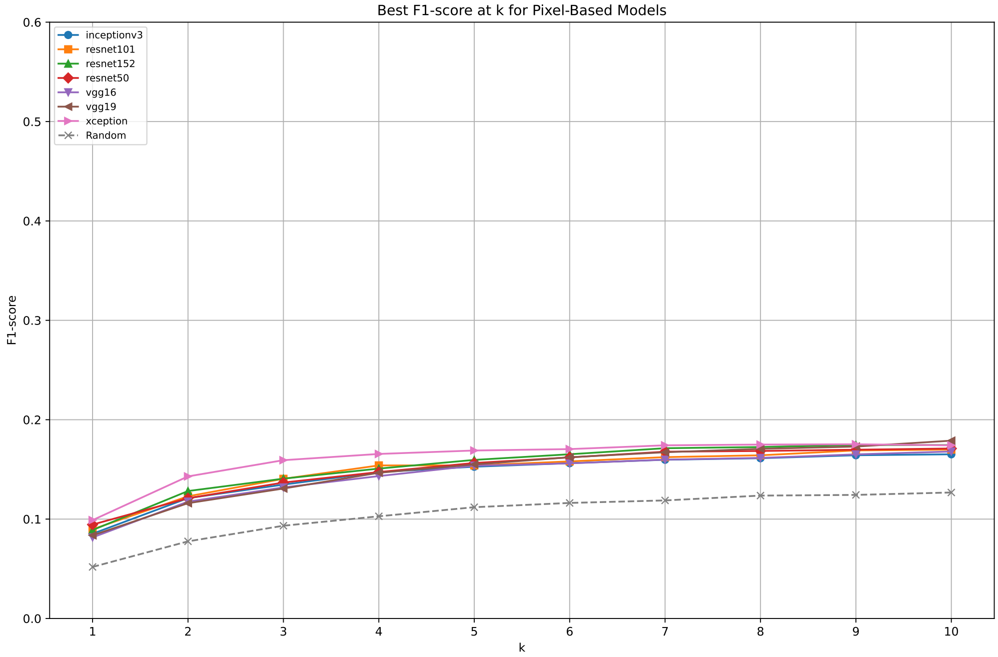


Figure 4.8: Results using f1-score@k for the pixel-based recommendation system. (best seen in color).

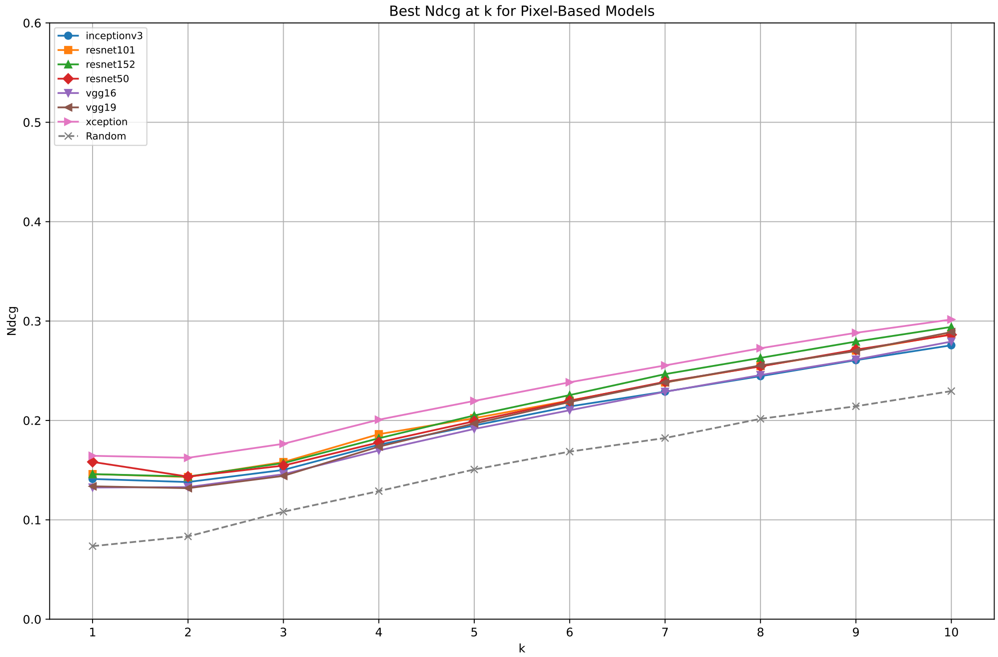


Figure 4.9: Results using nDCG@k for the pixel-based recommendation system. (best seen in color).

Table 4.2: Best configurations for each architecture based on n (i.e., the number of items to fetch from the train set/user profile), similarity function, and score (i.e., the method used to rank the recommended items).

Architecture	n	Similarity Function	Score
Xception	7	Euclidean	Squared Weighted Average
ResNet50	11	Euclidean	Squared Weighted Average
InceptionV3	5	Euclidean	Weighted Average
ResNet152	12	Euclidean	Squared Weighted Average
ResNet101	1	Euclidean	Simple Average
VGG19	4	Cosine	Simple Average
VGG16	2	Cosine	Simple Average

Moreover, in Table 4.2, we can see that for most architectures, a relatively higher number of items (n) allows the system to capture a more comprehensive representation of the user's preferences. Consequently, given their depth, the Xception and ResNet architectures are particularly well-suited for capturing complex patterns in the data. The larger the n, the less influence there is from items (outliers) outside the user's profile pattern. Euclidean distance captures better due to the geometric relationships between feature vectors, i.e., it allows the capture of the similarity of local and global patterns between vectors.

The Euclidean distance is also relevant since it measures the magnitude of the vectors. Regarding scores, the best results were achieved using two or three, which is aligned with our expectation since they introduce similarity as a weight. This gives more relevance to the items in the user's profile that are closest to the recommended item, thus ensuring that the items that most influence the score are those that are most closely related to the current item.

4.3 Pixel-based Recommendation System: Full Dataset

In the case of the pixels, since there was no obstacle to using the entire dataset due to the lack of data, a generalization and scalability analysis was carried out to see how the model behaves when analyzed for all the images. These results will only be compared with the results of the pixel recommendation system in the subset pictures evaluated to predict the scalability that the other models could achieve. The best combinations by precision for $k = 1$ for the recommendation system in each case will then be presented in Tables 4.3,4.4, 4.5, and 4.6.

Table 4.3: Results for Subset and Full Dataset for Precision at $k = 1, 5, 10$.

Dataset Type	**Precision@1**	**Precision@5**	**Precision@10**
Subset	**0.16442**	**0.12074**	**0.10798**
Full Dataset	0.14476	0.11310	0.10136

Table 4.4: Results for Subset and Full Dataset for Recall at $k = 1, 5, 10$.

Dataset Type	**Recall@1**	**Recall@5**	**Recall@10**
Subset	**0.05114**	**0.27446**	**0.48896**
Full Dataset	0.06566	0.25870	0.45972

Table 4.5: Results for Subset and Full Dataset for F1-Score at $k = 1, 5, 10$.

Dataset Type	**F1-Score@1**	**F1-Score@5**	**F1-Score@10**
Subset	**0.09796**	**0.16376**	**0.17396**
Full Dataset	0.08712	0.15318	0.16306

Table 4.6: Results for Subset and Full Dataset for nDCG at $k = 1, 5, 10$.

Dataset Type	nDCG@1	nDCG@5	nDCG@10
Subset	**0.16442**	**0.21502**	**0.30158**
Full Dataset	0.14476	0.20080	0.28198

The subset shows better results in all metrics and demonstrates that a smaller data set improves the prioritization of relevant items. The total dataset shows a drop in all metrics due to the greater diversity of items. However, the drop is not very sharp, showing the scope for scalability of the recommendation system.

It should be noted that Xception leads the way when it comes to the recommendation system applied to the subset of data, and it also leads when $k = 5$. The ResNet, especially the ResNet152 architecture, showed promising results for the entire dataset.

4.4 Metadata-based Recommendation System

In Table 4.7, we present the results achieved for the different methods used: BERT, TF-IDF and USE. In the figures 4.10, 4.11, 4.12 and 4.13, we can see the results by metric. It is important to note that the graphs were generated using the parameters of the best-performing architecture in terms of precision at $k = 1$ applied to all of the other architectures. The parameters are $n = 10$, Euclidean distance, and $score = 3$.

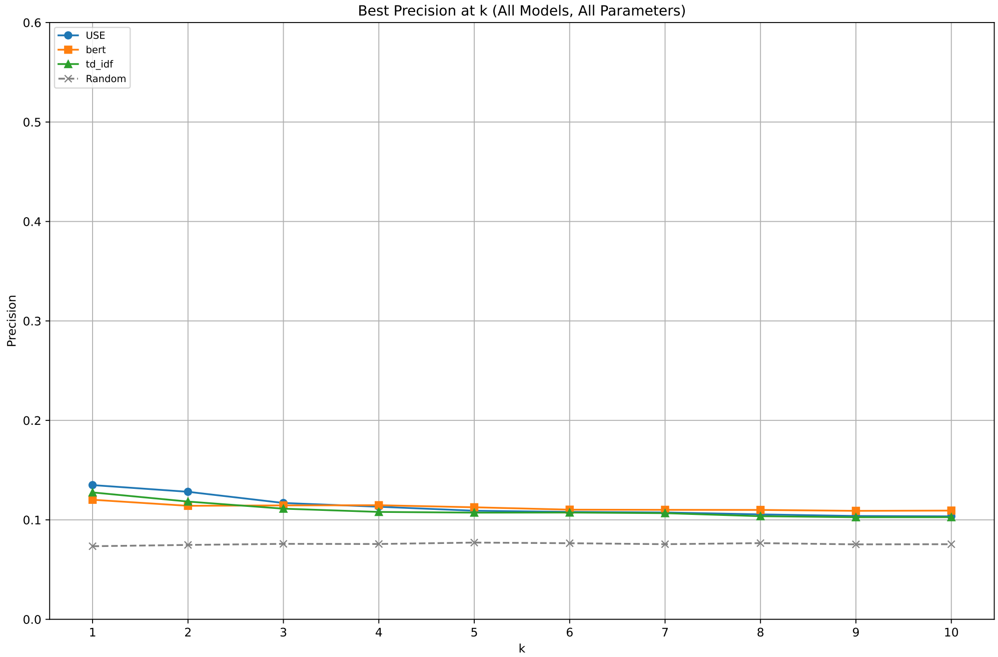


Figure 4.10: Results using precision@k for the metadata-based recommendation system. (best seen in color).

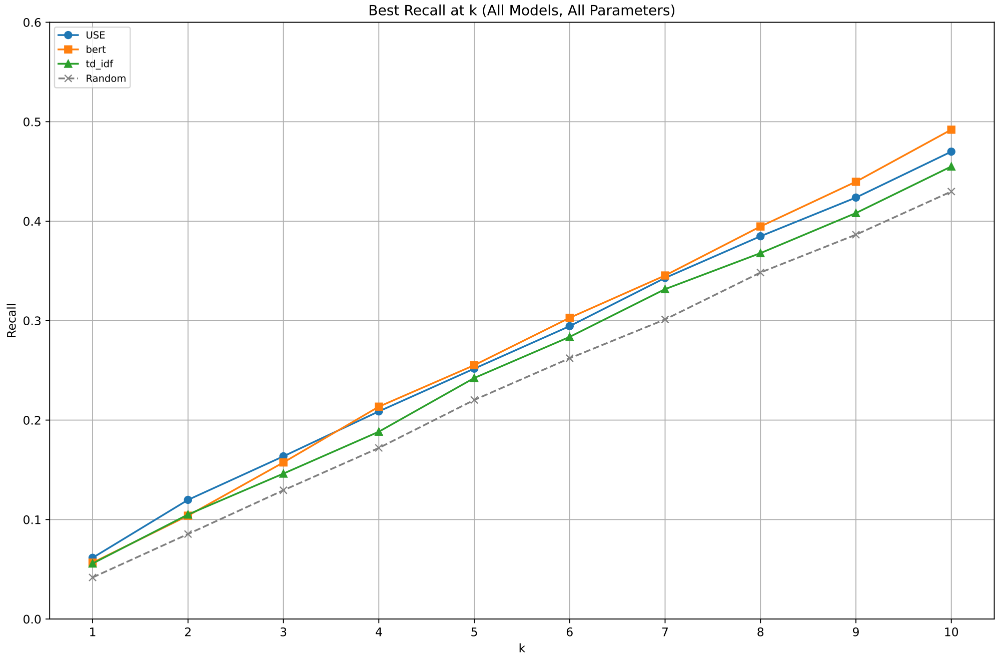


Figure 4.11: Results using recall@k for the metadata-based recommendation system. (best seen in color).

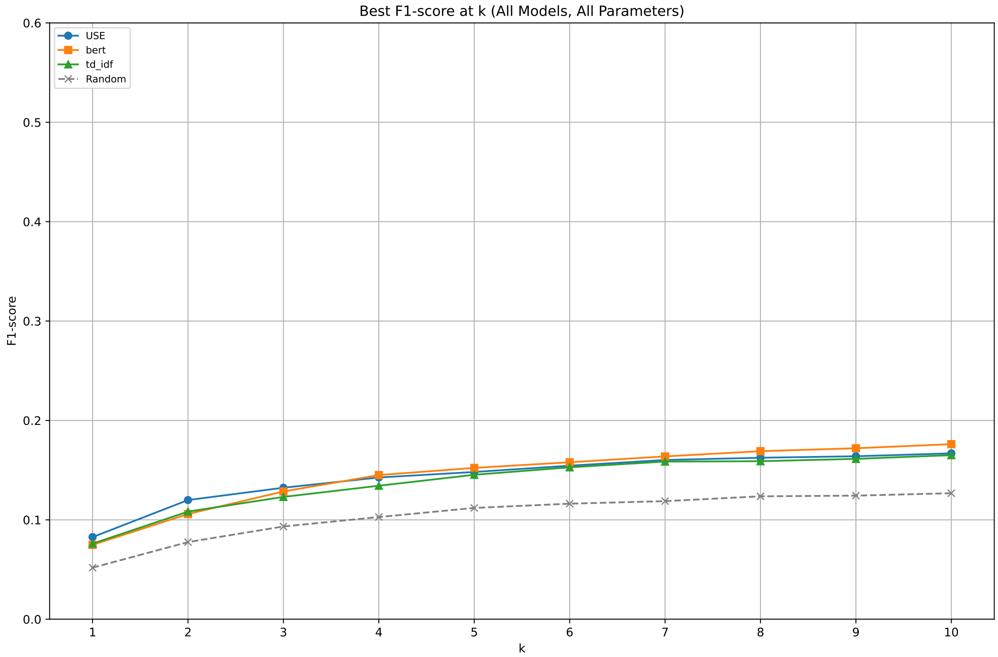


Figure 4.12: Results using f1-score@k for the metadata-based recommendation system. (best seen in color).

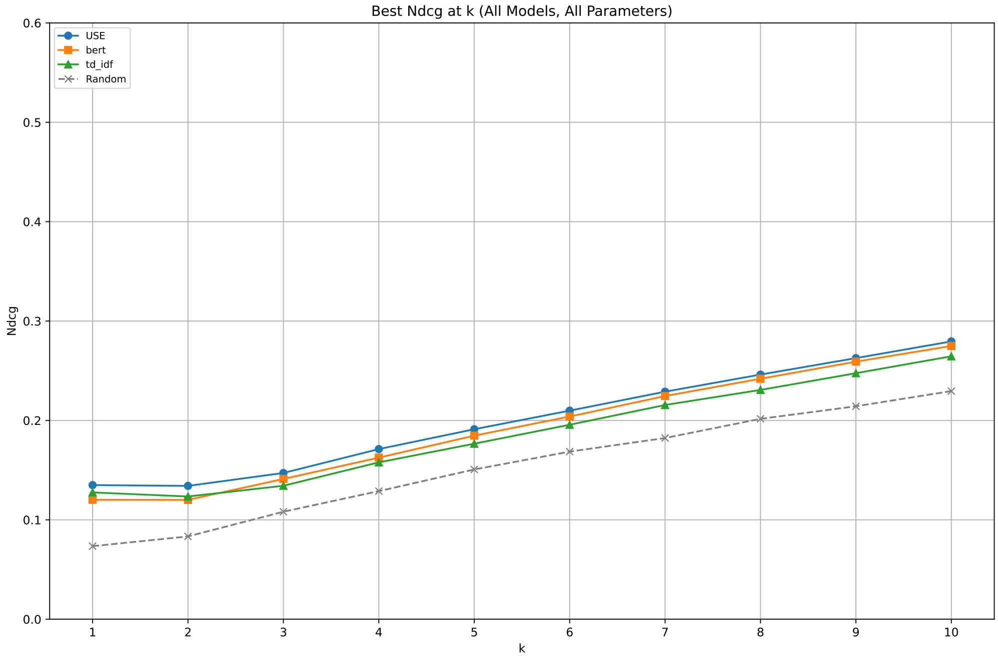


Figure 4.13: Results using nDCG@k for the metadata-based recommendation system. (best seen in color).

Table 4.7: Best configurations for each annotation model architecture based on n (i.e., that is the number of items to fetch from the train set/user profile), similarity function, and score (i.e., the method used to rank the recommended items.

Architecture	**n**	**Similarity Functions**	**Scores**
USE	12	Euclidean	Squared Weighted Average
BERT	2	Euclidean	Simple Average
TD-IDF	5	Cosine	Weighted Average

All models behave better than the random model in all metrics and have an equivalent performance relative to each other. For lower ks, we can detect that BERT is superior; however, as k increases, the USE model had a better performance. The best model overall is BERT. Nevertheless, the quality of each model is relatively equivalent. Concerning precision, we can say that as k increases, the list of recommendations reduces the proportion of relevant items, which is to be expected as the list contains more items. We can see that for $k \geq 4$, the BERT model shows a performance similar to the USE. Regarding recall, the model with the best capacity for $k \geq 4$ is BERT. The f1-score is a metric that balances precision and recall, indicating that from $k = 4$ BERT and USE have a similar performance. In the nDCG, we find that through every k USE has better results.

The architecture USE performs best overall because its transformer-based architecture is very effective at this task. TF-IDF does not perform better than the other two models since this model is a less complex solution, but it is still efficient due to its proximity to the other two models. BERT has a good performance overall because it prioritizes relevant items in the top positions of the list due to the model's specialization in capturing semantic similarities.

4.5 Random Recommendation System

The results of the random recommendation system are shown in Figures 4.14, 4.15, 4.16 and 4.17. We observe that the maximum Precision of the random model is 0.077 ± 0.008 at $k = 5$. For the other metrics, the highest values are obtained at $k = 10$, where Recall $= 0.430 \pm 0.029$, F1-Score $= 0.127 \pm 0.008$, and nDCG $= 0.223 \pm 0.019$.

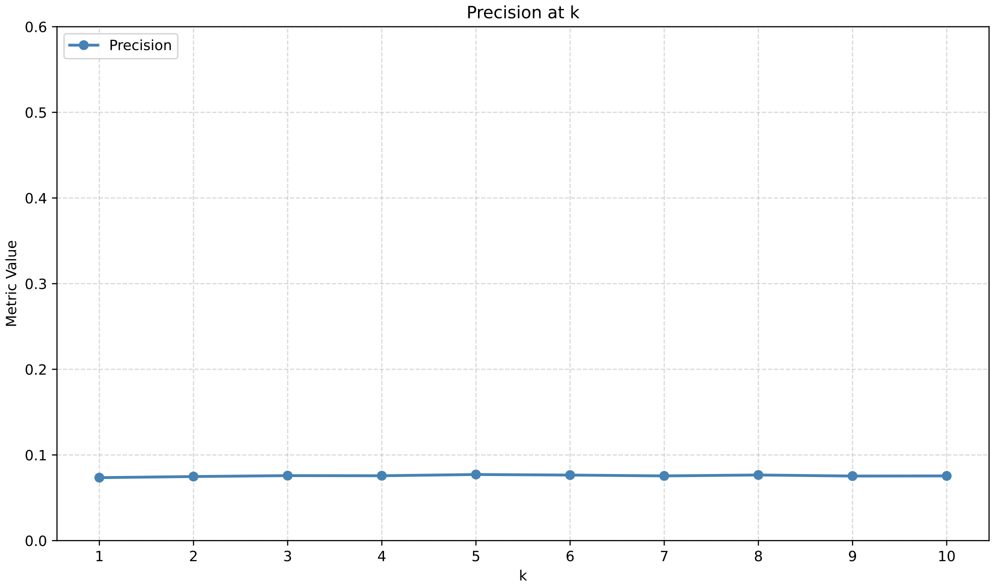


Figure 4.14: Results using precision@k for the random recommendation system.

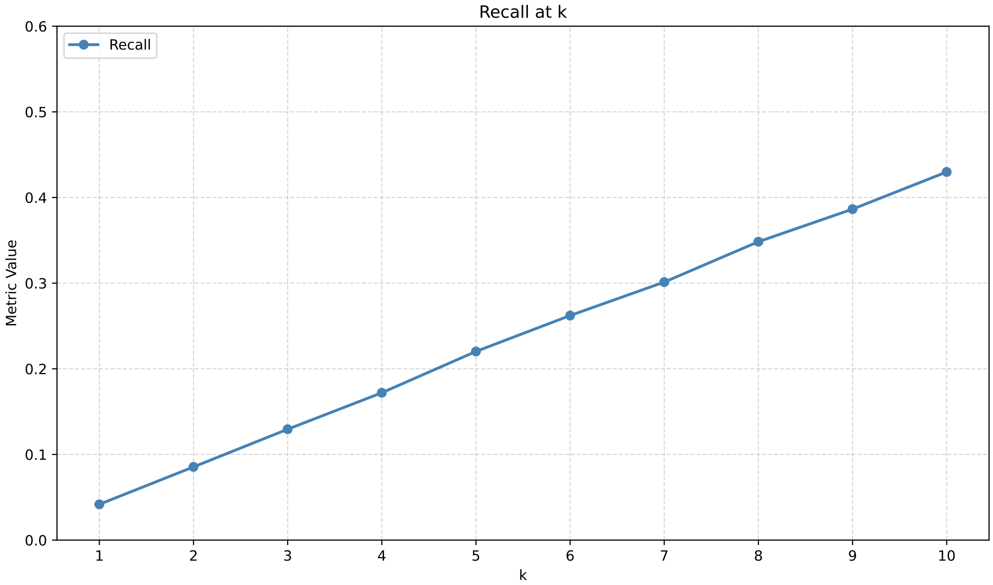


Figure 4.15: Results using recall@k for the random recommendation system.

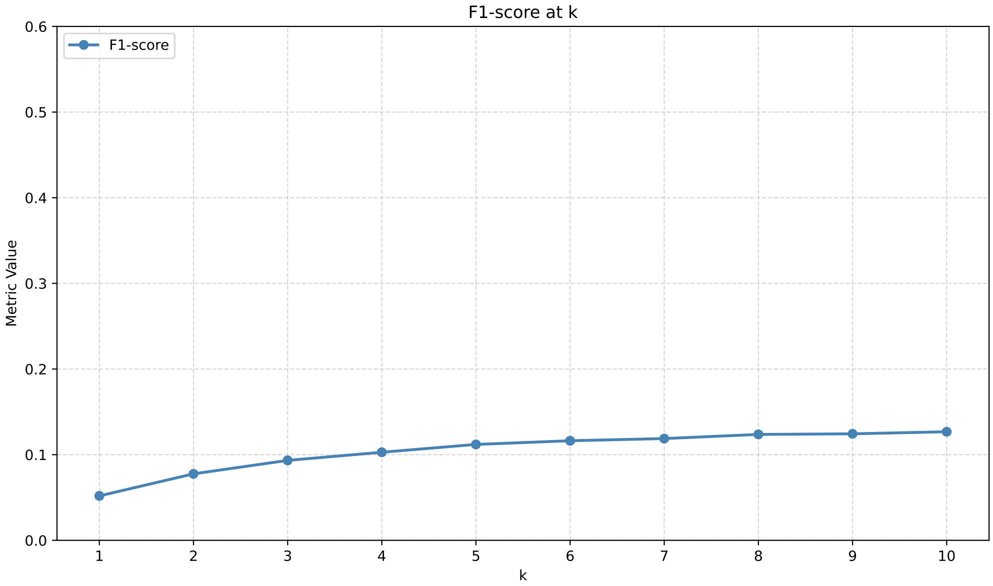


Figure 4.16: Results using f1-score@k for the random recommendation system.

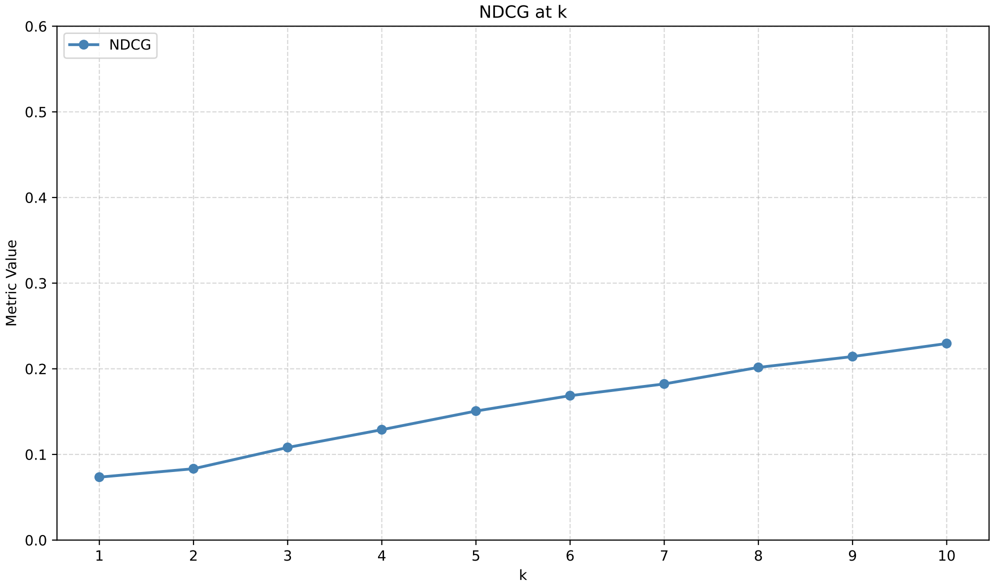


Figure 4.17: Results using nDCG@k for the random recommendation system.

4.6 Comparison of Top-5 Models Across Different Feature Types

This section focuses on the comparative analysis of the three model types. Next, we analyzed the performance across all metrics — precision, recall, F1-score, and nDCG — for $k = 1$, $k = 5$, and $k = 10$. These values of k represent distinct recommendation scenarios: a single top suggestion ($k = 1$), a balanced shortlist ($k = 5$), and a broader set of recommendations ($k = 10$). As can be seen in the Tables 4.8, 4.9, 4.10, and 4.11.

Table 4.8: Precision results for Emotion, Pixel, and Metadata models at $k = 1, 5, 10$.

Model	**Precision@1**	**Precision@5**	**Precision@10**
Emotion	0.11780	**0.12244**	**0.12198**
Pixel	**0.16442**	0.12074	0.10798
Metadata	0.09450	0.10356	0.10330

Table 4.9: Recall results for Emotion, Pixel, and Metadata models at $k = 1, 5, 10$.

Model	**Recall@1**	**Recall@5**	**Recall@10**
Emotion	0.05398	**0.28324**	**0.55480**
Pixel	**0.07200**	0.27446	0.48896
Metadata	0.04010	0.23354	0.46626

Table 4.10: F1-Score results for Emotion, Pixel, and Metadata models at $k = 1, 5, 10$.

Model	F1-Score@1	F1-Score@5	F1-Score@10
Emotion	0.07240	**0.16680**	**0.19670**
Pixel	**0.09796**	0.16376	0.17396
Metadata	0.05520	0.14008	0.16652

Table 4.11: nDCG results for Emotion, Pixel, and Metadata models at $k = 1, 5, 10$.

Model	nDCG@1	nDCG@5	nDCG@10
Emotion	0.11780	0.20820	**0.31760**
Pixel	**0.16442**	**0.21502**	0.30158
Metadata	0.09450	0.16982	0.26330

Each model performs differently depending on the value of k. When we only want to recommend one image, we analyze the different metrics for $k = 1$. By analyzing the different metrics, we can see that the pixel-based approach is superior. This can be attributed to the use of the Xception architecture, which is good at capturing small details in images through deptwise separable convolutions. The emotion-based and metadata-based systems follow respectively. Metadata probably has the worst results because the annotations can't capture the similarity of the images as well.

For a list of 5 recommended items, the pixel-based approach achieved better results for the nDCG metric, but the emotion approach surpasses the other approaches in all the other metrics. The pixel-based system can be favored in the nDCG because although it doesn't get the total number of relevant items right, it does get the top positions right, which is valued in this metric. For precision, recall, and F1-score, we can see the effectiveness of emotions. Emotions seem to better capture the user's explicit preferences, increasing the likelihood of capturing a relevant item when it's not at the top of the list. The metadata-based recommender system is close to the values of the other two systems for all metrics, but shows lower recommendation effectiveness.

As k increases to 10, where the goal is to recommend a list of 10 images, the emotion-based recommendation system shows better effectiveness. It achieves higher values than the other systems in all metrics. So we can suggest that emotions are better when we have a larger list to recommend, probably because they capture users' emotional preferences, allowing us to generate broader and more relevant recommendations. Pixels follow behind emotions, albeit slightly. And annotations are less efficient when we want to recommend a long list of items, in this case, images.

Figures 4.18 and 4.19 present some examples of the returned images for different values of k to provide additional evidence of the performance of the recommendation models.

In Figure 4.18, the emotion-based recommendation is shown. At the top of the figure, we have the images that belong to the user's profile, images that the user has already viewed. Images with a green border are items that the user liked, while those with a red border are images that the user disliked. Each image is also labeled with the dominant emotion provided by the user. At the bottom of the figure, the images with the blue background are the recommendations by the emotion-based system. The test images to be evaluated are also displayed to the left. It is possible to notice that one of the suggested images is one that the user liked previously (the image with the green background in the blue rectangle). All images are tagged with the average emotion computed across all users who have rated them. For the test images, the emotion label also includes the specific emotion given by the target user, except in those cases where the user's emotion matches the mean (e.g., "Mean Emotion (all users) / Emotion given by the target user"). It is noteworthy that the upper test image is labeled as "– / Sadness" because only this user provided an emotional classification for this particular image. It is also worth noting that the top recommended images mainly had the emotion of 'Happiness' because the user liked this emotion.

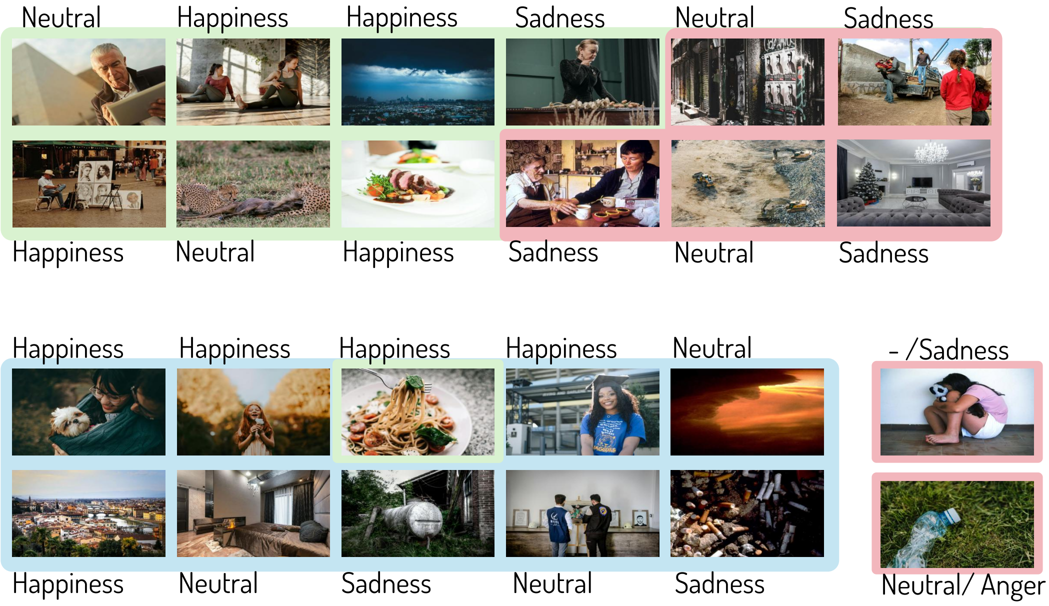


Figure 4.18: Emotion-based recommendations at the top we have the images from the user profile and at the bottom the recommended images, including the three images from the test.

On top of Figure 4.19 are the user's profile images. Green color images correspond to the user's likes, and red color images denote the user dislikes. The blue background images at the bottom are the recommendations from the pixel-based approach. There are three test images, with the same color scheme: green for images liked and red for images disliked. Both the user profile and the recommended list have images of food in them, suggesting a visual trend in the user's taste. Also, the image color palette of the liked images appears visually coherent with the top few recommended items.

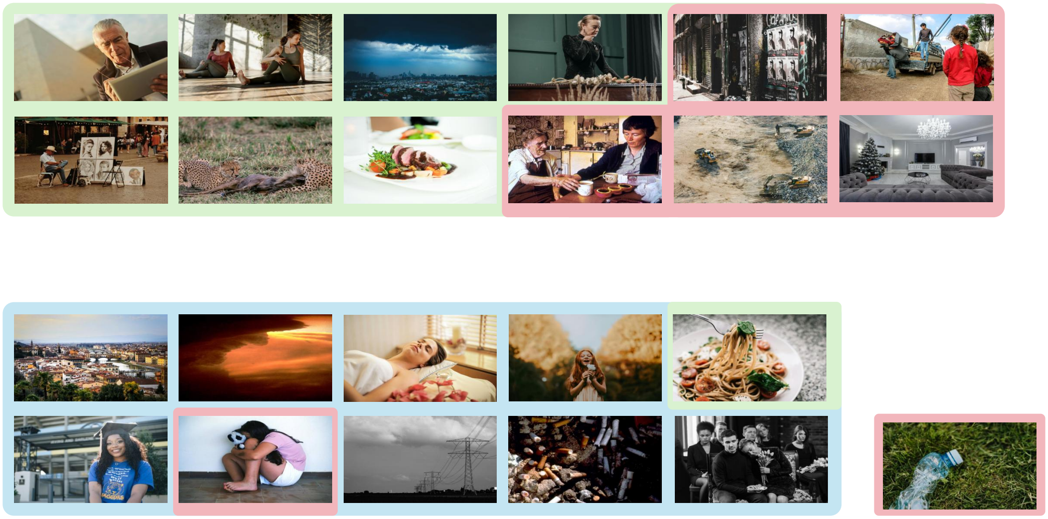


Figure 4.19: Pixel-based recommendations at the top we have the images from the user profile and at the bottom the recommended images, including the three images from the test.

These results highlight the strengths of each model: pixel data is highly effective for identifying the single best match; emotional data excels in broader, more exploratory recommendation scenarios; and metadata, while useful, is limited by its textual abstraction.

4.7 Summary

This chapter presented and compared four recommendation systems: pixel-based, emotion-based, metadata-based, and a random baseline. The evaluation employed four metrics — precision, recall, F1-score, and nDCG — at $k = 1$, $k = 5$, and $k = 10$, capturing different recommendation scenarios ranging from a single top item to broader ranked lists.

Among the three content-based systems, the pixel-based model consistently achieved the highest precision at $k = 1$, demonstrating strong performance in identifying the most relevant image. This outcome is primarily attributed to the Xception architecture.

The emotion-based system, particularly using dimensional emotion vectors, showed increasing performance as k increased. It achieved the best results at $k = 10$ across all metrics — recall, F1-score, and nDCG — suggesting that emotional representations are particularly well-suited for broader recommendation scenarios.

In contrast, the metadata-based system, while less expressive than the others, still outperformed the random baseline and showed comparable results. The best configuration employed the USE architecture. USE's contextual understanding enabled effective semantic matching, particularly at lower k values, although the BERT model showed competitive performance as k increased.

In conclusion, while the pixel-based system excels at pinpointing the single most relevant item ($k = 1$), the emotion-based system becomes more effective as the list grows ($k = 10$). These findings highlight the complementary strengths of different content features.

Chapter 5

Conclusion and Future Work

This chapter will focus on summarizing the results and contributions and providing insights into how to improve the work in the future.

5.1 Conclusions

The dissertation presents a comprehensive analysis of a recommendation system based on four types of models: emotion-based, pixel-based, annotation-based and random. The study explores the development of using these four models and various parameters to recommend various values of k. The findings provide insights into the strengths and limitations of each approach and their possible potential and applicability in recommender systems.

To address the initial research goals, two research questions were defined and answered based on the experimental results:

RQ1: Can emotions reported by different users be used as a single feature for recommending images?

Yes, the emotions reported by different users can be effective when used as the sole characteristic for recommending images. By predicting whether a user likes or dislikes an image through emotions, the system is able to capture the user's preference. This process is even better when longer lists of images are recommended.

RQ2: Is an emotion-based recommendation system more effective than traditional approaches such as pixel-based, metadata-based, and random baselines?

Overall, the emotion-based recommendation system outperforms the random system.When compared with metadata-based systems, the system showsWhen compared to the metadata-based systems, the system mostWhen compared to metadata-based systems, the moWhen compared to metadata-based systems, the m When compared to metadata-based systems, the emotion-based system is superior. When compared to the pixel-based recommendation system, emotions are better when we want a larger recommendation list, but when the recommendation involves fewer items, pixels are still superior.

This behavior of the models reveals that emotions contribute to better generalization the larger the lists considered, probably due to the subjective nature of emotional characteristics, and that it manages to prioritize relevant items consistently regardless of the list size. However, this characteristic may be due to the lack of data that this metric presents. Nevertheless, average emotions can also limit the model by failing to capture the individual variability of emotional responses to images. The metadata-based models performed well for many of the metrics, but pixels and emotions performed better in most cases. These models' limitations may be due to their difficulty in capturing the full meaning of the image.

This work emphasizes the importance of a recommender system's chosen features and different parameters. By comparing pixels, emotions, and metadata, the study provides insight into each approach's strengths and weaknesses. As expected, the results show the robustness of pixels in recommending images. However, they also show that emotions significantly contribute to and weigh in this type of task.

5.2 Future Work

Future research could explore incorporating hybrid recommendation systems, using pixels and emotions to enhance the system, leveraging the strengths of each approach, and mitigating the disadvantages. A hybrid system with a collaborative filtering recommendation system could also be interesting. Dimensionality reduction, particularly for pixels and annotations, could also be improved by simplifying the vectors so that the essentials are present. This could lead to more precise recommendations and could be done using a PCA, UMAP, or t-SNE. Other models, such as Contrastive Language-Image Pretraining (CLIP), could be tested for feature extraction. Experimenting with more similarity measures, such as Pearson's correlation or Manhattan distance, could help the system improve its recommendation. Testing more complex similarity measures, such as the Cross-modal Semantic Gap for Multi-modal Recommendation or adaptive similarity measures, could dynamically adjust to the user's preferences, adding value to this type of recommendation. Carrying out user studies would also be of added value insofar as they provide a general evaluation of the system. In addition to these improvements, it could also be interesting to try to predict emotions based on the images using, for example, a random foster with a multi-output classifier and using these generated emotions to make the recommendation.

5.3 Final Remarks

This dissertation demonstrates the potential that emotions have for creating a content-based recommender system. While the results demonstrate the capabilities of baseline models, they also pave the way for new approaches to improve personalization and user recommendations in future recommender systems. Addressing the issues that need to be improved in future work can serve as a foundation for further research into content-based recommender systems.

References

[1] K. Lawrence, R. Campbell, and D. Skuse, “Age, gender, and puberty influence the development of facial emotion recognition,” *Frontiers in Psychology*, vol. 6, 6 2015. XI, 6

[2] G. Maupomé and O. Isyutina, “Dental students’ and faculty members’ concepts and emotions associated with a caries risk assessment program,” *Journal of dental education*, vol. 77, pp. 1477–87, 11 2013. XI, 6

[3] S. Indolia, A. K. Goswami, S. P. Mishra, and P. Asopa, “Conceptual understanding of convolutional neural network- a deep learning approach,” in *Procedia Computer Science*, vol. 132, 2018. XI, 8

[4] S. V. Gaikwad, “Text mining methods and techniques,” pp. 975–8887, 2014. XI, 8, 9

[5] M. Barros, “Recommender system to support comprehensive exploration of large scale scientific datasets,” Ph.D. dissertation, UNIVERSIDADE DE LISBOA, FACULDADE DE CIÊNCIAS, 2021. XI, 13

[6] S. M. Alarcão, “An affective computing and image retrieval approach to support diversified and emotion-aware reminiscence therapy sessions,” Ph.D. dissertation, UNIVERSIDADE DE LISBOA, FACULDADE DE CIÊNCIAS, 2022. 5

[7] J. D. Ad Vingerhoets, Iven Nyklícek, *Emotion Regulation, Concpetual and Clinical Issues*. Springer, 2008. 5

[8] A. Bechara, H. Damasio, and A. R. Damasio, “Emotion, decision making and the orbitofrontal cortex,” *Cerebral Cortex*, vol. 10, pp. 295–307, March 2000. 5

[9] A. V. Geetha, T. Mala, D. Priyanka, and E. Uma, “Multimodal emotion recognition with deep learning: Advancements, challenges, and future directions,” 5 2024. 5, 7

[10] C. E. Izard, “The many meanings/aspects of emotion: Definitions, functions, activation, and regulation,” pp. 363–370, 2010. 5

[11] P. Ekman, *Basic emotions.* John Wiley Sons Lda, 1999. 5

[12] R. Plutchik, “The nature of emotions: Human emotions have deep evolutionary roots, a fact that may explain their complexity and provide tools for clinical practice,” *American Scientist*, vol. 89, no. 4, pp. 344–350, 2001. [Online]. Available: http://www.jstor.org/stable/27857503 5

[13] L. F. Barrett and J. A. Russell, “The structure of current affect: Controversies and emerging consensus,” *Current Directions in Psychological Science*, vol. 8, no. 1, pp. 10–14, 1999. [Online]. Available: https://doi.org/10.1111/1467-8721.00003 6

[14] M. M. Bradley and P. J. Lang, “Measuring emotion: The self-assessment manikin and the semantic differential,” pp. 49–59, 1994. 6

[15] J. A. Russell, “Core affect and the psychological construction of emotion,” *Psychological Review*, vol. 110, pp. 145–172, 2003. 6

[16] C. A. Smith and R. S. Lazarus, “Appraisal components, core relational themes, and the emotions,” *Cognition and Emotion*, vol. 7, no. 3-4, pp. 233–269, 1993. [Online]. Available: https://doi.org/10.1080/02699939308409189 6

[17] A. Mishra, “A comprehensive review of artificial intelligence and machine learning: Concepts, trends, and applications,” *International Journal of Scientific Research in Science and Technology*, vol. 11, no. 5, pp. 126–142, Sep. 2024. [Online]. Available: https://ijsrst.com/index.php/home/article/view/IJSRST2411587 7

[18] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, “Review of deep learning: concepts, cnn architectures, challenges, applications, future directions,” *Journal of Big Data*, vol. 8, 12 2021. 7

[19] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, “A survey of convolutional neural networks: Analysis, applications, and prospects,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, pp. 6999–7019, 12 2022. 7, 39

[20] R. Talib, M. K. Hanif, S. Ayesha, and F. Fatima, “Text mining: Techniques, applications and issues,” 2016. [Online]. Available: www.ijacsa.thesai.org 9

[21] M. Polignano, F. Narducci, M. de Gemmis, and G. Semeraro, “Towards emotion-aware recommender systems: an affective coherence model based on emotion-driven behaviors,” *Expert Systems with Applications*, vol. 170, 5 2021. 9, 26

[22] C. C. Aggarwal, *Recommender Systems*. Springer International Publishing, 2016. [Online]. Available: http://link.springer.com/10.1007/978-3-319-29659-3 10, 12, 13, 14

[23] Y. Deldjoo, M. Schedl, P. Cremonesi, and G. Pasi, “Recommender systems leveraging multimedia content,” *ACM Computing Surveys*, vol. 53, 9 2020. 12

[24] H. Al-bashiri, M. Abdulhak, A. Romli, and F. Hujainah, “Collaborative filtering recommender system: Overview and challenges,” *Advanced Science Letters*, vol. 23, pp. 9045–9049, 09 2017. 12

[25] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, “Recommendation systems: Principles, methods and evaluation,” pp. 261–273, 11 2015. 12, 13, 20

[26] R. Burke, “Hybrid recommender systems: Survey and experiments,” *User Modelling and User-Adapted Interaction*, vol. 12, pp. 331–370, 2002. 13

[27] L. M. de Campos, J. M. Fernández-Luna, J. F. Huete, and M. A. Rueda-Morales, “Combining content-based and collaborative recommendations: A hybrid approach based on bayesian networks,” *International Journal of Approximate Reasoning*, vol. 51, no. 7, pp. 785–799, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888613X10000460 13

[28] S. Milano, M. Taddeo, and L. Floridi, “Recommender systems and their ethical challenges,” *AI and Society*, vol. 35, pp. 957–967, 12 2020. 14

[29] S. Khusro, Z. Ali, and I. Ullah, “Recommender systems: Issues, challenges, and research opportunities,” in *Lecture Notes in Electrical Engineering*, vol. 376. Springer Verlag, 2016, pp. 1179–1189. 14

[30] D. Jannach, L. Lerche, and M. Zanker, *Recommending Based on Implicit Feedback*. Springer, 05 2018, pp. 510–569. 17

[31] J. Han, M. Kamber, and J. Pei, *Getting to Know Your Data*. Elsevier, 2012, pp. 39–82. 18

[32] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook: Third Edition*. Springer US, 1 2022. 18, 20

[33] P. Kumar and R. S. Thakur, “Recommendation system techniques and related issues: a survey,” *International Journal of Information Technology (Singapore)*, vol. 10, pp. 495–501, 12 2018. 19

[34] A. R. Sulthana, M. Gupta, S. Subramanian, and S. Mirza, “Improvising the performance of image-based recommendation system using convolution neural networks and deep learning,” *Soft Computing*, vol. 24, pp. 14 531–14 544, 10 2020. 22

[35] A. K. Jaiswal, H. Liu, and I. Frommholz, “Effects of foraging in personalized content-based image recommendation,” *CoRR*, vol. abs/1907.00483, 2019. [Online]. Available: http://arxiv.org/abs/1907.00483 22

[36] A. Ahmed, “Pre-trained cnns models for content based image retrieval,” *International Journal of Advanced Computer Science and Applications*, vol. 12, pp. 200–206, 2021. 23

[37] Z. Kurt and K. Özkan, "Öznitelik Çikarm tekniklerine dayali imge Öneri sistemi," in *2nd International Conference on Computer Science and Engineering, UBMK 2017*. Institute of Electrical and Electronics Engineers Inc., 10 2017, pp. 769–774. 23

[38] A. Angadi, S. K. Gorripati, V. Rachapudi, Y. K. Kuppili, and P. Dileep, "Image-based content recommendation system with cnn," in *Proceedings of the 5th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), I-SMAC 2021*. Institute of Electrical and Electronics Engineers Inc., 2021, pp. 1260–1264. 23

[39] Y. Deldjoo, T. D. Noia, D. Malitesta, and F. A. Merra, "A study on the relative importance of convolutional neural networks in visually-aware recommender systems," 2020. [Online]. Available: https://github.com/sisinflab/CNNs-in-VRSs 24

[40] V. Dominguez, I. Donoso-Guzmán, P. Messina, and D. Parra, "The effect of explanations and algorithmic accuracy on visual recommender systems of artistic images," vol. Part F147615. Association for Computing Machinery, 2019, pp. 408–416. 24

[41] S. K. Addagarla and A. Amalanathan, "Probabilistic unsupervised machine learning approach for a similar image recommender system for e-commerce," *Symmetry*, vol. 12, pp. 1–17, 11 2020. 24

[42] V. Moscato, A. Picariello, and G. Sperli, "An emotional recommender system for music," *IEEE Intelligent Systems*, vol. 36, pp. 57–68, 2021. 25

[43] J. Mizgajski and M. Morzy, "Affective recommender systems in online news industry: how emotions influence reading choices," *User Modeling and User-Adapted Interaction*, vol. 29, pp. 345–379, 4 2019. 25

[44] M. Tkalčič, A. Odić, A. KoTkalšičir, and J. Tasič, "Affective labeling in a content-based recommender system for images," *IEEE Transactions on Multimedia*, vol. 15, pp. 391–400, 2013. 26

[45] M. Tkalčič, U. Burnik, and A. Košir, "Using affective parameters in a content-based recommender system for images," *User Modeling and User-Adapted Interaction*, vol. 20, pp. 279–311, 2010. 26

[46] P. Lops, M. D. Gemmis, G. Semeraro, C. Musto, and F. Narducci, "Content-based and collaborative techniques for tag recommendation: An empirical evaluation," *Journal of Intelligent Information Systems*, vol. 40, pp. 41–61, 2 2013. 27

[47] A. B. Nidhi R.H, *ICCIDS : 2017 International Conference on Computational Intelligence in Data Science : 2-3 June 2017*. IEEE, 2018. 28

[48] K. Samosir and F. Ginting, "A comparative analysis of content-based filtering and tf-idf approaches for enhancing sports recommendation systems," pp. 90–97, 2024. [Online]. Available: http://innovatics.unsil.ac.id 28

[49] V. T. Nguyen, K. D. Le, M. T. Tran, and M. Fjeld, "Now and then: A social network-based photo recommendation tool supporting reminiscence." Association for Computing Machinery, 12 2016, pp. 159–168. 28

[50] T. Schultz, F. Putze, L. Steinert, R. Mikut, A. Depner, A. Kruse, I. Franz, P. Gaerte, T. Dimitrov, T. Gehrig, J. Lohse, and C. Simon, "I-care-an interaction system for the individual activation of people with dementia," *Geriatrics (Switzerland)*, vol. 6, 6 2021. 29

[51] L. Oliva-Felipe, E. Wolverson, K. Votis, C. Barrué, M. Antomarini, I. Paliokas, A. Cortés, I. Landrin, and U. Cortés, "Health recommender system design in the context of caregiverspro-mmd project." Association for Computing Machinery, 6 2018, pp. 462–469. 29

[52] F. Gräßer, S. Beckert, D. Küster, J. Schmitt, S. Abraham, H. Malberg, and S. Zaunseder, "Therapy decision support based on recommender system methods," *Journal of Healthcare Engineering*, vol. 2017, 2017. 30

[53] P. Chinnasamy, W. K. Wong, A. A. Raja, O. I. Khalaf, A. Kiran, and J. C. Babu, "Health recommendation system using deep learning-based collaborative filtering," *Heliyon*, vol. 9, 12 2023. 30

[54] I. B. Mazlan, N. Abdullah, N. Ahmad, and S. Z. Harun, "A review of personalized recommender system for mental health interventions," 2024. [Online]. Available: www.ijacsa.thesai.org 30

[55] J. A. Mikels, B. L. Fredrickson, G. R. Larkin, C. M. Lindberg, S. J. Maglio, and P. A. Reuter-Lorenz, "Emotional category data on images from the international affective picture system," 2005. [Online]. Available: www.psychonomic.org/archive/. 34

[56] P. J. Lang, M. M. Bradley, B. N. Cuthbert, M. Greenwald, A. Öhman, D. Vaitl, A. Hamm, E. Cook, A. Bertron, M. Petry, R. Bruner, M. McManis, D. Zabaldo, S. Martinez, S. Cuthbert, D. Ray, K. Koller, M. Kolchakian, M. Pappenheimer, A. Calpe, S. Eichler, S. Hayden, M. Karlsson, K. Barber, and A. Bittiker, "International affective picture system (iaps); 2008 1 international affective picture system (iaps): Affective ratings of pictures and instruction manual international affective picture system (iaps); 2008 2 international affective picture system (iaps): Technical manual and affective ratings," 2008. 34

[57] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009, pp. 248–255, accessed: 2024-12-15. [Online]. Available: http://www.image-net.org 36

[58] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 9 2014. [Online]. Available: http://arxiv.org/abs/1409.1556 36

[59] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 12 2015. [Online]. Available: http://arxiv.org/abs/1512.03385 37

[60] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” 12 2015. [Online]. Available: http://arxiv.org/abs/1512.00567 38

[61] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” 10 2016. [Online]. Available: http://arxiv.org/abs/1610.02357 38

[62] Q. Shi, Q. Liu, B. Chen, Y. Zhang, T. Liu, and J.-G. Lou, “Lemon: Language-based environment manipulation via execution-guided pre-training,” 1 2022. [Online]. Available: http://arxiv.org/abs/2201.08081 40

[63] C. Sammut and G. I. Webb, *TF-IDF*. Boston, MA: Springer US, 2010, pp. 986–987. [Online]. Available: https://doi.org/10.1007/978-0-387-30164-8_832 40

[64] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” 10 2018. [Online]. Available: http://arxiv.org/abs/1810.04805 40

[65] D. Cer, Y. Yang, S. yi Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y.-H. Sung, B. Strope, and R. Kurzweil, “Universal sentence encoder,” 3 2018. [Online]. Available: http://arxiv.org/abs/1803.11175 40

[66] A. R. Openai, K. N. Openai, T. S. Openai, and I. S. Openai, “Improving language understanding by generative pre-training,” 2018. [Online]. Available: https://gluebenchmark.com/leaderboard 41

[67] M. E. Peters, M. Neumann, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018, pp. 2227–2237. [Online]. Available: http://allennlp.org/elmo 41

Appendix A

Extra Information

This chapter includes supplementary information that, although not crucial to the main discussion, offers valuable context and further details to support a deeper understanding of the work.

A.1 EmoRecSys Project Dataset

In the next section, we will show the organization of the dataset of the group in which this work is inserted. In the tables A.1, A.2 and A.3 we can see each field, as well as its description and an example.

Table A.1: Survey Table: Information about survey participants, including demographic details, consent status, and survey-related data.

Column	**Description**	**Null**	**Data Type**	**Example**
id	Unique identifier of the survey's participant	No	Int	1
age	Age of the inquired person	No	Char	23
populational_aff	Population affiliation	No	Char	Caucasian
gender	Gender identified by the person	No	Enum	F
education	Education level of the person	No	Enum	Bachelor's Degree
city	City of residence	Yes	Char	Lisbon
country_residence	Country of residence	No	Char	Portugal
date_survey	Date of survey response	No	Datetime	2023-11-08 15:32:10
consented	User consented to the survey	No	TinyInt	1
hobby_other	Other hobbies mentioned	Yes	Varchar	Photography

Table A.2: Photos Table: Details about the images used in the study, such as file properties, dimensions, source, and alternative descriptions.

Column	**Description**	**Null**	**Data Type**	**Example**
id	Unique identifier of the photo	No	Int	1
file_name	Name of the photo file	Yes	Varchar	sunset
ext	File extension	No	Char	jpg
views	Number of times the photo was viewed	Yes	BigInt	1523
id_source	Foreign key for the source table	Yes	Int	3
source	Name of the image source	Yes	Varchar	Pexels
height	Image height in pixels	Yes	Int	1080
width	Image width in pixels	Yes	Int	1920
alt	Alternative text for the image	Yes	Text	A beautiful sunset over the ocean

Table A.3: Ratings Table: Participant ratings of emotions associated with photos, covering basic emotions as well as valence, arousal, and dominance scores.

Column	Description	Null	Data Type	Example
id	Unique identifier of the rating	No	Int	1
id_photo	Foreign key for the photo table	No	Int	10
id_survey	Foreign key for the survey table	No	Int	5
like_bool	Like/dislike of the photo	No	TinyInt	1
anger	Level of anger felt [0; 5]	No	Int	2
fear	Level of fear felt [0; 5]	No	Int	3
disgust	Level of disgust felt [0; 5]	No	Int	1
sadness	Level of sadness felt [0; 5]	No	Int	4
happiness	Level of happiness felt [0; 5]	No	Int	5
surprise	Level of surprise felt [0; 5]	No	Int	3
neutral	Level of neutrality felt [0; 5]	No	Int	2
valence	Level of valence felt [0; 5]	Yes	Int	1
arousal	Level of arousal felt [0; 5]	Yes	Int	3
dominance	Level of dominance felt [0; 5]	Yes	Int	5

A.2 Results for different hyperparameters

In the following section, we will present the results in order to analyze the top 10 best models. Showing each metric: Precision, Recall, F1-Score, and nDCG.

A.3 Emotion-based top 10

Table A.4: Top 10 dimensional emotion-based model configurations based on Precision@1, including @5 and @10.

Model	Similarity	Score	n	Precision@1	Precision@5	Precision@10
Dimensional	Euclidean	Weighted average	9	0.11412	0.12148	0.12080
Dimensional	Euclidean	Weighted average	8	0.11002	0.12204	0.12021
Dimensional	Euclidean	Weighted average	11	0.10676	0.12438	0.12096
Dimensional	Euclidean	Weighted average	7	0.10553	0.12137	0.11725
Dimensional	Euclidean	Weighted average	10	0.10430	0.12577	0.12066
Dimensional	Euclidean	Weighted average	8	0.10674	0.12246	0.12270
Dimensional	Euclidean	Weighted average	9	0.11166	0.12074	0.12060
Dimensional	Euclidean	Weighted average	10	0.10430	0.12416	0.11974
Dimensional	Euclidean	Weighted average	7	0.10430	0.12542	0.11892
Dimensional	Euclidean	Weighted average	8	0.10552	0.12122	0.11596

Table A.5: Top 10 dimensional emotion-based model configurations based on Recall@1, including @5 and @10.

Model	Similarity	Score	n	Recall@1	Recall@5	Recall@10
Dimensional	Euclidean	Weighted average	9	0.05267	0.27679	0.54898
Dimensional	Euclidean	Weighted average	8	0.05049	0.28119	0.54526
Dimensional	Euclidean	Weighted average	11	0.04704	0.28712	0.54580
Dimensional	Euclidean	Weighted average	7	0.05073	0.27934	0.52955
Dimensional	Euclidean	Weighted average	10	0.04816	0.28784	0.54725
Dimensional	Euclidean	Weighted average	8	0.04906	0.28326	0.55604
Dimensional	Euclidean	Weighted average	9	0.05216	0.27568	0.54726
Dimensional	Euclidean	Weighted average	10	0.04866	0.28202	0.54254
Dimensional	Euclidean	Weighted average	7	0.04848	0.29182	0.53824
Dimensional	Euclidean	Weighted average	8	0.04844	0.27708	0.52494

Table A.6: Top 10 dimensional emotion-based model configurations based on F1@1, including @5 and @10.

Model	Similarity	Score	n	F1@1	F1@5	F1@10
Dimensional	Euclidean	Weighted average	9	0.07046	0.16495	0.19482
Dimensional	Euclidean	Weighted average	8	0.06769	0.16621	0.19382
Dimensional	Euclidean	Weighted average	11	0.06402	0.16924	0.19490
Dimensional	Euclidean	Weighted average	7	0.06644	0.16514	0.18893
Dimensional	Euclidean	Weighted average	10	0.06431	0.17092	0.19451
Dimensional	Euclidean	Weighted average	8	0.06584	0.16692	0.19780
Dimensional	Euclidean	Weighted average	9	0.06956	0.16402	0.19448
Dimensional	Euclidean	Weighted average	10	0.06484	0.16846	0.19302
Dimensional	Euclidean	Weighted average	7	0.06440	0.17108	0.19164
Dimensional	Euclidean	Weighted average	8	0.06482	0.16492	0.18696

Table A.7: Top 10 dimensional emotion-based model configurations based on nDCG@1, including @5 and @10.

Model	Similarity	Score	n	nDCG@1	nDCG@5	nDCG@10
Dimensional	Euclidean	Weighted average	9	0.11412	0.20397	0.31305
Dimensional	Euclidean	Weighted average	8	0.11002	0.20433	0.31080
Dimensional	Euclidean	Weighted average	11	0.10676	0.20522	0.30954
Dimensional	Euclidean	Weighted average	7	0.10553	0.20300	0.30462
Dimensional	Euclidean	Weighted average	10	0.10430	0.20610	0.31101
Dimensional	Euclidean	Weighted average	8	0.10674	0.20496	0.31516
Dimensional	Euclidean	Weighted average	9	0.11166	0.20228	0.31146
Dimensional	Euclidean	Weighted average	10	0.10430	0.20306	0.30820
Dimensional	Euclidean	Weighted average	7	0.10430	0.20774	0.30822
Dimensional	Euclidean	Weighted average	8	0.10552	0.19984	0.29964

A.4 Pixel-based top 10

Table A.8: Top 10 model configurations based on Precision@1, including similarity type, score type, and profile size (n).

Model	**Similarity**	**Score**	**n**	**Precision@1**	**Precision@5**	**Precision@10**
Xception	Euclidean	Squared weighted average	7	**0.16442**	0.12074	0.10798
Xception	Euclidean	Simple average	12	0.15952	**0.12392**	0.10306
Xception	Euclidean	Weighted average	7	0.15952	0.12002	0.10774
Xception	Euclidean	Simple average	1	0.15950	0.11654	0.09388
Xception	Euclidean	Weighted average	1	0.15950	0.11654	0.09388
Xception	Euclidean	Squared weighted average	1	0.15950	0.11654	0.09388
ResNet50	Euclidean	Squared weighted average	11	0.15828	0.10970	0.10270
Xception	Euclidean	Squared weighted average	5	0.15826	0.11166	0.10170
Xception	Cosine	Squared weighted average	12	0.15704	0.11878	**0.10762**
Xception	Euclidean	Squared weighted average	12	0.15704	0.12294	0.10502

Table A.9: Top 10 model configurations based on Recall@1, including similarity type, score type, and profile size (n).

Model	**Similarity**	**Score**	**n**	**Recall@1**	**Recall@5**	**Recall@10**
Xception	Euclidean	Squared weighted average	7	**0.07200**	0.27446	0.48896
Xception	Euclidean	Simple average	12	0.07242	**0.27074**	0.45174
Xception	Euclidean	Weighted average	7	0.06870	0.27178	0.48772
Xception	Euclidean	Simple average	1	0.07138	0.25728	0.41226
Xception	Euclidean	Weighted average	1	0.07138	0.25728	0.41226
Xception	Euclidean	Squared weighted average	1	0.07138	0.25728	0.41226
ResNet50	Euclidean	Squared weighted average	11	0.06994	0.24602	0.45972
Xception	Euclidean	Squared weighted average	5	0.07526	0.25502	0.46116
Xception	Cosine	Squared weighted average	12	0.07036	0.26932	**0.48508**
Xception	Euclidean	Squared weighted average	12	0.07034	0.27588	0.47444

Table A.10: Top 10 model configurations based on F1@1, including similarity type, score type, and profile size (n).

Model	Similarity	Score	n	F1-Score@1	F1-Score@5	F1-Score@10
Xception	Euclidean	Squared weighted average	7	**0.09796**	0.16376	0.17396
Xception	Euclidean	Simple average	12	0.09716	**0.16664**	0.16544
Xception	Euclidean	Weighted average	7	0.09406	0.16262	0.17356
Xception	Euclidean	Simple average	1	0.09652	0.15730	0.15080
Xception	Euclidean	Weighted average	1	0.09652	0.15730	0.15080
Xception	Euclidean	Squared weighted average	1	0.09652	0.15730	0.15080
ResNet50	Euclidean	Squared weighted average	11	0.09448	0.14818	0.16512
Xception	Euclidean	Squared weighted average	5	0.09898	0.15166	0.16396
Xception	Cosine	Squared weighted average	12	0.09488	0.16112	**0.17344**
Xception	Euclidean	Squared weighted average	12	0.09510	0.16634	0.16930

Table A.11: Top 10 model configurations based on nDCG@1, including similarity type, score type, and profile size (n).

Model	Similarity	Score	n	nDCG@1	nDCG@5	nDCG@10
Xception	Euclidean	Squared weighted average	7	**0.16442**	0.21502	0.30158
Xception	Euclidean	Simple average	12	0.15952	**0.21942**	0.29282
Xception	Euclidean	Weighted average	7	0.15952	0.21170	0.29876
Xception	Euclidean	Simple average	1	0.15950	0.21034	0.27356
Xception	Euclidean	Weighted average	1	0.15950	0.21034	0.27356
Xception	Euclidean	Squared weighted average	1	0.15950	0.21034	0.27356
ResNet50	Euclidean	Squared weighted average	11	0.15828	0.19768	0.28378
Xception	Euclidean	Squared weighted average	5	0.15826	0.20680	0.28972
Xception	Cosine	Squared weighted average	12	0.15704	0.21144	**0.29858**
Xception	Euclidean	Squared weighted average	12	0.15704	0.21964	0.29928

A.5 Metadata-based top 10

Table A.12: Top 10 model configurations ranked by **Precision**.

Model	Similarity	Metric	n	k	Precision	Recall	F1-Score
USE	Euclidean	Squared weighted average	12	1	0.13496	0.06156	0.08264
USE	Euclidean	Weighted average	12	1	0.13372	0.06114	0.08202
TD-IDF	Cosine	Weighted average	5	1	0.12760	0.05582	0.07606
USE	Cosine	Weighted average	10	1	0.12638	0.05748	0.07690
USE	Cosine	Squared weighted average	12	1	0.12638	0.06074	0.07956
TD-IDF	Cosine	Weighted average	7	1	0.12514	0.05460	0.07444
USE	Euclidean	Weighted average	11	1	0.12394	0.05420	0.07382
USE	Euclidean	Squared weighted average	11	1	0.12392	0.05420	0.07384
TD-IDF	Cosine	Squared weighted average	5	1	0.12392	0.05438	0.07402
USE	Cosine	Weighted average	8	1	0.12392	0.05644	0.07568

Table A.13: Top 10 model configurations ranked by **Recall**.

Model	Similarity	Metric	n	k	Recall	Precision	F1-Score
USE	Euclidean	Squared weighted average	12	10	0.46502	0.10234	0.16506
USE	Euclidean	Weighted average	12	10	0.46276	0.10160	0.16392
USE	Cosine	Weighted average	10	10	0.45566	0.10086	0.16252
TD-IDF	Cosine	Weighted average	5	10	0.44498	0.10014	0.16104
TD-IDF	Cosine	Weighted average	7	10	0.44498	0.10062	0.16170
USE	Cosine	Squared weighted average	12	10	0.44090	0.09768	0.15740
USE	Euclidean	Weighted average	11	10	0.43006	0.09692	0.15584
USE	Euclidean	Squared weighted average	11	10	0.42944	0.09668	0.15548
TD-IDF	Cosine	Squared weighted average	5	10	0.44868	0.10048	0.16170
USE	Cosine	Weighted average	8	10	0.46134	0.10196	0.16436

Table A.14: Top 10 model configurations ranked by **F1-Score**.

Model	Similarity	Metric	n	k	F1-Score	Precision	Recall
USE	Euclidean	Squared weighted average	12	10	0.16506	0.10234	0.46502
USE	Euclidean	Weighted average	12	10	0.16392	0.10160	0.46276
USE	Cosine	Weighted average	10	10	0.16252	0.10086	0.45566
TD-IDF	Cosine	Weighted average	7	10	0.16170	0.10062	0.44498
TD-IDF	Cosine	Squared weighted average	5	10	0.16170	0.10048	0.44868
TD-IDF	Cosine	Weighted average	5	10	0.16104	0.10014	0.44498
USE	Cosine	Weighted average	8	10	0.16436	0.10196	0.46134
USE	Cosine	Squared weighted average	12	10	0.15740	0.09768	0.44090
USE	Euclidean	Weighted average	11	10	0.15584	0.09692	0.43006
USE	Euclidean	Squared weighted average	11	10	0.15548	0.09668	0.42944

Table A.15: Top 10 model configurations ranked by **nDCG**.

Model	Similarity	Metric	n	k	nDCG	Precision	Recall
USE	Euclidean	Squared weighted average	12	10	0.27634	0.10234	0.46502
USE	Euclidean	Weighted average	12	10	0.27512	0.10160	0.46276
USE	Cosine	Weighted average	8	10	0.27284	0.10196	0.46134
USE	Cosine	Weighted average	10	10	0.27002	0.10086	0.45566
USE	Cosine	Squared weighted average	12	10	0.26626	0.09768	0.44090
TD-IDF	Cosine	Squared weighted average	5	10	0.26412	0.10048	0.44868
TD-IDF	Cosine	Weighted average	5	10	0.26338	0.10014	0.44498
TD-IDF	Cosine	Weighted average	7	10	0.26216	0.10062	0.44498
USE	Euclidean	Weighted average	11	10	0.25752	0.09692	0.43006
USE	Euclidean	Squared weighted average	11	10	0.25720	0.09668	0.42944