Movie Recommendation based on *Stylistic* Visual Features

David Kvasnes Olsen

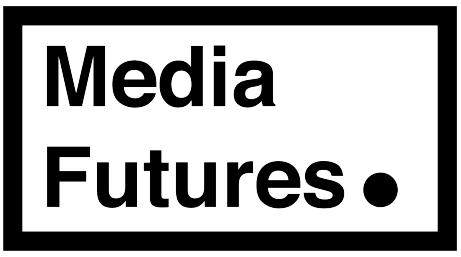
Supervisor: Assoc. Prof. Dr. Mehdi Elahi
Co-supervisor: Dr. Lars Skjærven

Master's Thesis
Department of Information Science and Media Studies
University of Bergen

June 1, 2022

Scientific environment

This study is carried out at the Department of Information Science and Media Studies, at the University of Bergen. The research work is a part of Work Package 2 (User Modeling, Personalisation and Engagement) at MediaFutures center. The work is conducted in collaboration with the media platform TV 2.



Acknowledgements

I would like to thank my supervisors, Prof. Dr. Mehdi Elahi and Dr. Lars Skjaerven, first and foremost. Throughout the entire process of developing and writing this thesis, they have served as a significant source of inspiration and support. I am extremely grateful to have worked under their supervision. The support, assistance, and level of participation they have demonstrated throughout this entire process have been exceptional. It was a privilege to work with and to learn from these two gentlemen. I would also like to thank MediaFutures and TV2 for the giving me the unique opportunity to working on this thesis. Last but not least, I would like to thank my girlfriend for always being encouraging, providing laughs, and overall being a great sport throughout this experience of writing this thesis from a broom closet in our living room; I could not have done it without you.

David Kvasnes Olsen
Bergen, June 2022

Abstract

When a new movie is added to the catalogue of a recommendation-empowered movie streaming platform, the system exploits various types of data (e.g., clicks, views, and ratings) in order to generate personalized recommendations for the users. However, in the absence of sufficient data, undesired situations can arise where the system may fail to include the new movie in the recommendation list. This is known as the *Cold Start* problem. A solution can be using content features attributed to the movies (e.g., tags, genre, and description). However, such features require expensive editorial efforts and it is not necessarily available in good quantity or quality.

This thesis investigates the viability of using novel *stylistic* visual features as metadata to incorporate in the movie recommendation process. The visual features represent the stylistic properties of the movies and can have a wide range of forms, e.g., color palette, contrast, and brightness. The stylistic visual features can be *automatically* extracted, and hence, do not require any (manual) human annotation. Accordingly, the thesis proposes a novel technique for generating recommendation based on such visual features and describes the technical details for different stages of the process. The technique has been evaluated in both offline and online experiments and different scenarios, i.e., cold start and warm start. The online experiment has been conducted in collaboration with TV 2, one of Noways largest digital streaming platforms adopting an A/B testing methodology. The proposed technique includes utilizing the extracted visual features when used individually (in a similarity based recommendation process), and when combined with other types of data (in a *hybrid* recommendation process). The results of the experiments have been promising and shown that the stylistic visual features can be beneficial particularly in the hybrid recommendation process in the cold start scenario.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Over the past 15 years streaming video content through digital streaming services has become much more commonplace. With this emerging trend it has become increasingly more difficult to navigate and pick out relevant content for consumers. This has led to the challenge known as *C*hoice Overload, where the consumers are presented with a plethora of options, but lack the personal experience of the options available to make good decisions on what to watch(*Elahi et al.*, 2019). User based video platforms such as Vimeo, or YouTube, are especially prone to Choice Overload. On YouTube alone 500 hours of video content is uploaded every minute (2020)[1], and a consumer is only able to browse through a tiny fraction of the content available. Recommender Systems are therefore an important part of curating short, personally tailored lists of content that can satisfy the users needs and preferences.

There has been an emergence of various video recommendation algorithms in all application domains in recent years. A popular approach to recommendation is Collaborative Filtering (CF), which utilities user similarity to generate recommendations. In the movie domain, this approach compares user data such as their ratings for items, or their watch-history and generates recommendations based on what similar users have watched and enjoyed previously. This approach performs well with respect to user satisfaction and diversity, but it falls short when there is a lack of data. Movies with little to no ratings, or users who have yet to watch a lot of content are both points in which the performance of this approach suffers. This is known as the New Item problem, which is a part of a larger set of problems related to recommender systems known as Cold Start(*Hazrati and Elahi*, 2021).

Another popular approach is Content-Based Filtering (CBF). These algorithms typically receive various input data, e.g., content features, which they use to calculate item-similarity to provide recommendations for consumers (*Beheshti et al.*, 2022). These algorithms are very effective in generating relevant recommendations for consumers, but much like their CF counterpart, they fall short when there is insufficient data. If a movie has little to no descriptive metadata, CBF also suffers from the cold start problem, and a CBF system will not be able to recommend the movie. To mitigate these limitations we can combine these approaches in a *Hybrid* recommendation process. A

[1] https://www.statista.com/statistics/259477/hours-of-video-uploaded-to-youtube-every-minute/

hybrid recommender system can combine the power user-item interactions and descriptive metadata about the items in order to provide meaningful recommendations to the user.

There are more limitations to CBF than the New Item problem. CBF is vulnerable to overspecialization, i.e. lacking the ability to provide diversity in recommendation. Another major problem with CBF in the movie domain; collecting quality data to represent the content itself. This data has traditionally been collected manually. Tags and genre requires manual labeling from a group of experts. The collection of quality data is also a problem for user-centric CF approaches as collecting sufficient amounts of ratings and views requires large amounts of user-item interaction which can be sparse and hard to collect. Therefore there is a place in this domain for automatically extracted features to help mitigate the need for this manual data annotation, and help aid in the New Item problem. Utilizing new automatically extracted features to represent media content could also help make recommendations that are less prone to human biases and errors. Automatically extracted features from movie trailers have already been demonstrated to provide promising results in generating movie recommendations (*Deldjoo et al.*, 2016a; *Kvifte et al.*, 2021; *Zhao et al.*, 2016). But still, to the best of the authors knowledge, most of these experiments have been done with open source non-commercial datasets, not on real-world digital streaming services.

1.2 Problem Statement

Although much progress has been made in the field of recommender systems, there are still challenges that need to be addressed. The issues vary from cold start challenge to diversity and scalability issues with recommender systems *Elahi et al.* (2021a). Typical content features used in movie recommendation (e.g., tags, genres, and descriptions) must be manually annotated, which is an expensive process and requires costly human involvement. However, a novel set of features based on visual analysis of the movies that can be retrieved automatically could help alleviate that problem (*Kvifte et al.*, 2021). As a result, visual features could aid with the cold start problem by offering descriptive features without requiring manual labor *Elahi et al.* (2020, 2021b). Furthermore, hybridizing individual approaches (i.e., CBF with CF) and incorporating the visual features could improve recommendation performance (*Kvifte et al.*, 2021). This thesis addresses the cold start problem by proposing a novel hybrid recommendation technique using stylistic visual features extracted from trailers and posters. The approach is evaluated in a predictive analysis comparing models using purely user-item interactions with hybrid recommendation models that use novel stylistic visual features. In order to account for factors that go beyond predictive analysis in an offline environment a novel content-based recommender system has been proposed and deployed on one of Norway's largest online streaming services TV2 Play evaluating the user interaction performance of this novel system in comparison with the current CF system they use. The purpose of this thesis is to investigate whether automatically extracted stylistic visual features extracted from trailers and posters can improve performance, aid in the cold start and new item problem for movie recommendation.

1.3 Research questions

To address the various facets of the general problem statement, the thesis attempts to answer the following research questions:

- **RQ1:** Can a hybrid recommender system utilizing *stylistic* visual features, automatically extracted from movies, improve quality of recommendation in comparison to the pure collaborative filtering?
  - **RQ1.1:** Can recommendation quality based on visual features, automatically extracted from movie “trailers", be improved in comparison to the pure collaborative filtering?
  - **RQ1.2:** Can recommendation quality based on visual features, automatically extracted from movie “posters", be improved in comparison to the pure collaborative filtering?
- **RQ2:** Can recommendation quality based on visual features, automatically extracted from movies be improved in the *Cold Start* scenario?
  - **RQ2.1:** Can recommendation quality based on visual features, automatically extracted from movie “trailers", be improved in the *Cold Start* scenario?
  - **RQ2.2:** Can recommendation quality based on visual features, automatically extracted from movie “posters", be improved in the *Cold Start* scenario?
- **RQ3:** How does a content-based recommender system, utilizing *stylistic* visual features, perform on a digital streaming platform in comparison to the pure collaborative filtering?

1.4 Contributions

The main contributions of my thesis include the following items:

- Proposing a novel hybrid recommendation technique based on stylistic visual features from movie trailers and posters.
- A comprehensive offline evaluation of a proposed novel hybrid recommendation technique in terms of loss, accuracy and beyond accuracy metrics.
- A comprehensive offline evaluation of a proposed novel hybrid recommendation technique in various stages of cold and warm start scenarios.
- Developing and deploying a novel content-based filtering recommendation technique on one of Norway’s largest digital streaming platforms TV2 Play.

1.5 Thesis outline

- **Chapter 2: Background.** The background chapter provides an overview of previous works and concepts relating to this thesis. Section 2.1 provides background knowledge relating to concepts within movie recommendation. Section 2.2 details previous works using visual features for movie recommendation. Section 2.3 summarises the previous works and details how this thesis differs from the previous works.
- **Chapter 3: Methods.** The Methods chapter describes the techniques and procedures used to answer the thesis's defined research questions. Section 3.1 covers the process of extracting visual features from trailers and posters. Section 3.2 details the process of converting raw visual features from keyframes to a single feature vector representing a trailer. Section 3.3 describes the datasets extracted from movie trailers and posters. Section 3.4 details the recommendation algorithms, implementation and metrics used in the offline and online evaluation.
- **Chapter 4: Results.** The results chapter details the analysis conducted on the accumulated datasets for various recommender models, and online tests. The chapter is categorized according to the various experiments conducted. Section 4.1, **Experiment A: Exploratory Analysis**, details the exploratory analysis of the *Trailer Features* and *Poster Features* datasets. Section 4.2, **Experiment B: Quality of recommendation**, details the offline evaluation using various recommendation algorithms in tandem with the *Trailer Features* and *Poster Features* datasets, and compares them with baseline pure CBF algorithms. Section 4.3, Experiment C: Cold Start simulation, details the metrics gathered by simulating the aforementioned algorithms in various cold start states. Section 4.4, **experiment D: Online Testing**, details the online tests performed on the digital streaming service TV2 Play.
- **Chapter 4: Conclusions and future work.** The conclusion chapter is dived into three Sections. Section 5.1 summarizes what was performed in the thesis. Section 5.3 covers the results according to the set research questions. Section 5.4 covers the limitations of the thesis, aswell as what could be done in future works.

Chapter 2

Background

The background chapter provides an overview of previous works and concepts relating to this thesis. Section 2.1 provides background knowledge relating to concepts within movie recommendation. Section 2.2 details previous works using visual features for movie recommendation. Section 2.3 summarises the previous works and details how this thesis differs from the previous works.

2.1 Movie Recommender Systems

Due to the expansive nature of the online space there is a continuous growth of available products to consumers. In line with this emergence of products, consumers are faced with more difficulty choosing relevant choices than ever before (*Hazrati et al.*, 2020). This leads to *Choice Overload*. A situation where the consumers are faced with a large amount of options, without having the sufficient personal experience to make a good decision (*Elahi et al.*, 2019). This is especially prevalent on online streaming platforms such as YouTube or Vimeo. These are platforms where there is an enormous amount of content uploaded each second of the day, and consumers are only able to scratch the surface of their catalog (*Hazrati and Elahi*, 2021). Recommender systems are therefore useful tools to help users navigate these Choice Overload prone platforms to help them curate short and personally tailored lists that satisfy their preferences, needs and constraints. *Elahi et al.* (2019)

There are multiple different approaches of creating personalized video recommendations, but the two most used methods are Collaborative Filtering (CF) and Content-based Filtering (CBF). CF is a user-centric approach where similar users are grouped together based on their previous ratings of items. These ratings can be given as explicit information in the form of likes / dislikes, favorites etc., or as implicit information in the form of viewing sessions. The numerical value of explicit feedback indicates preference, as opposed to the numerical value of implicit feedback which indicates confidence. (*Hu et al.*, 2008) This is because using explicit feedback a user expresses their preference of an item by giving it a rating, but implicit feedback describes the frequency of actions. A larger value does not indicate higher preference as a user might have loved a movie he/she watched only once, but the same user might watch a show he/she likes every week. However, this implicit feedback is definitively useful as recurring actions are more likely to reflect a users' opinion. *Hu et al.* (2008) An example

CF using implicit feedback could be a user *X* that has watched *Midsommar, Hereditary, and The Green Knight* might get recommended *Tusk*, because a similar user *Y* had watched *Midsommar, Hereditary, The Green Knight,* and *Tusk*.

In contrast, CBF is an item-centric strategy. This indicates that item similarity, as opposed to user similarity, is the basis for the recommendations (*Van Meteren and Van Someren*, 2000). If you've ever watched a movie on an internet streaming service and then been provided with a list titled "Because you saw Movie X," followed by a list of comparable films, you may be familiar with this technique. Here, the content, in this example movies, are suggested because they share comparable characteristics, such as tags, genre, description, and director.

Knowledge-based recommendation is a less popular but nonetheless significant approach to movie recommendation. These are systems in which the user inputs his or her particular preferences and tastes, and the system then offers the user with content tailored to those preferences and tastes (*Burke*, 2000).

When the system lacks adequate information to give recommendations, complications arise. This holds true for both user-centric CF systems and item-centric CBF systems. In a CF system, a movie that has never been viewed by a user will not be identified by the recommendation algorithm, and the same is true in a CBF system for a movie that lacks adequate metadata. This is known as the *Cold Start* problem. A related problem to the cold start problem is the *New Item* problem. The new item problem occurs when you upload a new item to a recommender system(*Elahi et al.*, 2019). A CF system will not be able to recommend this item as it likely has little to no views, and a CBF system will not be able to recommend this item if it lacks sufficient descriptive metadata. Using a hybrid-recommender system is one solution to this problem. This method combines user and item information in an effort to compensate for each other's deficiencies, hence enhancing the recommendations offered to users. An example could be in the absence of user-item interaction data for a particular movie, content-based algorithms could leverage available metadata about the movie be used to create a prediction.

In CBF the content is described with a set of descriptive features that represent the item. In the movie domain these sets of descriptive features can be divided into three hierarchical levels(*Deldjoo et al.*, 2016b, 2018);

1. At the highest level, there are semantic features pertaining to concepts or events in a film. An example could be the plot of the movie *La Haine*, which follows a day of the life of three young men in France after a violent riot;

2. There are syntactic elements at the intermediate level that deal with the objects and interactions in a film. In the aforementioned film, for instance, there are Vincent Cassel, Hubert Koundé, Saïd Taghmaoui, and various firearms and vehicles;

3. At the lowest level, there are stylistic features related to the mise-en-scène form of a movie, i.e., the design aspects that characterize the aesthetic and style of a movie (e.g., colors or textures); for instance, in the same film, the predominant colors are various shades of grey, and there are a lot of close-up camera shots of the main characters' faces.

2.2 Visual Features in Movie Recommender Systems

Most earlier work in the field of content-based video recommender systems has relied on semantic features. Structured data, such as genre, cast, and director, and unstructured data, such as tags, reviews, and plot, are examples of these semantic properties. More recent research has proved the feasibility of computationally retrieved low-level visual mise-en-scène features as the basis for suggestions(*Deldjoo et al.*, 2016a; *Moghaddam et al.*, 2019; *Rimaz et al.*, 2019).

Visual features are an inherently *stylistic* way of representing movies. One of the advantages of using visual features, in contrast to the aforementioned traditional features, is that they do not require expensive human-annotation, as they can be extracted automatically by computing features using *Computer Vision* techniques. Consequently, they could be a feasible option for movie recommendation in cold start, i.e., when proposing movies with insufficient descriptive manual features (*Elahi et al.*, 2019, 2018).

Zhao et al. (2016) addressed the gap in context-based recommendation by utilizing visual features extracted from movie posters and still frames from movies. They created a novel recommender system using visual features to enrich the knowledge of movies and users' preferences, stating that; a user may want to watch a movie the minute they see posters or still frames from movies. The authors note that these are unique features that can not be extracted from user ratings. The features extracted were color histograms and object detection from scale-invariant feature transformation. Their experiments showed using movie posters and still frames from movies could improve the rating predictions root mean square error (RMSE) score

Deldjoo et al. (2016a) proposed a novel content-based video recommendation system based on stylistic visual features extracted from both full length movies and movie trailers. The features they used in their experiments were Average Shot Length, Color Variance, Motion average, Motion standard deviation, and Lighting Key. Their findings were that using low-level stylistic visual features led to higher accuracy in recommending movies than traditional expert annotated features such as genre, cast, and reviews. Another important takeaway from this paper is the fact that they discovered a high correlation in cosine similarity between the features extracted from movie trailers and full length movies with the features they used.

Moghaddam et al. (2019) investigated whether low-level features extracted from movie trailers within the context of a hybrid recommender model could be used to predict movie popularity and ratings. Their reasoning for investigating this was that when a movie recommender system is unable to provide personalized recommendations due to a cold start, many systems will instead recommend popular movies. While popularity is typically determined by the number of ratings provided by existing users, this method may not work for new movies that have little to no ratings yet. Through the extraction and aggregation of low-level stylistic visual features from key frames of movie trailers, recommender models were trained to predict the popularity and rating of movies. While there is a correlation between rating and popularity, there is also a correlation between visual features and popularity, according to their experimental findings. Through predictive analysis, they confirm that their classification model can be used to predict the rating and popularity of a movie even before the entire film is released.

Rimaz et al. (2019) investigate the viability of using low-level visual features in movie recommendation systems. Using the MovieLens 1M dataset, visual features were extracted from 1800 movie trailers and combined with semantic features from corresponding movie data. The features they extracted from the movies were average shot length, mean of color variance, standard deviation of color variance, mean of motion average, mean of motion standard deviation, mean of lightning key, and the number of shots. In their exploratory analysis of visual features, the authors investigate the evolution of visual characteristics over time as well as visually similar clusters that may exist among movies. In their experimental evaluation comparing a recommender based on extracted visual features to models based on other content features such as genre, tags, and a combination of these, the visual features-based model outperformed the other models.

2.3 Previous works and differences

This chapter provided an overview of the existing literature related to the research questions of this thesis. The correlation between movie trailers and full length movies in terms of stylistic visual features has been shown to be high (*Deldjoo et al.*, 2016a). The use of key frames have been demonstrated as serving well as representations for full length movies(*Deldjoo et al.*, 2016a).The use of stylistic visual features has been demonstrated giving good results in terms of traditional accuracy metrics (*Rimaz et al.*, 2019; *Zhao et al.*, 2016). As far as the author is aware most previous works focus on offline evaluations and small-scale user studies, and no large scale deployment of a content-based recommendation system based on a real-world digital streaming platform has been performed. This thesis includes both offline evaluations for the proposed techniques, as well as an online experiment where a novel-content based recommendation technique utilizing stylistic features was built, deployed, and tested on one of Norway's biggest digital streaming platforms. Most previous work also focus on movie trailers, whereas this thesis also includes an offline evaluation done on recommendation algorithms utilizing stylistic visual features extracted from posters. *Zhao et al.* (2016) addressed the use of posters for movie recommendation, but did not include the magnitude of features proposed in this thesis.

Chapter 3

Methods

This chapter describes the techniques and procedures used to answer the thesis's defined research questions. Section 3.1 covers the process of extracting visual features from trailers and posters. Section 3.2 details the process of converting raw visual features from keyframes to a single feature vector representing a trailer. Section 3.3 describes the datasets extracted from movie trailers and posters. Section 3.4 details the recommendation algorithms, implementation and metrics used in the offline and online evaluation.

3.1 Feature Extraction

This section will describe the feature extraction utilized for generating recommendation. The section will be divided in to three parts. The first part will cover the extraction of key frames from the trailers, and the subsequent parts will cover visual feature extraction and feature aggregation, respectively.

3.1.1 Key-frame Extraction

A typical video file is displayed playing 24 frames (images) a second. This is done to create the illusion that objects within the frames are moving. But when we want to extract features from a video file the amount of frames in a videofile can become redundant as there is not much difference between one frame and the next. Therefore we use key frame extraction to extract a subset of frames from a video file that can be used as a representation of the video.

In this project we used the Python library Katna[1] to extract Key Frames. The Katna library has a built in keyframe extraction module that works by first dividing the movie into smaller chunks, then calling a *video frame extraction* module and a *frame selector* module in parallel.

Using absolute differences in the LUV colorspace, the *video frame extractor* module will return any video frames that are sufficiently dissimilar from the ones that before them when given an input video. Extracted frames are filtered based on their brightness scores. On the basis of entropy and contrast, extracted frames are filtered.

[1] https://katna.readthedocs.io/en/latest/

After frames are filtered depending on the needed number of frames, K-means clustering is used to construct K clusters. Afterward, clustering is performed using an image histogram-based approach. After K-means clustering has been completed, the variance of Laplacian sorting is used to each cluster to find the optimal frame from inside the cluster. For the detection of image blur in the field of image processing, the variance of the Laplacian technique is widely applied. This sorting and selection ensures the image with the least amount of blurring is selected from the cluster.

All movie trailers are downloaded in a 16:9 aspect ratio, regardless of the aspect ratio of the actual content. This means that the extracted keyframes from a movie trailer containing content in a 2.35:1 or 4:3 aspect ratio will have borders on top and bottom, or left and right, respectively. These borders could interfere with the visual features extracted from them, and therefore each frame is trimmed to only contain the content within the frame. This is done by recursively checking each side of the frame if the RGB values add up to a threshold we can set in the program. The threshold is set to 20, meaning that black, or close to black borders will be trimmed and only the content is kept.

Figure 3.1: Frame extracted containing content in 4:3 aspect ratio

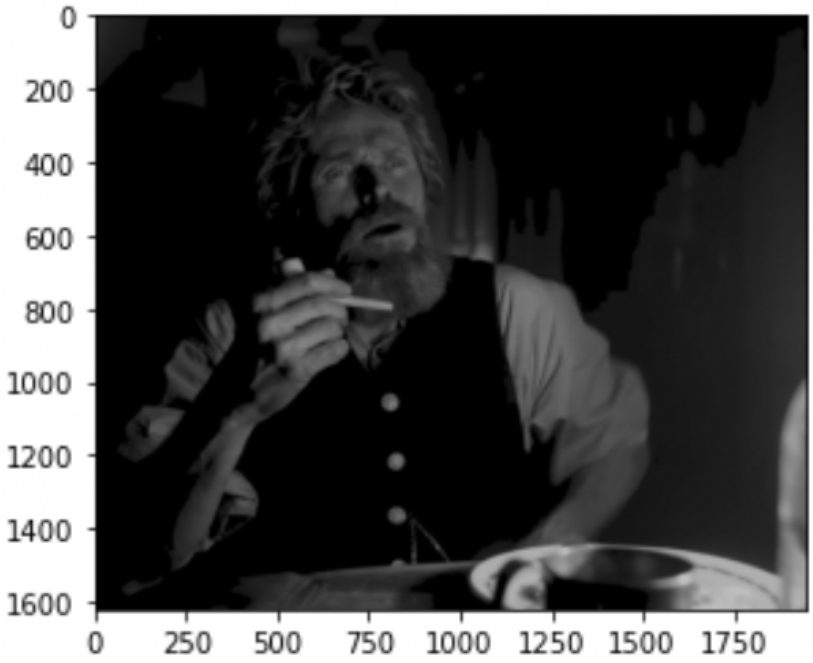


Figure 3.2: Frame extracted containing content in 4:3 aspect ratio trimmed

Figure 3.3: Frame extracted containing content in 2.35:1 aspect ratio trimmed

Figure 3.4: Frame extracted containing content in 2.35:1 aspect ratio trimmed

3.1.2 Visual Feature Extraction

The feature extraction was done utilizing the very popular OpenCV[2] (Open Source Computer Vision) Python implimentation. The following features were extracted from the trailers and movie posters.

Brightness

To extract the brightness of an image the image is converted from BGR color space to HSV (hue, saturation, value), also known as HSB (Hue, saturation, brightness). In HSV color space the V is synonymous with brightness, thus it can be used to extract the brightness of a given image. The brightness extraction function is defined as:

$$brightness = \frac{\sum V \times 100}{N \times 255}$$

Where S represents value, e.g., brightness, and N represents the number of pixels in the given image.

Saturation

Much like brightness, saturation is extracted by converting the image to HSV color space as the S in HSV denotes saturation. The S value is then used to extract the saturation of the image. The saturation extraction function is defined as:

$$saturation = \frac{\sum S \times 100}{N \times 255}$$

Where S represents the saturation, and N represents the number of pixels in the given image.

Entropy

To get the entropy from an image the Python library Scikit-Image[3] is used. The entropy function from SciKit-Image is computed using base 2 logarithm. This means that it is given an image as an array, and it will return the image with the minimum number of bits needed to encode the local grey level distribution. The attribute feature is calculated as:

$$entropy = \frac{E}{N}$$

Where E is the array returned from the imported Scikit-Image function, and N is the number of pixels in the image.

Sharpness

The sharpness feature is calculated by measuring the variation in the Laplacian of the image.

[2] https://opencv.org/
[3] https://scikit-image.org/

Contrast

The contrast feature is given by root mean square (RMS) contrast. This is calculated by converting the image to grey-scale and measuring the standard deviation of the pixel intensities.

Colorfulness

Given that colorfulness is a subjective measurement that relates to how we as humans interpret colorfulness in images. I have utilized the proposed colorfulness measure created by Sabine Süsstrunk and David Hasler in their article titled "Measuring colourfulness in natural images"(*Hasler and Suesstrunk*, 2003). Hasler and Süsstrunk conducted an experiment in which 20 volunteers were asked to rank 84 natural images on a 7-point scale ranging from 1 (no color) to 7 (intense color). After analyzing the data, the following was offered as a computationally efficient metric for calculating the colorfulness of an image:

$$rg = R - G,$$

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

$$\sigma_{rgyb} := \sqrt{\sigma_{rg}^2 + \sigma_{yb}^2},$$

$$\mu_{rgyb} := \sqrt{\mu_{rg}^2 + \mu_{yb}^2},$$

$$colorfulness = \sigma_{rgyb} + 0.3 \times \mu rgyb$$

Color palette

The color palette feature is intended to show the most dominant colors within a movie poster or movie trailer. To extract this feature K-means clustering is used as K-means requires a pre-given predetermined number of clusters k. The centroid of each cluster will be representing a dominant color in the image.

K-means works by dividing the color space into k clusters, with each cluster's *mean vector* m^k serving as a parameter. Since K-means is based on measuring the distance between data and cluster centroid vectors, a custom distance function $d : x \times m \to \mathbb{R}$ with mean μ_c can be employed. Given a set of color values $x_p \in x_1, ..., x_m$, the algorithm generates the following k centroids $\{m_1, ..., m_k\}$ as follows.(*Moosburger*, 2017):

1. Initiate random cluster centroids $\{m_1, ..., m_k\}$
2. Repeat until convergence:
    - Assign step: assign each x_p to the closest cluster for m_i for $1 \leq i \leq k$

$$C_i := P \in I | d(x_p, m_i) = \min_{1 \leq i \leq k} d(x_p, m_j)$$

- Update step for: m_i for $1 \leq i \leq k$

The process of generating a color palette goes as follows:

1. Read image(s) using OpenCV
2. Convert color space from BGR to RGB
3. Flatten the matrix of RGB values to a vector
4. Fit vector to K-means algorithm from Scikit-Learn with k-number of clusters.
5. Extract centroid from each cluster k
6. Rank cluster size by counting magnitude of labels in each respective cluster to get the dominance of the color
7. Use ranked cluster size to order the centroid and return k-number of dominant colors.

For a movie poster this process is run once. For a movie trailer *mt* the process repeats as step 1.,2. and 3. for all keyframes related to *mt*. Thus creating a vector $\mathbf{v}$ containing all the color values for all keyframes extracted from *mt* before clustering. For every movie 5 dominant colors are extracted, and for every movie trailer 10 dominant colors are extracted.

Figure 3.5: 'Babyteeth' and 'Amy' poster with their corresponding dominant color palette

3.1.3 Scene Detection

Another feature used is scene detection. Here we extract how many scenes there are in a movie trailer, and thus we can calculate the metrics shown in 3.2 for shot length in trailers.

To retrieve this feature a library known as PySceneDetect[4] was used. This library works by inputting a videofile which is read and each frame is converted to HSV color space. They have implemented an algorithm that measures the average difference between the H,S, and V values for each frame, and if the difference is over a set threshold this is recognized as a scene change. Using this library we get an index and a timestamp for each new scene in the videofile.

3.2 Feature Aggregation

As posters are a single image, the feature extraction process already produces a single vector for every poster. For movie trailers however, the features are extracted from multiple keyframes and feature aggregation is needed to condense the information down to a single vector to be used in recommendation tasks.

The process of feature extraction for movie trailers is run for N keyframes related to a movie trailer. Thus our raw data consists of N rows of raw features for every movie trailer. A correlated keyframe number ID is also present within the data. To aggregate all raw visual features from keyframes to a single vector representing a trailer the following techniques were applied:

- Arithmetic mean (average): The sum of numbers in a collection divided by the count of the numbers.
$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{1}{n}(x_1, ..., x_n)$$
- Median: The middle value that separates the lower half from the higher half in a dataset.
- Standard deviation: The square root of the variance of X
$$\sigma := \sqrt{E[(X-\mu)^2]} = \sqrt{\int_{-\infty}^{+\infty} (x-\mu)^2 f(x) dx},$$
- Polynomial regression: By using polynomial regression we can see if the feature change during the movie trailer. In this thesis I have used 1st and 2nd order polynomials.
$$y_i = \beta_0 + \beta_2 x_i + \beta_2 x_i^2 + ... + \beta_m x_i^m + \varepsilon_i (i = 1, 2, ..., n)$$

These metrics are calculated from each of the features[5], except for the dominant color palette feature. This is because the dominant color feature calculates the dominant color from all keyframes related to a movie trailer *mt*. This list of RGB values are split into color channels representing each RGB value, and joined with the rest of the features to make a vector $\mathbf{V}$ to represent each individual movie trailer.

[4] https://github.com/Breakthrough/PySceneDetect

[5] Polynomial regression was not applied to scene features due to errors occuring when trying to implement this

3.3 Datasets

After the feature extraction and feature aggregation two datasets were made combining viewing sessions from TV2 Play along with corresponding stylistic visual features for all movies and posters present in the dataset. A description for the dataset can be seen in Table 3.1. All trailers and posters were downloaded using TV2's internal API's.

Dataset	#Features	#Items	#Interactions	#Users
Trailer	64	546	742207	95100
Poster	21	1908	2509183	99454

Table 3.1: Dataset description

3.4 Recommendation Algorithms

All recommender algorithms used are implemented in the Python recommendation tool LibRecommender[6]. The recommendation algorithms and the number of training epochs run for each algorithm can be seen in Figure 3.2

Recommender Algorithm	Type	Number of epochs
Alernating Least Squares(ALS)	Pure CF	2
Bayesian personalized ranking (BPR)	Pure CF	3
Factorization Machines (FM)	Hybrid	3
Wide and Deep (WD)	Hybrid	2
AutoInt (AINT)	Hybrid	2

Table 3.2: Recommendation algorithms used

ALS and BPR are both pure CF algorithms optimized for implicit datasets. Both use matrix factorization which decomposes the user-item interactions into two lower dimensionality matrixes for users and items(*Hu et al.*, 2008; *Rendle et al.*, 2012). ALS' approach is to minimize the loss function implemented by alternating optimizing for users and items measured by the square error(*Hu et al.*, 2008). BPR's approach is a little different and instead uses an optimization critereon BPR-Opt and an algorithm known as LearnBPR for optimization (*Rendle et al.*, 2012). The biggest difference between these two algorithms is that BPR predicts the users preference for all item pairs instead of predicting a precise rating for each item. This in return makes BPR optimized for ranking purposes.(*Rendle et al.*, 2012)

FM, WD and AINT are all hybrid models, meaning that they take advantage of using both user-item interactions as well as metadata about the items to generate recommendations. FM was developed to combine the power of Support Vector Machines (SVM) with factorization models to make it work with sparse data as SVM's perform badly with sparse datasets (*Rendle*, 2010). WD and AINT are both based on neural networks,

[6]https://github.com/massquantity/LibRecommender

but have different approaches. WD combines wide linear models with deep neural networks to optimize memorisation and generalisation(*Cheng et al.*, 2016). Memorisation being learning the co-occurence of items and features based on historical data, and generalisation being the exploration of new feature combinations that are rarely found in the historical data(*Cheng et al.*, 2016). AINT on the otherhand is purely a neural network approach using self-attentive neural networks to calculate the probability of a user clicking on an item(*Song et al.*, 2019).

3.4.1 Content based recommendation for online evaluation

For the online evaluation we used cosine similarity as our recommendation algorithm. Cosine similarity popular is a technique for finding item-similarity in content-based recommendation. The cosine similarity between two items A and B is defined as (*Lops et al.*, 2011):

$$CosineSimilarity = cos(\theta) = \frac{A \cdot B}{||A||||B||}$$

To compute the cosine similarity between all movies and make a cosine similarity matrix we use a double for loop as such:

Algorithm 1: Process of generating cosine similarity matrix

```
Input: feature_vector_array
Output: cosine_similarity_matrix
n_rows = Number of rows in feature_vector_array
cosine_sim_array = Array of size n_rows ×n_rows
for row1 in n_rows do
    for row2 in n_rows do
        cosine_sim_array[row1,row2] = cos(feature_vector_array[row1],
          feature_vector_array[row2])
return cosine_sim_array
```

3.5 Technical details

For the offline evaluation all experiments were run in Python 3.10.4 running on Ubuntu 22.04 LTS. The hardware used was an AMD Ryzen 5 3600 CPU, 32GB RAM, and a NVIDIA GeForce RTX 3070 GPU.

3.6 Experiment design

This section will cover the design and metrics for evaluating the recommender system performance. This is done in two parts relating to the offline evaluation and the online evaluation, respectively.

3.6.1 Offline evaluation

There are two datasets used in the offline evaluation. The trailer dataset contains 742207 user interactions, 95100 users and with 546 movies with 64 dense numerical features. The poster dataset contains 2509183 user interactions, 99454 users and 1908 posters with 21 dense numerical features.

EXPERIMENT A: Exploratory analysis

In order to visualize the datasets generated various techniques were deployed. For item similarity cosine similarity was used, and for plot visualization PCA was used to reduce the dimensionality of the item vectors, and K-Means was used for clustering.

Standard linear principal components (PCA) are derived from the covariance matrix's eigenvectors and reveal the directions in which the data have the greatest variance(*Hastie et al.*, 2001). This also means that when provided with a feature vector PCA will transform the data so that the first principle component PC1 shows the largest variance, the second principle component PC2 shows the 2. most variance, etc., for N number of components. This is important at it allows us to plot multidimensional data using the principle components that have the largest variance. For the exploratory analysis section of results the PCA implementation from Sci-kit Learn was used.

EXPERIMENT B: Loss and accuracy metrics

For accuracy metrics the following metrics were used to predict the accuracy of the recommendations; Log loss, balanced accuracy, ROC-AUC, PR-AUC, precision@10, recall@10, Map@10, NDCG@10.

Log loss is a metric used to show the performance of a recommender system by calculating the cross entropy of the predicted rating from an actual rating. In the function below y denotes the actual rating, and $\hat{y}$ denotes the predicted rating calculated by the recommender system. This metric ranges from 0 to ∞, with the goal of our model being to minimize this value. A perfect model would have a log loss of 0. The metric is calculated as (*Vovk*, 2015):

$$LogLoss = \frac{1}{n}\sum(y \times log(\hat{y})) + (1-y) \times log(1-\hat{y})$$

Balanced accuracy is a metric that deals with imbalanced datasets, which is useful for implicit datasets oftentimes contain more positive data than negative data. The calculation of Balanced Accuracy is defined as (*Brodersen et al.*, 2010):

$$BalancedAccuracy = \frac{1}{2}(\frac{TP}{P} + \frac{TN}{N})$$

Where TP represents the true positives, P represents the predicted positives, TN represents the true negatives, and N represents the predicted negatives. A perfect Balanced Accuracy score would be 1.

Receiver Operator Characteristic - Area Under the Curve (ROC-AUC) is a metric that measures a recommender system's ability to distinguish between items liked by a user (relevant items) and all other items (irrelevant items) (*Lü et al.*, 2012). ROC-AUC is calculated by comparing the likelihood that relevant objects will be suggested

to that of irrelevant objects. For n independent comparisons (each comparison refers to selecting one relevant and one irrelevant object), if there are n instances in which the relevant object has a higher score than the irrelevant and n^n instances in which the scores are equal, then the relevant object is more important. (*Zhou et al.*, 2009)

$$RocAuc = \frac{n' + 0.5n^n}{n}$$

A perfect ROC-AUC score would be 1, and a random prediction model would have a ROC-AUC of 0.5.

Precision at top K recommendation (Precision@K) is a metric that measures the accuracy of a recommender system in commanding relevant items (*Schedl et al.*, 2018). To calculate $P@K$, for each user, the top K recommended items whose ratings also appear in the test set T are taken into account. For each user u, $P_u@K$ is computed as follows(*Schedl et al.*, 2018):

$$P_u@K = \frac{|L_u \cap \hat{L}_u|}{|\hat{L}_u|}$$

Where L_u is the set of relevant items for the user u in the test set T, and $\hat{L}_u$ is the set of the K items in T with the highest predicted ratings for the user u. The overall $P@K$ is then determined by averaging the $P_u@K$ values for every user in the test set.

Recall at top K recommendation ($Recall@K$) is a metric that shows the fraction of relevant items that are recommended at K positions. For a user u, $R_u@K$ is defined as:

$$R_u@K = \frac{|L_u \cap \hat{L}_u|}{|L_u|}$$

Where L_u is the set of relevant items for user u in the test set T and $\hat{L}_u$ denotes the recommended set containing the K items in T with the highest predicted ratings for a user u. The overall $R@K$ is calculated by averaging $R_u@K$ values for all the users in the test set.

Mean average precision at K ($Map@K$) is a rank-based metric that gives indication of the overall precision of a recommender system at different lengths of recommendation lists. The MAP@K score is computed as the arithmetic mean of the average precision over the entire set of users in the test set. Average precision for the top K recommendations (AP@K) is defined as (*Schedl et al.*, 2018):

$$AP@K = \frac{1}{N} \sum_{i=1}^{K} P@i \times rel(i)$$

Where $rel(i)$ is an indicator signaling if the i^th recommended item is relevant, i.e., $rel(i) = 1$, or not, i.e., $rel(i) = 0$; N is the total number of relevant items.

Normalized discounted cumulative gain at K ($NDCG@K$)is a measure used for the ranking quality of recommendations. Assuming that the recommendations for user u are sorted according to the predicted rating values in descending order $DCG_u@K$ is defined as (*Schedl et al.*, 2018):

$$DCG_{u@K} = \sum_{i=1}^{K} \frac{r_{u,i}}{log_2(i+1)}$$

where $r_{u,i}$ is the true rating found in the test set T for the item ranked at position i for user u, and K is the length of the recommendation list. Since the rating distribution depends on the users' behavior, the DCG values for different users are not directly comparable. Therefore, the cummulitive gain for each user should be normalized. This is done by computing the ideal DCG for user u, denoted as $IDCG_u$, which is the DCG_u value for the best possible ranking, obtained by ordering the items by true ratings in descending order(*Schedl et al.*, 2018). Normalized discounted cumulative gain for user u is then calculated as:

$$NDCG_u@K = \frac{DCG_u@K}{IDCG_u@K}$$

Finally, the overall normalized discounted cumulative gain at K $NDCG@K$ is computed by averaging the $NDCG_u@10$ over the entire set of users.

PR-AUC stands for the area under the (precision-recall) curve. To calculate this the Precision and Recall is needed. Precision can be defined as:

$$Precision = \frac{TP}{TP+FP}$$

Recall can be defined as:

$$Recall = \frac{TP}{TP+FN}$$

And finally the area under the curve can be found using the trapezoidal rule.

$$\int_a^b f(x)dx.$$

Beyond accuracy metrics

For beyond accuracy metrics catalog coverage and novelty metrics were used. The metrics were calculated on a dataset containing all users along with the top 7 recommendations from the various recommendation algorithms for every user. The number of recommendations was chosen to be 7 as the list on my TV presenting recommendations is 7 in length before you start scrolling. This is visualized in Figure 3.7.

Catalog coverage is a metric that shows the percentage of the total available set of items gets recommended by the recommendation algorithms. Catalog coverage is measured as (*Ge et al.*, 2010):

$$CatalogCoverage = \frac{|U_{j=1...N} I_L^j|}{|I|}$$

Where I_L^j is denoted as the set of all items contained in the list L returned by the j^{th} recommendations returned to users. N is the total number of recommendations observed during the measurement time, and I is the set of all available items, i.e., the catalog.

Novelty is a diversity metric that measures a recommender system's ability to generate novel and unexpected results that a user is unlikely to have encountered before. This metric is calculated using the self-information or "surprise" of recommended objects, which provides a measure of an object's unexpectedness relative to its global

popularity. Given an object α, the probability that a randomly chosen user has collected it is given by k_α/u; therefore, the object's self-information is $I_\alpha = log2(u/k_\alpha)$. From this, we can calculate the mean self-information $Ii(L)$ for each user's top L objects, as well as the mean top L surprise $I(L)$, also known as novelty, by averaging over all users with at least one deleted link. (*Zhou et al.*, 2010).

3.6.2 Online experiment

Implementation

For the online evaluation we deployed the feature extraction process on TV2's internal servers and generated recommendations based on cosine similarity. The recommendations were presented in lists such as the one shown in Figure 3.7. A simplified flowchart of the feature extraction process is shown in Figure 3.6.

Initially, the process of feature extraction was implemented using Jupyter Notebooks that read data from local CSV files. To deploy the project on TV2 Play, however, we had to refactor the entire codebase. As depicted in the simplified flowchart of Figure 3.6, the project was subdivided into a number of Python scripts that handled various tasks of the procedure. The CSV files utilized for data input and output were replaced with an Apache Kafka system and a database. Using internal APIs, the system would attempt to download the poster and trailer associated with each ID read from the database. If the system was able to collect the poster and/or trailer, it would analyze the items and report back to the database a single vector containing all features per item. Therefore, the system would be able to determine which Posters and Trailers it has already analyzed, and would be able to pick up where it left off in the event of a crash. The entire feature extraction process was made into a Docker[7] image and deployed onto one of TV2's Kubernetes[8] clusters.

Another service was implemented, being responsible for reading the feature vectors extracted, normalizing all the features, and applying cosine similarity to find the similarity between all items. The process of calculating the cosine similarity between items is explained in Section 3.4.1.

Recommendation lists were generated and presented to users as titled lists saying "Because you watched Movie X", with movie X being a movie the user has previously watched, and the recommendations consisting of the most visually similar movies to movie X. From this point on this content-based recommendation system based on stylistic visual features will be referred to as VF.

Experiment design and evaluation

An online experiment according to A/B testing methodology has been conducted on TV2 Play from the 23. to the 29. of May, comparing the user interactions between recommendations generated from VF and ALS. The recommendation lists were found under the Film category of TV2 Play's online platforms. The experiment was presented to 50 percent of the platforms users. The user interactions were measured in terms of views, clicks, and click-through-rate (CTR). To test the statistical significance of the

[7] https://www.docker.com/

[8] https://kubernetes.io/

experiment Fisher's exact test is applied to see if the P value is above 0.05. A P value below 0.05 signifies statistical significance.

For our test we will be calculating the Fisher's exact test P value on a matrix $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ where a and b refers to the clicks on a recommendation lists from recommendation list A and B, and c and d refers to the views on those recommendation lists which were not clicks from recommendation list A and B. The total sum of the matrix is then denoted as n. The P value is then calculated as (*Weisstein*):

$$p = \frac{(a+b)!(c+d)!(a+c)!(b+d)!}{a!b!c!d!n!}$$

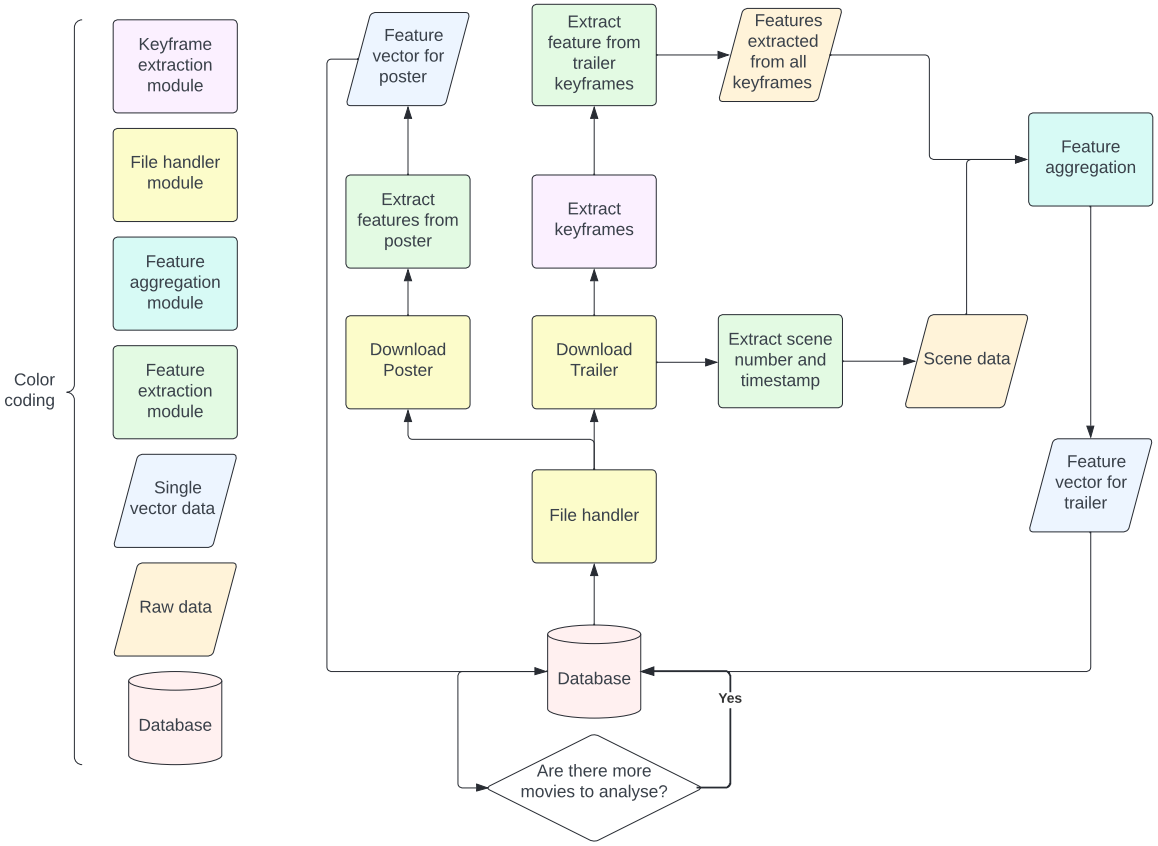


Figure 3.6: The feature extraction process

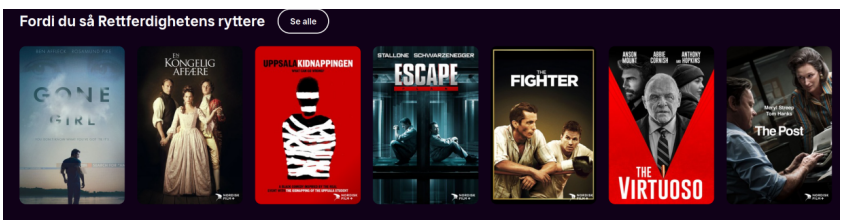


Figure 3.7: List of movies from visual trailer recommender on TV2 Play

Chapter 4

Results and Discussion

This chapter details the analysis conducted on the accumulated datasets for varioyus recommender models, and online tests. The chapter is categorized according to the various experiments conducted. Section 4.1, **Experiment A: Exploratory Analysis**, details the exploratory analysis of the *Trailer Features* and *Poster Features* datasets. Section 4.2, **Experiment B: Quality of recommendation**, details the offline evaluation using various recommendation algorithms in tandem with the *Trailer Features* and *Poster Features* datasets, and compares them with baseline pure CBF algorithms. Section 4.3, Experiment C: Cold Start simulation, details the metrics gathered by simulating the aforementioned algorithms in various cold start states. Section 4.4, **experiment D: Online Testing**, details the online tests performed on the digital streaming service TV2 Play.

4.1 Experiment A: Exploratory Data Analysis

The first set of experiments include an exploratory analysis of the two datasets, i.e., the dataset with Trailer Features and Poster Features, in order to get a better understanding of the data.

For visualization of the data various techniques were applied. For Figures 4.2,4.1, 4.8 and 4.7 a matrix of cosine similarities between all items were used.

For Figures 4.3, 4.4, 4.9, and 4.10 PCA was applied to the data-set, and clusters were determined using K-Means. The number of clusters were determined using the *Elbow* method. For Figure 4.2, 4.8 and Figure4.1 Cosine similarity has been utilized to find the similarity between the users.

4.1.1 Trailers

The histogram of cosine similarities show that the similarity between trailers follow a Bell curve, and that most movie trailers have a similarity score in the range of 0.75 to 0.95 as visualized in Figure 4.1.A similar result has been reported in prior work *Deldjoo et al.* (2016a). The cosine similarity matrix has also been visualized in Figure 4.2 as a heatmap along with an attached dendogram that show 6 hierarchical clusters that are present within the data.

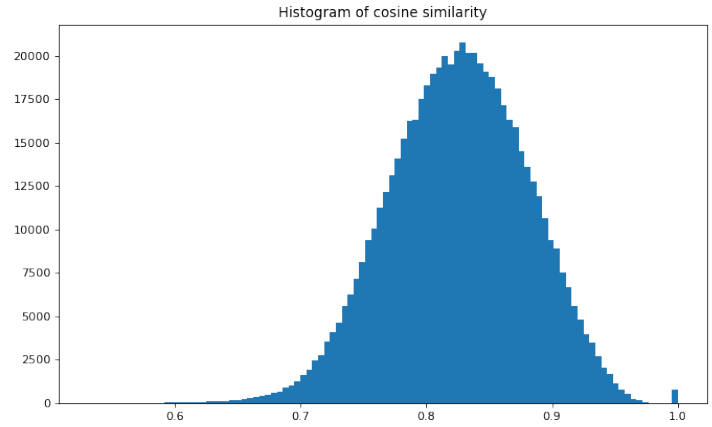


Figure 4.1: Histogram of cosine similarity on trailers

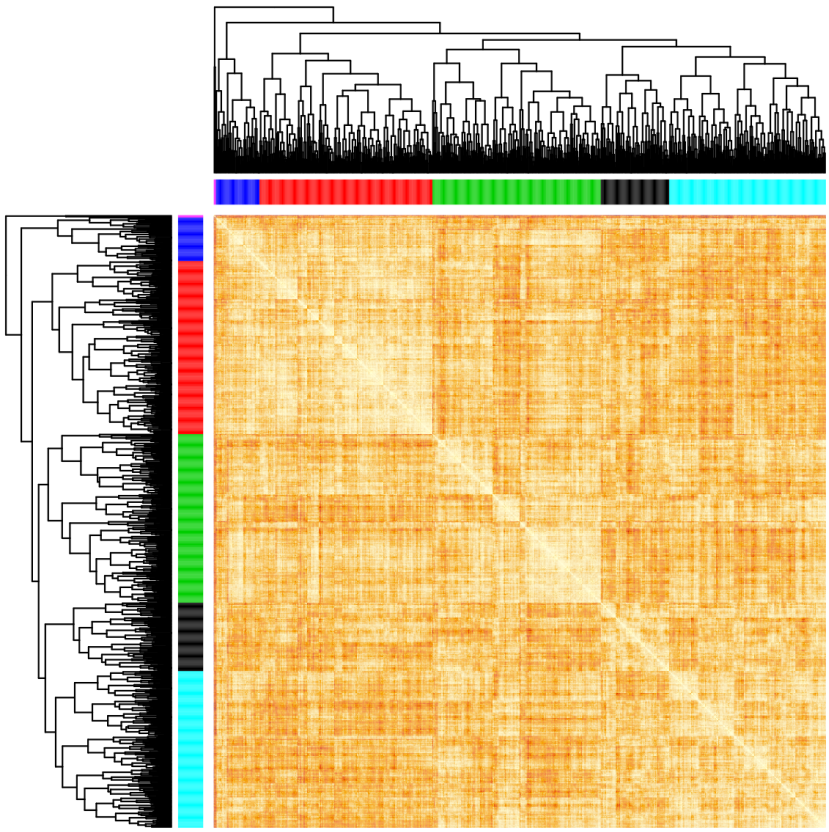


Figure 4.2: Cosine similarity matrix trailers based the extracted visual features

To cluster the data K-Means was performed using 11 clusters. The number of clusters was decided by inspecting the Elbow curve. As the data contained a vector of size 64 rendering it unplottable, PCA was performed to reduce the dimensionality of the data. Using a variance of 0.8, 10 principle components were used. Figure 4.3 visualizes the first and second principle component as these are the principle components with the largest variance.

There is clear seperation between most clusters as seen in Figure 4.3, with some some overlapping of clusters' 7, 10 and 0, and clusters' 8, 6 and 10. This is expected as

the similarity between most items is high, and we are only using the 1. and 2. principle component out of 10 components to visualize the data.

Visualizing principle component 1. and 3. (Figure 4.4 shows a similar plot, but from this perspective cluster 8 has mostly seperated from cluster's 6 and 10.

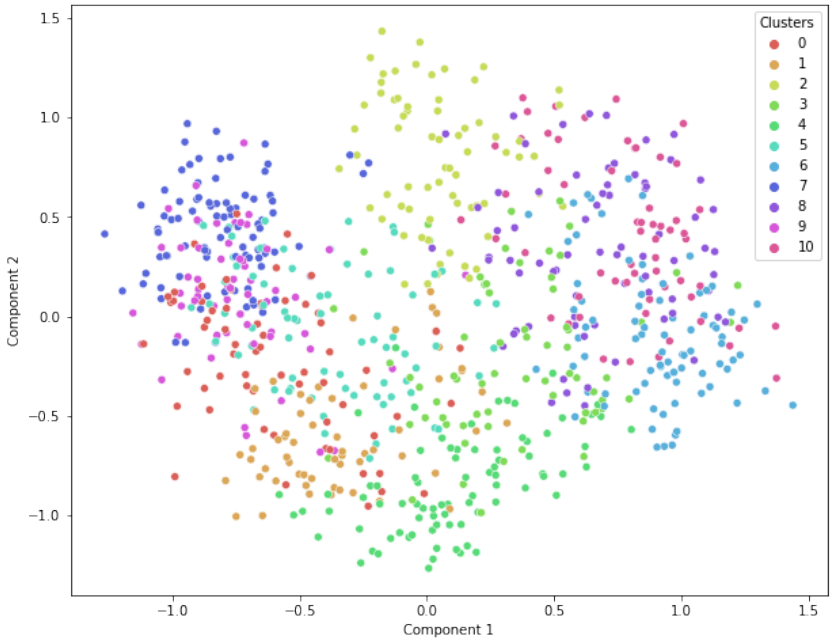


Figure 4.3: KMeans clustering movie trailers 2d plot using 1. and 2. principle component

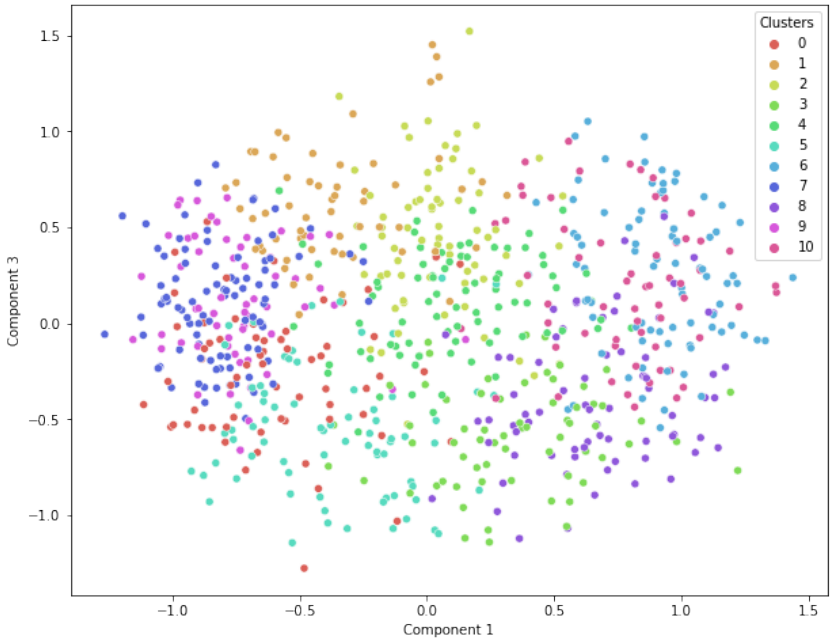


Figure 4.4: KMeans clustering movie trailers 3d plot using 1. and 3. principle component

Cluster	Movie	Year	Genre (IMDB)
Cluster0	Beck: Døden i Samarra	2021	Crime, Mystery, Thriller
	Charter	2020	Drama
	Olsenbanden jr. og det sorte gullet	2009	Family
Cluster1	Beck: Uten tvil	2020	Crime, Mystery, Thriller
	The Doorman	2020	Action, Drama, Thriller
	Mortal	2020	Action, Adventure, Drama
Cluster2	Bombshell	2019	Biography, Drama
	Tina and Bettina: The Movie	2012	Comedy
	Another Round	2020	Comedy, Drama
Cluster3	Fjolls til Fjells	2020	Comedy
	Despicable Me 3	2017	Animation, Adventure, Comedy
	Louis and Luca		
	- The Big Cheese Race	2015	Animation, Family
Cluster4	Knives Out	2019	Comedy, Crime, Drama
	All Inclusive	2017	Comedy, Drama
	Betrayed	2020	Drama, History, War
Cluster5	Utøya - July 22.	2018	Drama, Thriller
	Godzilla: King of the Monsters	2019	Action, Adventure, Fantasy
	The Big Short	2015	Biography, Comedy, Drama
Cluster6	1917	2019	Action, Drama, War
	Kamilla og Tyven	1988	Drama, Family
	Robin Hood	2018	Action, Adventure, Drama
Cluster7	Beck: Den Fortapte Sønn	2021	Crime, Mystery, Drama
	Greenland	2020	Action, Drama, Sci-fi
	The Ash Lad:		
	In the Hall of the Mountain King	2017	Adventure, Family, Fantasy
Cluster8	Beck: Undercover	2020	Crime, Mystery, Thriller
	Fireman Sam: Set for Action!	2018	Animation, Action, Adventure
	The Giant Pear	2017	Animation
Cluster9	Olsenbander Jr. Sølvgruvens Hemmelighet	2007	Family
	Apocalypto	2006	Action, Adventure, Drama
	Whitney Houston and Bobbi Kristina:		
	Didn't We Almost Have It All	2021	Documentary, Biography
Cluster10	The Crossing	2020	Adventure, Drama, Family
	Queen of Hearts	2019	Drama
	Norske Byggeklosser	2018	Comedy

Table 4.1: List of most popular movies within each cluster

In order to explore the K-Means clusters, the most popular movies from each cluster were extracted and presented in Table 4.1. The information gathered for these movies include the title, the year of release, and the genre tags from IMDB. Inspecting the clusters you can see there is a correlation between genres in the different clusters.

There are trailers present that intuitively should possibly be clustered together, such as the very popular Beck movies that appear in multiple clusters. Upon further inspection one can see that there is a difference in the visual look of these movies. This is exemplified in Figure 4.5 where keyframes from the beginning, middle, and end of the trailers for three of the Beck movies that appeared in different clusters are shown. In Figure 4.5 you can see that Beck - Døden i Samsarra has low brightness levels, and low entropy, where as Beck - Uten Tvil has higher brightness, more entropy, as well

as a very distinct green-ish color palette. The final example Beck - Undercover falls somewhere in between these two movies in terms of visual style.

Another example is looking at keyframes from movies within the same cluster. This has been exemplified in Figure 4.6, where keyframes from the beginning, middle, and end of Knives Out, All Inclusive, and Betrayed are presented. Here you can see that the trailers have a similar composition, as well as a similar color palette.

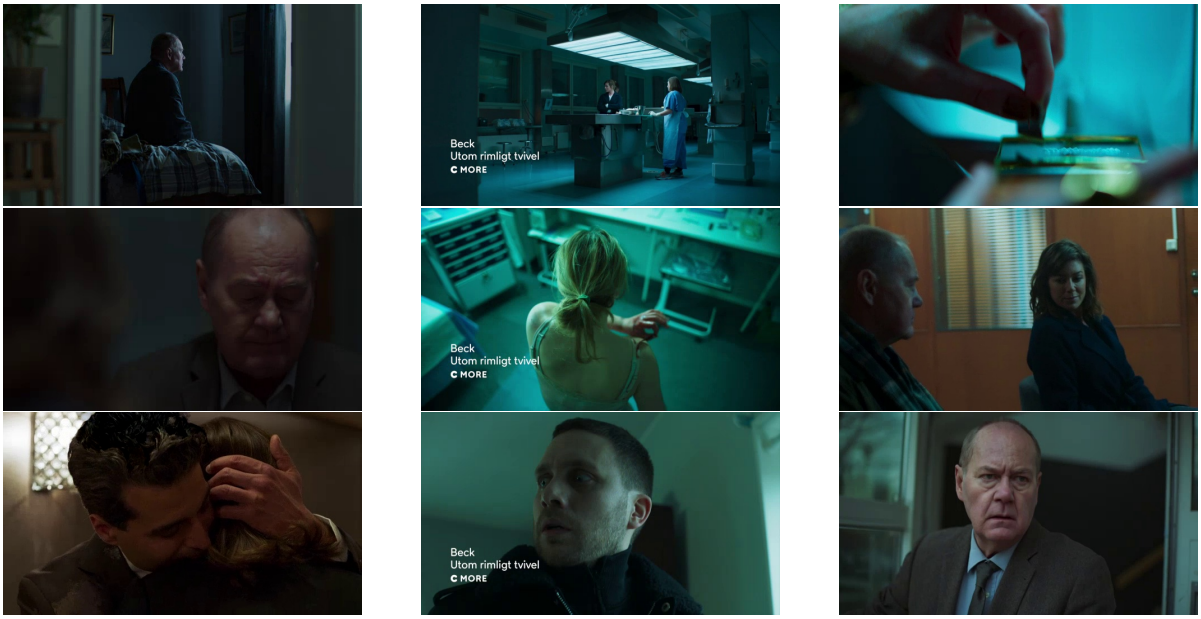


Figure 4.5: Keyframes extracted from Beck - Døden i Samarra, Beck - Uten Tvil, Beck - Undercover

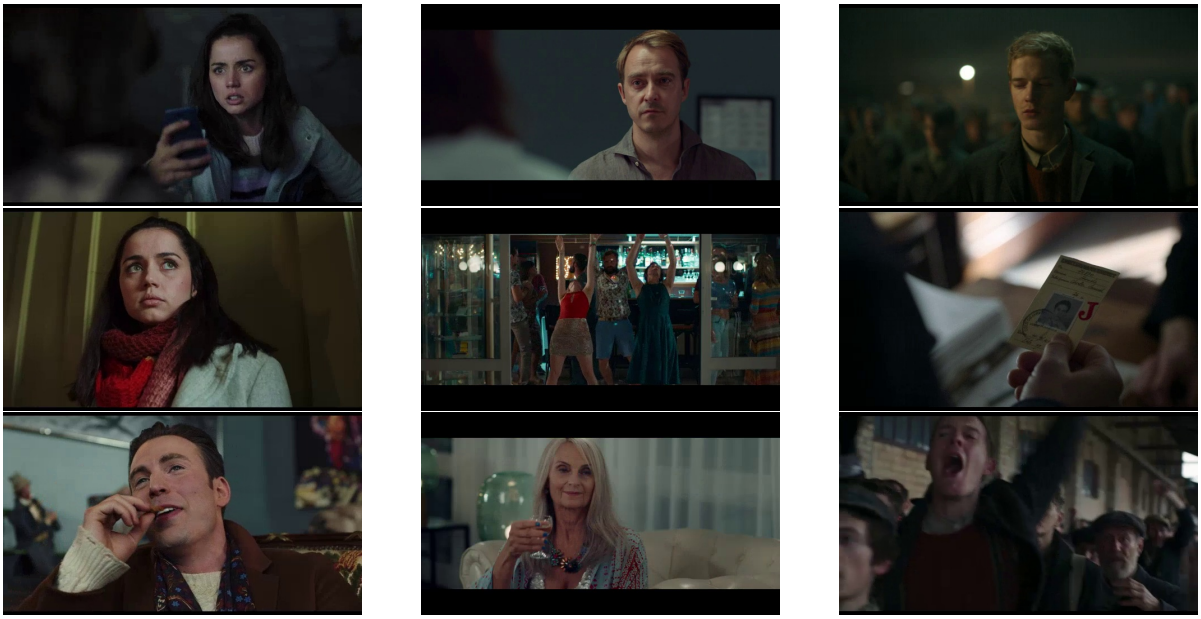


Figure 4.6: Cluster 4 examples. Keyframes from movie trailers Knives Out, All Inclusive, Betrayed, in that order left to right.

4.1.2 Posters

The histogram of cosine similarities in Figure 4.7 show that the similarities between posters do not follow a uniform Bell curve, but is instead shows a skewed distribution. It shows slightly top heavy with a peak at 0.83, and with a longer and more prominent tail than the trailers which stretches below 0.4.

The cosine similarity matrix has also been visualized in Figure 4.8 as a heatmap along with an attached dendogram that show 6 hierarchical clusters that are present within the data.

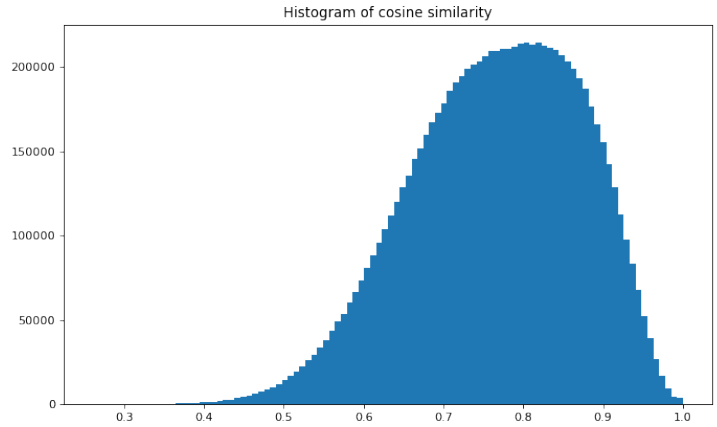


Figure 4.7: Histogram of cosine similarity on posters

To cluster the data K-Means was performed using 6 clusters. The number of clusters was decided by inspecting the Elbow curve. As the data contained a vector of size 15 rendering it unplottable, PCA was performed to reduce the dimensionality of the data. Using a variance of 0.8, 6 principle components were used. Figure 4.3 visualizes the first and second principle component as these are the principle components with the largest variance.

There is clear seperation between most clusters as seen in Figure 4.9, with some some overlapping of clusters' 4 and 5, and clusters' 2 and 3. Visualizing principle component 1. and 3. (Figure 4.10 shows a similar plot, but from this perspective cluster 8 has mostly seperated from cluster's 6 and 10.

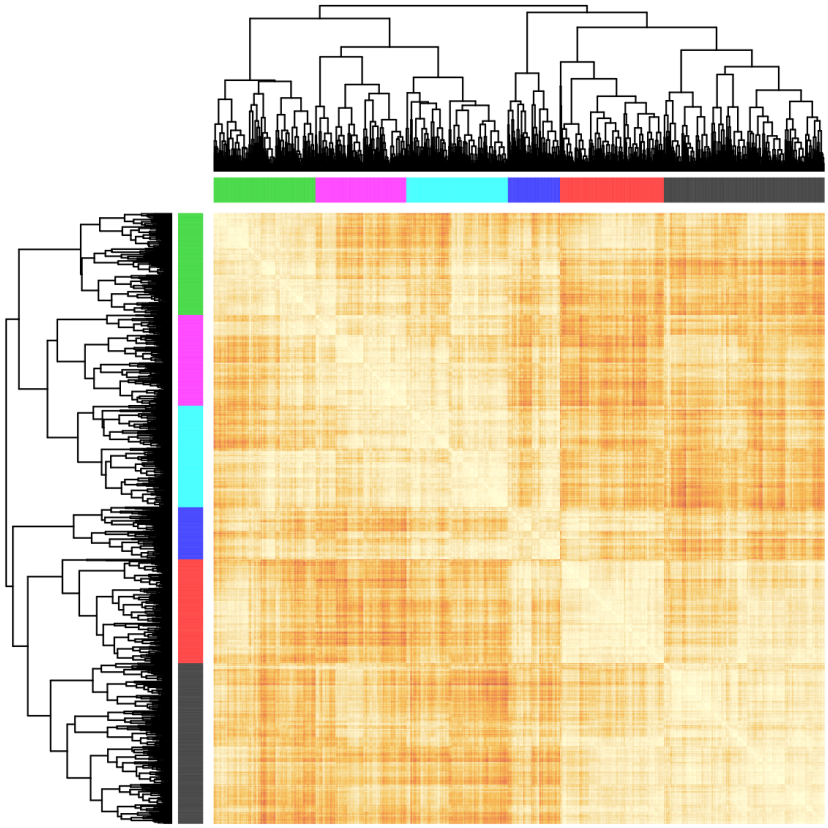


Figure 4.8: Cosine similarity matrix posters

Cluster	Movie	Year	IMDB(genre)
Cluster0	Karius og Baktus	1955	Animation, Short, Family
	Reveenka	1962	Short, Animation, Family
	Karsten og Petra på vinterferie	2014	Family
Cluster1	Beck: Døden i Samarra	2021	Crime, Mystery, Thriller
	The Gruffalo	2009	Animation, Family, Fantasy
	Charter	2020	Drama
Cluster2	The Ashlad and the Hungry Troll	1967	Animation, Short, Adventure
	Beck: Den Fortapte Sønn	2021	Crime, Mystery, Drama
	Fjolls til Fjells	2020	Comedy
Cluster3	Karsten og Petra på skattejakt	2018	Family
	Karsten og Petra lager teater	2017	Family
	The Ashlad and the Good Helpers	1961	Animation, Short, Adventure
Cluster4	Karsten og Petra på safari	2018	Family
	Kaptein Sabeltann og skatten i Lama Rama	2017	Action, Adventure, Comedy
	Long Flat Balls	2006	Comedy, Drama, Sport
Cluster5	Dyrene i Hakkebakkeskogen	2016	Animation, Family, Musical
	Kaptein Sabeltann og Grusomme Gabriels skatt	2005	Family
	Beck: Uten tvil	2020	Crime, Mystery, Thriller

Table 4.2: List of most popular movies within each cluster

As seen in Table 4.2 there seems to be lot of animation and family movies in the different clusters. This is expected as they are oftentimes the most watched movies on the platform. A more interesting find is that the Beck movies *Beck - Døden i Samarra*,

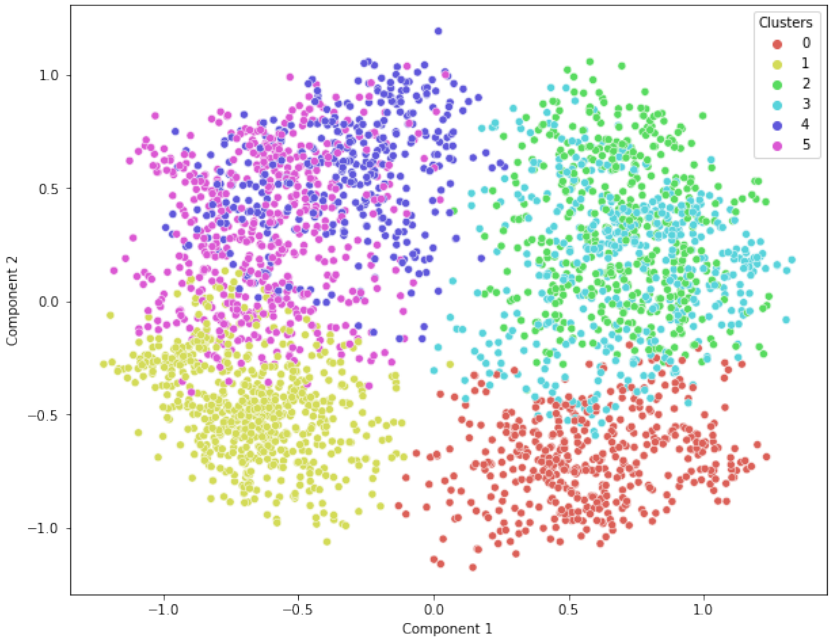


Figure 4.9: KMeans clusters movie posters 2d plot using 1. and 2. principle component

Beck - Den fortapte Sønn, and *Beck - Uten Tvil*, appear in different clusters even though the posters are aesthetically similar. Upon inspecting the features extracted from these posters it is apparent that the reason for their spread is the difference in color palette between the different posters. In future iterations of the project adding edge detection features for movie posters could be a possible method of finding similarities in the composition of movie posters.

Figure 4.11: Beck posters from different clusters

4.2 Experiment B: Quality of Recommendation

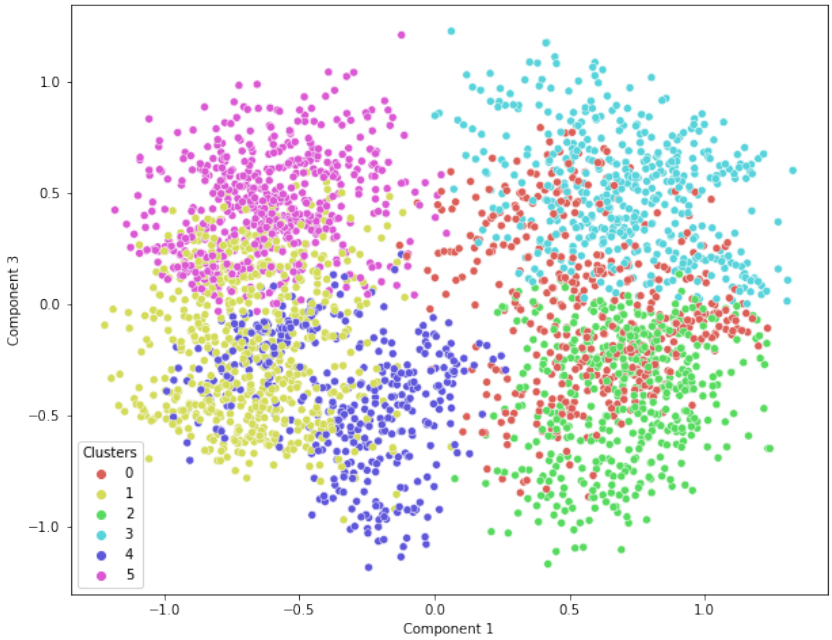


Figure 4.10: KMeans clusters movie posters 2d plot using 1. and 3. principle component

4.2.1 Trailers

In the second experiment, the quality of recommendations based on stylistic visual features extracted from movie trailers is being measured using various loss and accuracy metrics, as well as beyond accuracy metrics. The loss and accuracy results are presented in Table 4.3. The beyond accuracy metrics, catalog coverage and novelty, is represented in Figure 4.12 and 4.13

Evaluation Metrics	Recommenders				
	ALS*	BPR*	FM	WD	AINT
Log Loss	0.6205	0.4079	0.3605	0.3513	0.3496
Balanced accuracy	0.5899	0.8411	0.8419	0.8393	0.8413
ROC-AUC	0.8337	0.9241	0.9212	0.9167	0.9181
PR-AUC	0.8644	0.9091	0.9068	0.8968	0.8961
Precision@10	0.0498	0.0603	0.0557	0.0528	0.0535
Recall@10	0.2404	0.2956	0.2673	0.2626	0.2662
Map@10	0.1921	0.2192	0.1892	0.1719	0.1652
NDCG@10	0.2438	0.2801	0.2485	0.2313	0.2276

*Table 4.3: Loss and accuracy metrics for movie trailer recommendation. Algorithms marked with * are pure collaborative filtering algorithms used as a baseline.*

As seen in table 4.3 all recommendation algorithms using stylistic visual features outperform the pure CF baseline algorithms in terms of log loss, with AINT having the best score at 0.3496. In terms of balanced accuracy score FM achieved the best score of 0.8419, being tightly followed by AINT, BPR, and WD with a score of 0.8413, 0.8413, and 0.8413, respectively.

In terms of the ROC-AUC score, all algorithms using visual features outperform ALS, but BPR still has the highest score with 0.9241. The PR-AUC score follows a similar formula with the algorithms using visual features outperforming ALS, but BPR still has the highest score with 0.9091.

In terms of Precision@10 the best score 0.0603 was achieved by BPR, followed by 0.0557, 0.0535, 0.0528, 0.0498 achieved by FM, WD, AINT and ALS respectively. In terms of Recall@10 the best score is once again achieved by BPR at 0.2956, followed by FM, AINT, WD, and ALS with scores of 0.2673, 0.2662, 0.2626, 0.2404, respectively. In terms of Map@10 both ALS and BPR outperform the recommendation algorithms using stylistic visual features, with scores of 0.1921 and 0.2192, respectively. FM, WD, and AINT achieved Map@10 scores of 0.1892, 0.1719, 0.1652, respectively. In terms of NDCG@10 scores BPR outperforms the rest with a score of 0.2801, followed by FM, WD, AINT and ALS all scoring closely together with scores of 0.2485, 0.2313, 0.2276, and 0.2438, respectively.

The hybrid recommendation algorithms that utilize stylistic visual features FM, WD and AINT, routinely outperforms ALS on most loss and accuracy metrics, but BPR pulls slightly ahead.

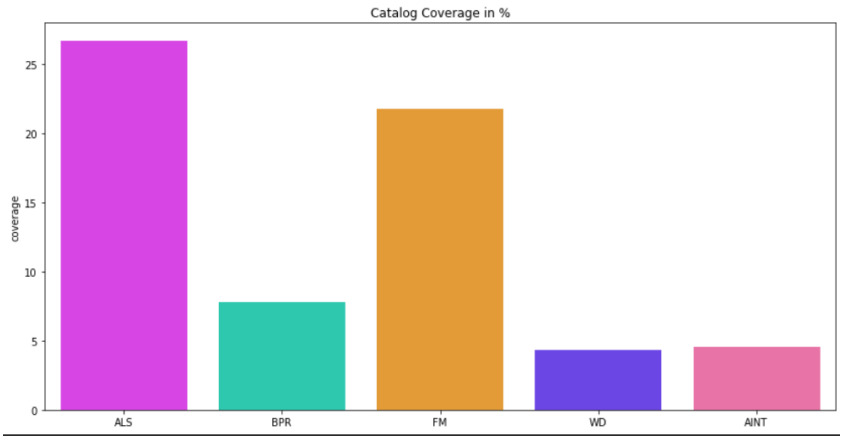


Figure 4.12: Coverage for movie trailers

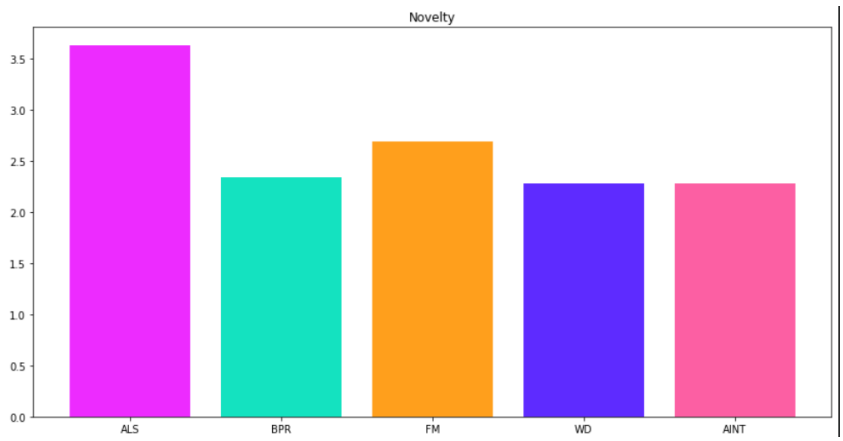


Figure 4.13: Novelty of recommendation for movie trailers

In terms of catalog coverage and novelty ALS performs the best, followed by FM. BPR, WD, and AINT all recommend very few movies, but they score well in terms of

loss and accuracy metrics as noted previously. FM utilizing stylistic visual features on the otherhand scored significantly better than ALS in terms of loss and accuracy metrics, and has good coverage and novelty. This implies that using stylistic visual features with a FM algorithm can improve upon accuracy over ALS, while still keeping good catalog coverage and novelty of recommendation, thus improving the overall quality of recommendation.

4.2.2 Posters

In the second experiment, the quality of recommendations based on stylistic visual features extracted from movie posters is described, based on various loss and accuracy metrics. The results are presented in Table 4.4.

Evaluation Metrics	Recommenders				
	ALS*	BPR*	FM	WD	AINT
Log Loss	0.5848	0.3474	0.2604	0.2926	0.2887
Balanced accuracy	0.6427	0.8365	0.8906	0.8726	0.8725
ROC-AUC	0.9416	0.9418	0.9576	0.9439	0.9456
PR-AUC	0.9464	0.9356	0.9516	0.9347	0.9367
Precision@10	0.0994	0.0812	0.0892	0.0543	0.0591
Recall@10	0.2001	0.1601	0.1744	0.1178	0.1198
Map@10	0.2499	0.1998	0.2252	0.1612	0.1576
NDCG@10	0.3276	0.2736	0.2985	0.2237	0.2218

*Table 4.4: Loss and accuracy metrics for movie posters. Algorithms marked with * are pure collaborative filtering algorithms used as a baseline.*

As seen in Table 4.4 all recommendation algorithms utilizing stylistic visual features outperform the baseline CF algorithms in terms of Log Loss, with FM having the best score of 0.2604, followed by AINT, WD, BPR, and ALS achieving scores of 0.2887, 0.2926, 0.3474 and 0.5848, respectively. In terms of Balanced Accuracy the hybrid algorithms still perform higher than the pure CF algorithms, with FM achieving a great score of 0.8906 followed by WD, AINT, BPR, and ALS with scores of 0.8726, 0.8725, 0.8365 and 0.6427, respectively.

The ROC-AUC scores are all close, with the recommendation algorithms utilizing stylistic visual features slightly outperforming the pure CF algorithms. FM achieved the best score with a score of 0.9576, followed by AINT, WD, BPR and ALS with scores of 0.9456, 0.9347, 0.9418, and 0.9416, respectively. PR-AUC scores follow a similar structure with FM achieving a great score of 0.9516, followed by ALS, BPR, WD, and AINT achieving scores of 0.9464, 0.9356, 0.9347, and 0.9367, respectively.

In terms of Precision@10 FM achieved a score of 0.0892 outperforming BPR's score of 0.0812, but is beat by ALS achieving the best score of 0.0994. WD and AINT lags behind the rest in terms of Precision@10 scores achieving scores of 0.0543 and 0.0591, respectively. In terms of Recall@10 ALS achieves the highest score of 0.2001, followed by FM, BPR, AINT, and WD achieving scores of 0.1744, 0.1601, 0.1198, and 0.1178, respectively. In terms of Map@10 scores ALS achieves the highest score

at 0.2499, followed by FM, BPR, WD, and AINT achieving scores of 0.2252, 0.1998, 0.1612, and 0.1576, respectively. Finally, in terms of NDCG@10 scores ALS once again comes out on top achieving a score of 0.3276, followed by FM, BPR, WD, and AINT achieving scores of 0.2985, 0.2736, 0.2237, and 0.2218, respectively.

In terms of catalog coverage the FM algorithm using stylistic visual features extracted from posters scored the highest . ALS is only slightly behind by 1 percent, while BPR, AINT and WD all have low coverage as seen in Figure 4.14. In terms of novelty ALS scored the best, followed by FM, AINT, BPR, and WD, respectively(Figure 4.15). Overall from all metrics we can infer that ALS outperforms all algorithms in terms of both loss and accuracy metrics, and beyond accuracy metrics. This also means that the effectiveness of using stylistic visual features for movie posters is slightly worse than pure collaborative filtering, and would possibly need different visual features than the trailers for more effective and higher quality recommendations.

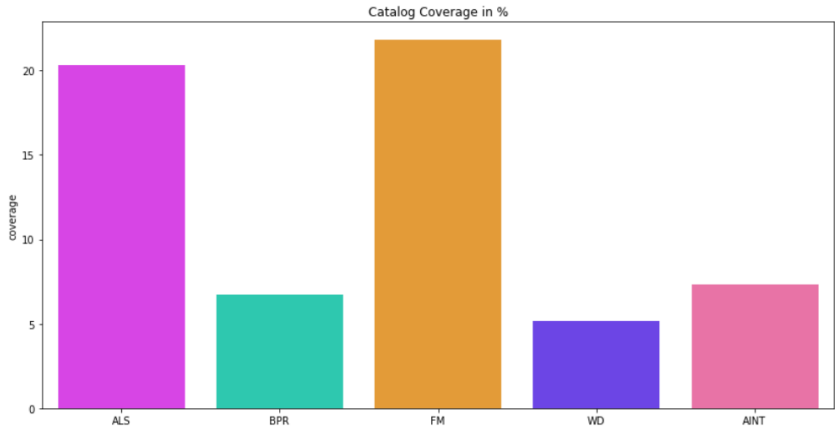


Figure 4.14: Catalog coverage for movie posters

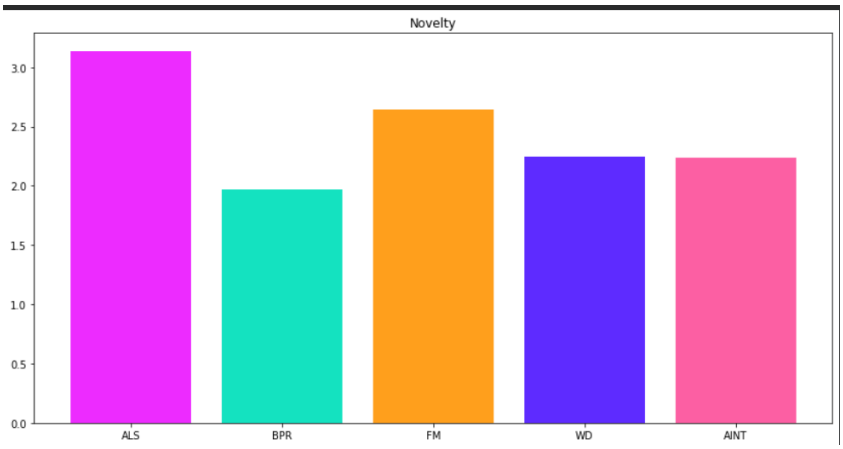


Figure 4.15: Novelty of recommendation for movie posters

4.3 Experiment C: Recommendation in Cold Start

This describes the experiments focused on simulations of the Cold Start scenario. The experiments have been conducted by random sampling of the percentage of datasets to

see how the different algorithms perform in various stages of cold start. For a complete set of loss and accuracy metrics for trailers and posters see Chapter 6 Appendix A.

4.3.1 Trailers

Evaluation Metrics	Percentage	Recommenders				
		ALS*	BPR*	FM	WD	AINT
Log Loss	10	0.6827	0.6893	0.6890	0.6665	0.6931
Balanced Accuracy	10	0.5795	0.7990	0.7609	0.7983	0.7969
ROC-AUC	10	0.5213	0.8807	0.8283	0.8811	0.8810
PR-AUC	10	0.6425	0.8930	0.8711	0.8924	0.8929
Precision@10	10	0.0148	0.0262	0.0197	0.0276	0.0276
Recall@10	10	0.1326	0.2293	0.1745	0.2419	0.2424
Map@10	10	0.0478	0.0835	0.0621	0.0897	0.0883
NDCG@10	10	0.0706	0.1220	0.0923	0.1299	0.1290
Log Loss	40	0.6585	0.5526	0.5241	0.3763	0.3742
Balanced Accuracy	40	0.5524	0.8252	0.8162	0.8244	0.8260
ROC-AUC	40	0.7101	0.9058	0.8946	0.9045	0.9058
PR-AUC	40	0.7756	0.8859	0.8722	0.8819	0.8835
Precision@10	40	0.0250	0.0388	0.0311	0.0373	0.0377
Recall@10	40	0.1745	0.2715	0.2172	0.2617	0.2651
Map@10	40	0.0873	0.1317	0.1066	0.1264	0.1264
NDCG@10	40	0.1223	0.1829	0.1492	0.1762	0.1770

Table 4.5: Cold start metrics for trailer recommendation using 10 and 40 percent of the original data

The cold start recommendation tests showed consistent performance from the algorithms hybrid using stylistic visual features. The hybrid algorithms consistently scored significantly better loss and accuracy metrics than the ALS algorithm in various stages of cold start. BPR outperformed the hybrid algorithms in terms of most of the loss and accuracy metrics, but was consistently beat by FM in terms of catalog coverage and novelty of recommendation in various stages of cold start as seen in Figures 4.16 and 4.18. The results are consistent in viewing the combination of stylistic visual features from trailers with an FM algorithm as a good middle ground, giving better loss and accuracy metrics than ALS, as well as better coverage and novelty of recommendation than BPR.

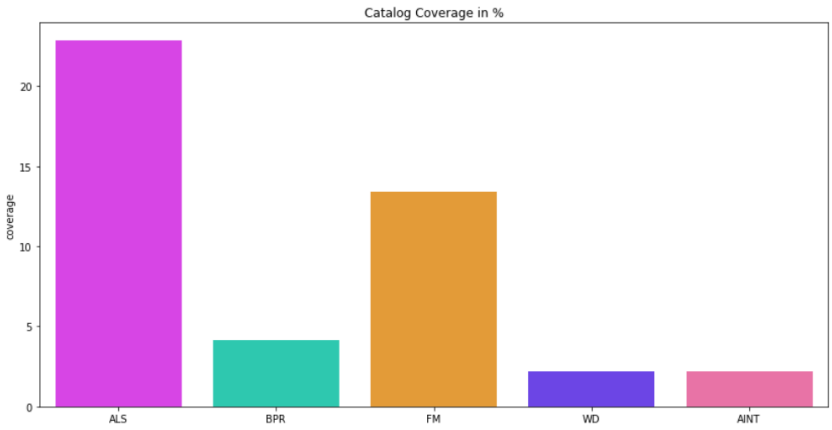


Figure 4.16: Catalog coverage of recommendation for movie posters using 10 percent of the original data

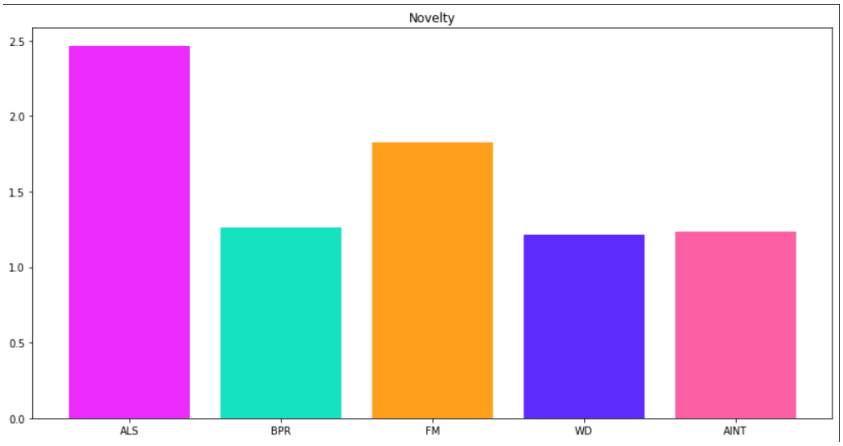


Figure 4.17: Novelty of recommendation for movie posters using 10 percent of the original data

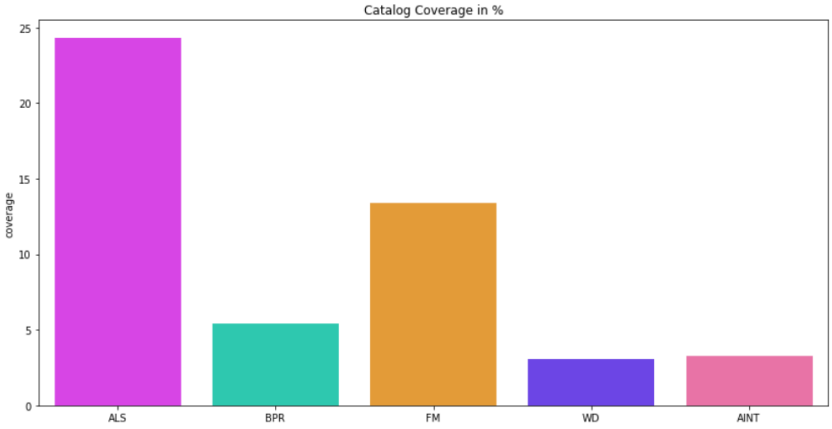


Figure 4.18: Catalog coverage of recommendation for movie posters using 40 percent of the original data

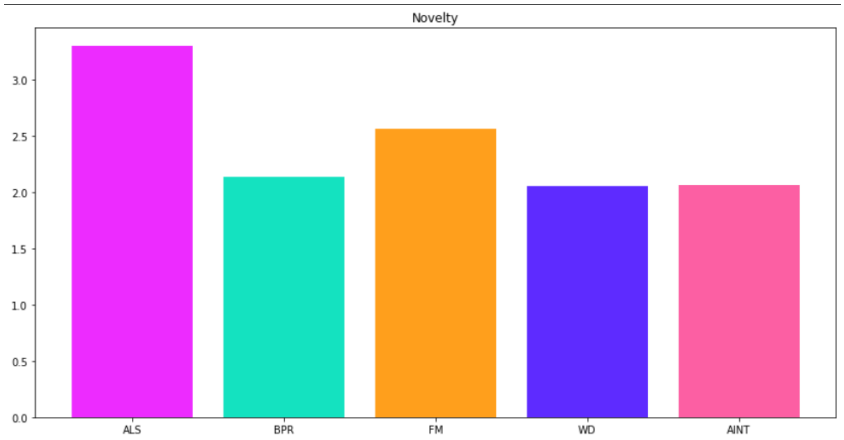


Figure 4.19: Novelty of recommendation for movie posters using 40 percent of the original data

4.3.2 Posters

Evaluation Metrics	Percentage	Recommenders				
		ALS*	BPR*	FM	WD	AINT
Log Loss	10	0.6690	0.6125	0.6049	0.4320	0.4299
Balanced Accuracy	10	0.5710	0.7889	0.7688	0.7873	0.7885
ROC-AUC	10	0.8708	0.9181	0.9208	0.9038	0.9118
PR-AUC	10	0.7720	0.8625	0.8380	0.8582	0.8588
Precision@10	10	0.0391	0.0339	0.0331	0.0218	0.0239
Recall@10	10	0.1059	0.1267	0.0938	0.1151	0.1152
Map@10	10	0.0484	0.0695	0.0456	0.0631	0.0634
NDCG@10	10	0.0708	0.0921	0.0651	0.0843	0.0846
Log Loss	40	0.6220	0.4247	0.3654	0.3765	0.3621
Balanced Accuracy	40	0.6150	0.8250	0.8362	0.8217	0.8309
ROC-AUC	40	0.8708	0.9181	0.9208	0.9038	0.9118
PR-AUC	40	0.8951	0.9082	0.9121	0.8878	0.8960
Precision@10	40	0.0391	0.0339	0.0331	0.0218	0.0239
Recall@10	40	0.1560	0.1324	0.1260	0.0850	0.1000
Map@10	40	0.1228	0.1082	0.1046	0.0757	0.0783
NDCG@10	40	0.1706	0.1494	0.1448	0.1024	0.1099

Table 4.6: Cold start metrics for poster recommendation using 10 and 40 percent of the original data

The cold start recommendation tests showed consistently better scores for the hybrid algorithms in various stages of cold start in terms of Log Loss, Balanced Accuracy, ROC-AUC and PR-AUC, whereas Precision@10 and Recall@10 scores were better for the pure CF algorithms. Catalog coverage for the FM also went significantly down at 10 percent of the original data as seen in 4.20, but managed to go back up again at around 40 percent of data as seen in Figure 4.22. Novelty did not change as much, but looking at Figure 4.21 and 4.21 it is visible that BPR and AINT had the biggest change in novelty going from 10 to 40 percent of the original data.

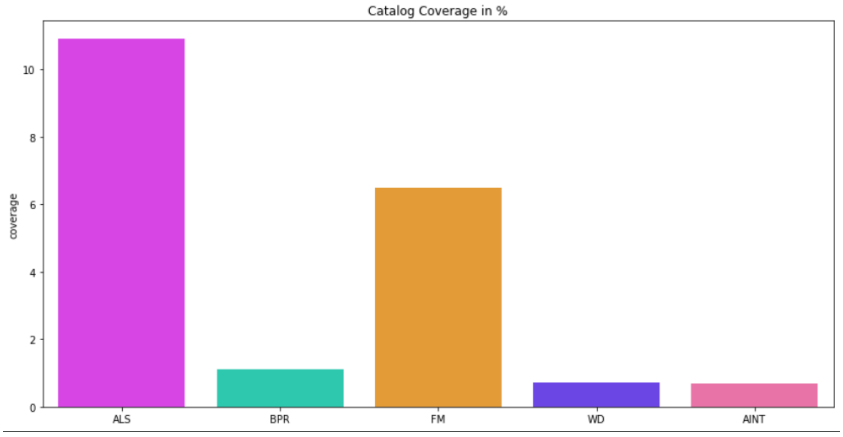


Figure 4.20: Catalog coverage of recommendation for movie posters using 10 percent of the original data

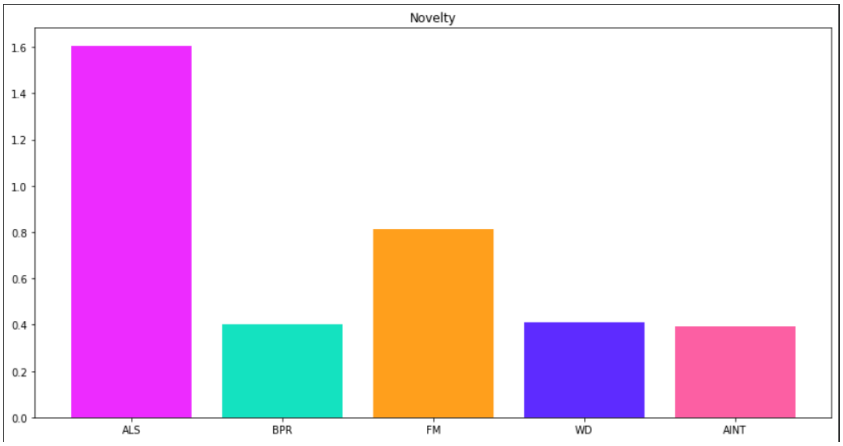


Figure 4.21: Novelty of recommendation for movie posters using 10 percent of the original data

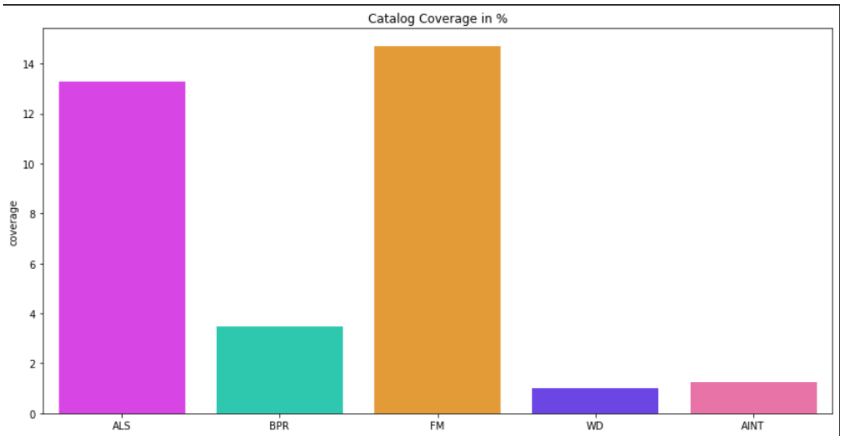


Figure 4.22: Catalog coverage of recommendation for movie posters using 40 percent of the original data

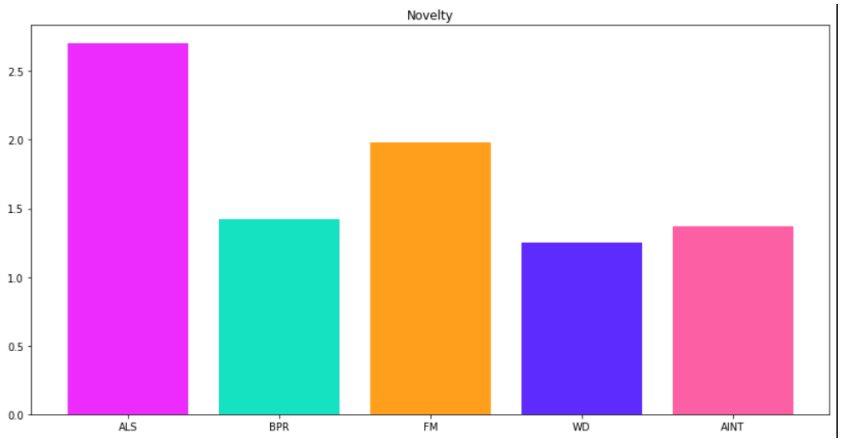


Figure 4.23: Novelty of recommendation for movie posters using 40 percent of the original data

4.4 Experiment D: Online evaluation

An online experiment has been conducted in order to evaluate the quality of content-based recommendation based on visual features in a *Warm start* condition, where the movies have already received a number of interactions from users (e.g., views or clicks). The online experiment has been run on TV2's online video streaming platform (TV2 Play) for 6 days, from the 23. to the 29. of May, 2022. The recommendation has been generated using the similarity scores between movies computed based on cosine similarity. Users were presented with recommendations lists of movies, most similar to the the movies users have watched before (see the example in Figure 4.24). The results of the experiment are presented in Figure 4.25, representing the number of views, clicks and click-through-rate,.

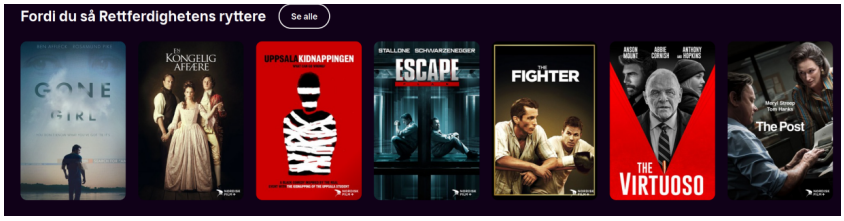


Figure 4.24: Screenshot from TV2 Play showing the kind of list a user would be presented with. The list title translated to English from Norwegian states: "Because you watched Riders of Justice:"

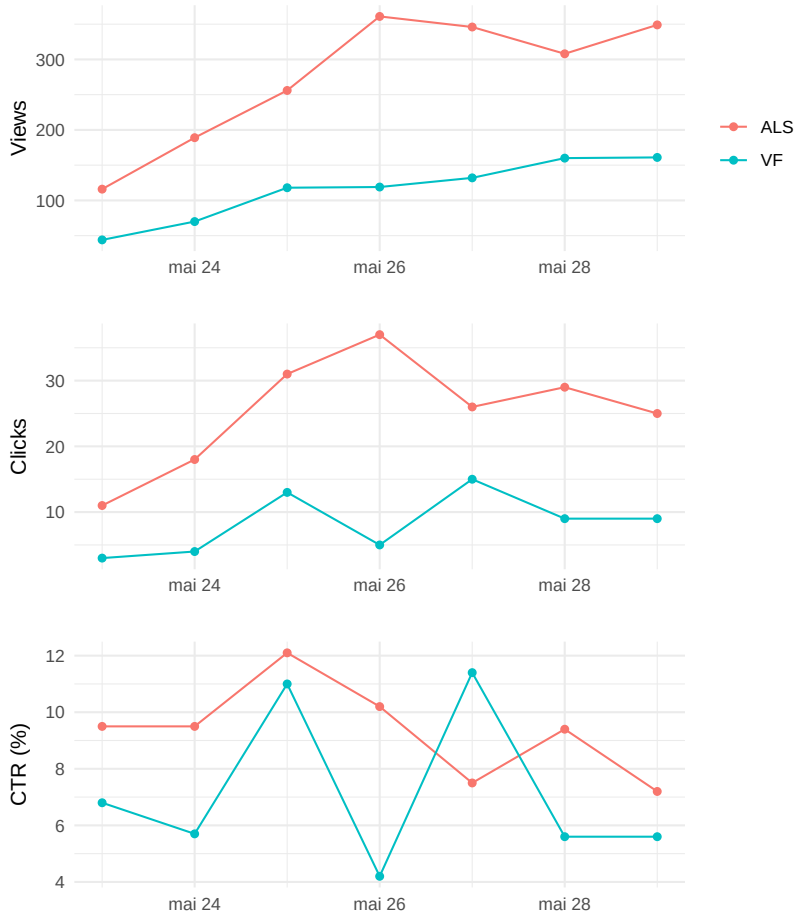


Figure 4.25: Plot from the five day online experiment run on TV2 Play showing Views, Clicks and CTR(click-through-rate). ALS refers to recommendations based on pure collaborative filtering, and VF refers to recommendations based on visual features.

Metric	CF (ALS)	CBF (Visual Features)
#Views	1925	804
#Clicks	177	58
#CTR	0.092	0.072

Table 4.7: Statistics from online experiment (Warm Start scenario)

It is important to note that the views for the recommendation lists in this experiment are quite low, but still give some insight into how the user behaves when presented with the recommendation lists. The reasoning for the low views may lie in their placement on the TV2 Play platform. A user needed to manually enter the Film category on the site, or app, before being presented with the recommendation lists. The recommendation list from visual features (i.e., VF) has also been placed as one of the last lists on the noted category page, thus leading to it having less overall views than the ALS recommender lists.

As expected, in the warm start scenario, where quality and quantity of the user data might be sufficient, the clicks for VF recommendation lists are lower than the ones

from ALS recommendation lists. This could also be due to how the ALS recommendation lists are personalized and based on user-similarity from a large collection of viewing sessions, whereas the VF recommendation lists only present the users with similar movies with a high cosine similarity score to one they have watched in the past.

The click-through-rate was also lower than ALS on most of the days, again, likely due to how ALS recommendations are personalized. Interestingly VF, despite having lower views and clicks, managed to outperform ALS in terms of CTR on the 27th of May. The statistical test (Fishers exact test) shows a p-value of 0.09 which is above 0.05, thus the null hypothesis cannot be rejected and we would need more data to determine whether or not ALS consistently outperforms VF in terms of CTR.

Despite having a lower number of clicks and a lower CTR scores (on most of the days), the recommendation based on visual features (VF) can still have the distinct advantage of being able to recommend movies to users without needing a watch history for the movie. Indeed, a simple form of recommendation based on visual similarities among movies can be a potential solution for the cold start scenario. However, utilizing a more complicated technique might result in improved quality of recommendation. Moreover, the proposed technique does not require any form of expensive editorial effort with manual human involvement. This means that incorporating visual features for recommendation process can still be beneficial to help in certain undesired situations in comparison to ALS, which may require weeks from the time movie is published on the platform till it starts getting recommended to users.

Chapter 5

Conclusions and Future Work

5.1 Summary

In this thesis both a novel recommendation technique based on automatically extracted visual features has been evaluated. The technique can be applied when using visual features individually with a content-based filtering process, or combined with other type of data with hybrid recommendation process. The technique can be used to mitigate the cold start problem. In addressing this problem, the thesis compared the quality of recommendation from pure collaborative filtering techniques with the quality of recommendation using a recommendation technique based on visual features with various algorithms using data from both movie trailers and posters.

The research was carried out by following a procedure which included:

- Conducting a literature review to determine the research context and state of the art (detailed in Chapter 2).
- Developing a novel feature extraction technique automatically extracting visual features from movie trailers and posters
  - Forming two novel datasets containing stylistic visual features from both movie trailers and movie posters (as detailed in Chapter 3.)
  - Conducted an offline evaluation of the datasets to compare the performance using these novel features in hybrid algorithms in comparison to collaborative filtering techniques (as detailed in Chapter 4)
- Develop a novel recommendation system using stylistic visual features and deploying it on a real-world digital streaming platform (TV2 Play) (as detailed in 3)
- Evaluate the performance of the proposed recommender system (as detailed in 4)

5.2 Main contributions

This thesis advances the state of the art of movie recommender systems through the following contributions:

- *Proposing a novel recommender system using stylistic visual features:* In Chapter 3, a content-based recommendation system is proposed. The proposed system was then deployed on a real-world digital streaming platform and A/B tested against a recommendation system based on collaborative filtering.
- *Proposing a hybrid based recommendation technique using stylistic features from movie trailers and movie posters* In Chapter 3 a hybrid based recommendation technique using stylistic visual features automatically extracted from movie posters and trailers was proposed.
- *A comprehensive evaluation of the proposed hybrid recommendation technique with offline and online experiments:* Chapter 3 proposed evaluation metrics to measure the performance of the hybrid based recommendation technique in comparison with pure collaborative-filtering techniques. In Chapter 4, a comprehensive evaluation with both offline and online experiments included an exploratory analysis, quality of recommendation and quality of recommendation in various stages of cold start scenarios as well as warm start scenario. In the exploratory analysis K-means clusters were identified and explored. PCA and cosine similarity was used to visualize the data showing item-similarity. In the quality of recommendation analysis multiple algorithms utilizing stylistic visual features were compared to collaborative-filtering algorithms using loss, accuracy and beyond accuracy metrics. In the cold start experiment the performance of the various implemented algorithms was tested in various stages of cold start. Finally, in an online evaluation, the quality of recommendation has been evaluated in a realistic scenario in one of the biggest movie streaming platforms.

5.3 Conclusion

The results of the offline evaluation of recommendation quality using features extracted from trailers detailed in Section 4.2 showed that models using stylistic visual features outperformed ALS on most metrics. The hybrid models were outperfomed by BPR in terms of PR-AUC, Precision@10, Recall@10, and Map@10 . But when taking the beyond accuracy metrics catalog coverage and novelty in to account we can conclude that using a hybrid model based on factorization machines (FM) provide the best overall performance in terms of both loss, accuracy and beyond accuracy metrics. As a conclusion of RQ. 1.1, the findings demonstrated that using stylistic visual features extracted from trailers used in a FM hybrid model provide overall better results in terms of quality of recommendation than the baseline collaborative-filtering techniques.

The results of the offline evaluation of recommendation quality using features extracted from posters detailed in Section 4.2 showed improved performance using stylistic visual features in terms of Log loss, balanced accuracy, and ROC-AUC metrics. But the overall top performing algorithm was ALS providing better Precision@10, Recall@10, Map@10, NDCG@10 as well as having more novel recommendations. Regarding RQ 1.1 the findings demonstrated that the stylistic features extracted from posters in were able to improve most loss, accuracy and beyond accuracy compared to BPR, but was beat by ALS in terms of Precision@10, Recall@10, and novelty of recommendation. Thus the answer to RQ 1.2 is inconclusive.

The results from the cold start experiment using features extracted from trailers detailed in Section 4.3 showed that using hybrid models consinstently outperformed ALS in terms of most loss and accuracy metrics, being slightly outperformed by BPR. Again, when taking beyond accuracy metrics catalog coverage and novelty in to account it becomes apparent that using FM with stylistic visual features serves as a sweet spot for great accuracy, and good catalog coverage and novelty of recommendation. In response to RQ 2.2 using stylistic visual features in a hybrid recommendation FM model showed increased overall performance in cold-start scenarios.

The results from the cold start experiment using features extracted from posters detailed in Section 4.3 showed that using hybrid models showed consistent increased performance in terms of log loss, balanced accuracy, ROC-AUC and PR-AUC over the collaborative filtering algorithms, but was outperformed by ALS in terms of Precision@10, Recall@10, Map@10 and NDCG@10. In an extreme cold start scenario using only 10 percent of the original data all hybrid recommendation algorithms were outperformed in terms of catalog coverage and novelty by ALS. But in a slightly warmer start using 40 percent of the original data using a hybrid model with a FM algorithm outperformed all algorithms in terms of catalog coverage. ALS was still the top performer in terms of novelty in all stages of cold start. Thus in response RQ 2.1 using stylistic visual features from movie posters for recommendation in cold start scenarios was not shown to significantly improve performance.

Regarding RQ3 the online evaluation did not provide enough data for the p-value from Fisher's exact test to say whether or not the performance difference in the A/B test was statistically significant. However, the content-based recommendation technique based on visual features can still have the distinct advantages of being able to recommend movies without needing expensive human annotation as well as being able to recommend movies without a watch history in the cold start scenario.

5.4 Limitations and future work

The research problems and approach of this thesis include limitations as well as possibilities for further research to be carried out.

Due to the recommendation library used it took multiple days extracting recommendations for all users across all stages of cold-start. This resulted in minimal tweaking of the recommendation algorithm parameters. Therefore, in further research the algorithm parameters should be tried with various modifications to see the impact of this on the quality of recommendation metrics proposed.

The number of trailers and posters which was extracted features from was also quite low. This was due to how not all movies on the TV2 Play platform had corresponding trailers and posters. Th

As stated in Chapter 4 the amount of interactions from users on the A/B tests were not enough to provide a statistically significant result in terms of how the users preferred recommendations coming from the novel content-based visual recommender, or the ALS collaborative-filtering recommender.

For further research on movie posters additional features such as edge detection could also be implemented to get a better understanding and knowledge of the compositional differences in movie posters.

The use of auditory features in combination with visual features is also a direction future works could go in. It is known that different genres of movies have different sound signatures, and these could be used for recommendation.

Another thing that has yet to be explored in the domain of visual movie recommendation is creating recommendations from frame to frame similarities.

Chapter 6

Appendix A: Cold Start Metrics

6.1 Metrics extracted from trailers

Evaluation Metrics	Percentage	Recommenders				
		ALS*	BPR*	FM	WD	AINT
Log-Loss	1	0.6903	0.6893	0.6890	0.6665	0.6931
Log-Loss	10	0.6827	0.6552	0.6585	0.4220	0.4270
Log-Loss	20	0.6728	0.6198	0.6177	0.4002	0.3993
Log-Loss	30	0.6641	0.5860	0.5714	0.3842	0.3832
Log-Loss	40	0.6585	0.5526	0.5241	0.3763	0.3742
Log-Loss	50	0.6480	0.5181	0.4777	0.3685	0.3674
Log-Loss	60	0.6417	0.4899	0.4371	0.3659	0.3638
Log-Loss	70	0.6364	0.4579	0.4081	0.3613	0.3621
Log-Loss	80	0.6285	0.4426	0.3898	0.3600	0.3581
Log-Loss	90	0.6237	0.4226	0.3746	0.3560	0.3565
Log-Loss	100	0.6205	0.4079	0.3605	0.3513	0.3496
Balanced-Accuracy	1	0.5115	0.7846	0.5846	0.7731	0.5000
Balanced-Accuracy	10	0.5795	0.7990	0.7609	0.7983	0.7969
Balanced-Accuracy	20	0.5388	0.8141	0.7988	0.8116	0.8141
Balanced-Accuracy	30	0.5493	0.8220	0.8090	0.8208	0.8207
Balanced-Accuracy	40	0.5524	0.8252	0.8162	0.8244	0.8260
Balanced-Accuracy	50	0.5673	0.8298	0.8211	0.8296	0.8295
Balanced-Accuracy	60	0.5720	0.8329	0.8247	0.8310	0.8343
Balanced-Accuracy	70	0.5766	0.8338	0.8297	0.8331	0.8332
Balanced-Accuracy	80	0.5871	0.8351	0.8341	0.8335	0.8351
Balanced-Accuracy	90	0.5875	0.8378	0.8363	0.8369	0.8375
Balanced-Accuracy	100	0.5899	0.8411	0.8419	0.8393	0.8413

Table 6.1: Log Loss and Balanced Accuracy metrics for movie trailers simulated cold start

Evaluation Metrics	Percentage	Recommenders				
		ALS*	BPR*	FM	WD	AINT
ROC-AUC	1	0.5028	0.8439	0.6021	0.8262	0.5605
ROC-AUC	10	0.5213	0.8807	0.8283	0.8811	0.8810
ROC-AUC	20	0.6425	0.8930	0.8711	0.8924	0.8929
ROC-AUC	30	0.6866	0.9020	0.8872	0.8999	0.9002
ROC-AUC	40	0.7101	0.9058	0.8946	0.9045	0.9058
ROC-AUC	50	0.7479	0.9108	0.9011	0.9085	0.9093
ROC-AUC	60	0.7709	0.9152	0.9055	0.9100	0.9115
ROC-AUC	70	0.7890	0.9143	0.9100	0.9119	0.9119
ROC-AUC	80	0.8133	0.9165	0.9129	0.9125	0.9141
ROC-AUC	90	0.8274	0.9202	0.9170	0.9144	0.9150
ROC-AUC	100	0.8337	0.9241	0.9212	0.9167	0.9181
PR-AUC	1	0.6020	0.8299	0.5765	0.8186	0.6069
PR-AUC	10	0.6758	0.8529	0.8079	0.8558	0.8534
PR-AUC	20	0.7272	0.8698	0.8441	0.8684	0.8683
PR-AUC	30	0.7586	0.8810	0.8628	0.8746	0.8745
PR-AUC	40	0.7756	0.8859	0.8722	0.8819	0.8835
PR-AUC	50	0.8040	0.8921	0.8811	0.8865	0.8869
PR-AUC	60	0.8198	0.8972	0.8868	0.8880	0.8891
PR-AUC	70	0.8327	0.8937	0.8913	0.8907	0.8899
PR-AUC	80	0.8505	0.8993	0.8952	0.8911	0.8931
PR-AUC	90	0.8605	0.9043	0.9019	0.8935	0.8930
PR-AUC	100	0.8644	0.9091	0.9068	0.8968	0.8961

Table 6.2: ROC-AUC and PR-AUC metrics for movie trailers simulated cold start

Evaluation Metrics	Percentage	Recommenders				
		ALS*	BPR*	FM	WD	AINT
Precision@10	1	0.0117	0.0141	0.0070	0.0180	0.0070
Precision@10	10	0.0148	0.0262	0.0197	0.0276	0.0276
Precision@10	20	0.0175	0.0296	0.0234	0.0297	0.0298
Precision@10	30	0.0219	0.0359	0.0264	0.0344	0.0346
Precision@10	40	0.0250	0.0388	0.0311	0.0373	0.0377
Precision@10	50	0.0326	0.0445	0.0346	0.0405	0.0405
Precision@10	60	0.0373	0.0489	0.0394	0.0432	0.0428
Precision@10	70	0.0402	0.0501	0.0439	0.0451	0.0452
Precision@10	80	0.0454	0.0536	0.0488	0.0493	0.0490
Precision@10	90	0.0476	0.0591	0.0530	0.0538	0.0536
Precision@10	100	0.0498	0.0603	0.0557	0.0528	0.0535
Recall@10	1	0.1172	0.1328	0.0703	0.1797	0.0703
Recall@10	10	0.1326	0.2293	0.1745	0.2419	0.2424
Recall@10	20	0.1363	0.2408	0.1879	0.2438	0.2445
Recall@10	30	0.1603	0.2625	0.1916	0.2524	0.2540
Recall@10	40	0.1745	0.2715	0.2172	0.2617	0.2651
Recall@10	50	0.2102	0.2821	0.2162	0.2581	0.2585
Recall@10	60	0.2185	0.2946	0.2375	0.2631	0.2620
Recall@10	70	0.2270	0.2713	0.2400	0.2460	0.2456
Recall@10	80	0.2461	0.2751	0.2513	0.2534	0.2525
Recall@10	90	0.2378	0.2934	0.2643	0.2661	0.2658
Recall@10	100	0.2404	0.2956	0.2673	0.2626	0.2662

Table 6.3: Precision@10 and Recall@10 metrics for movie trailers simulated cold start

Evaluation Metrics	Percentage	Recommenders				
		ALS*	BPR*	FM	WD	AINT
Map@10	1	0.0398	0.0548	0.0257	0.0999	0.0350
Map@10	10	0.0478	0.0835	0.0621	0.0897	0.0883
Map@10	20	0.0535	0.1018	0.0696	0.1099	0.1101
Map@10	30	0.0769	0.1212	0.0820	0.1164	0.1165
Map@10	40	0.0873	0.1317	0.1066	0.1264	0.1264
Map@10	50	0.1209	0.1573	0.1169	0.1393	0.1340
Map@10	60	0.1370	0.1715	0.1285	0.1408	0.1337
Map@10	70	0.1483	0.1615	0.1400	0.1432	0.1439
Map@10	80	0.1712	0.1839	0.1557	0.1609	0.1634
Map@10	90	0.1718	0.2094	0.1822	0.1772	0.1768
Map@10	100	0.1921	0.2192	0.1892	0.1719	0.1652
NDCG@10	1	0.0578	0.0727	0.0358	0.1189	0.0433
NDCG@10	10	0.0706	0.1220	0.0923	0.1299	0.1290
NDCG@10	20	0.0798	0.1439	0.1051	0.1505	0.1508
NDCG@10	30	0.1074	0.1700	0.1205	0.1628	0.1634
NDCG@10	40	0.1223	0.1829	0.1492	0.1762	0.1770
NDCG@10	50	0.1632	0.2133	0.1627	0.1911	0.1873
NDCG@10	60	0.1810	0.2296	0.1789	0.1954	0.1895
NDCG@10	70	0.1964	0.2204	0.1947	0.1974	0.1980
NDCG@10	80	0.2235	0.2423	0.2137	0.2170	0.2190
NDCG@10	90	0.2252	0.2741	0.2427	0.2387	0.2384
NDCG@10	100	0.2438	0.2801	0.2485	0.2313	0.2276

Table 6.4: Map@10 and NDCG@10 metrics for movie trailers simulated cold start

6.2 Metrics extracted from posters

Evaluation Metrics	Percentage	Recommenders				
		ALS*	BPR*	FM	WD	AINT
Log-Loss	1	0.6927	0.6849	0.6881	0.5215	0.6931
Log-Loss	10	0.6690	0.6125	0.6049	0.4320	0.4299
Log-Loss	20	0.6483	0.5302	0.4984	0.4030	0.3999
Log-Loss	30	0.6327	0.4686	0.4101	0.3884	0.3818
Log-Loss	40	0.6220	0.4247	0.3654	0.3765	0.3621
Log-Loss	50	0.6130	0.3978	0.3325	0.3616	0.3466
Log-Loss	60	0.6045	0.3769	0.3094	0.3469	0.3265
Log-Loss	70	0.5975	0.3684	0.2945	0.3303	0.3176
Log-Loss	80	0.5927	0.3553	0.2801	0.3106	0.3072
Log-Loss	90	0.5897	0.3508	0.2693	0.2994	0.2966
Log-Loss	100	0.5848	0.3474	0.2604	0.2926	0.2887
Balanced-Accuracy	1	0.4938	0.7403	0.5812	0.7407	0.5000
Balanced-Accuracy	10	0.5710	0.7889	0.7688	0.7873	0.7885
Balanced-Accuracy	20	0.5988	0.8112	0.7933	0.8048	0.8099
Balanced-Accuracy	30	0.6103	0.8204	0.8190	0.8137	0.8204
Balanced-Accuracy	40	0.6150	0.8250	0.8362	0.8217	0.8309
Balanced-Accuracy	50	0.6270	0.8262	0.8517	0.8318	0.8387
Balanced-Accuracy	60	0.6334	0.8357	0.8645	0.8407	0.8521
Balanced-Accuracy	70	0.6386	0.8287	0.8727	0.8468	0.8569
Balanced-Accuracy	80	0.6436	0.8336	0.8810	0.8623	0.8636
Balanced-Accuracy	90	0.6392	0.8347	0.8861	0.8689	0.8710
Balanced-Accuracy	100	0.6427	0.8365	0.8906	0.8726	0.8725

Table 6.5: Log loss and Balanced Accuracy metrics for movie posters in a simulated cold start environment

Evaluation Metrics	Percentage	Recommenders				
		ALS*	BPR*	FM	WD	AINT
ROC-AUC	1	0.4917	0.8211	0.6101	0.8210	0.7364
ROC-AUC	10	0.7025	0.8766	0.8498	0.8746	0.8758
ROC-AUC	20	0.7959	0.8979	0.8829	0.8903	0.8926
ROC-AUC	30	0.8453	0.9082	0.9071	0.8981	0.9018
ROC-AUC	40	0.8708	0.9181	0.9208	0.9038	0.9118
ROC-AUC	50	0.8892	0.9217	0.9327	0.9106	0.9191
ROC-AUC	60	0.9098	0.9334	0.9411	0.9180	0.9294
ROC-AUC	70	0.9217	0.9303	0.9462	0.9276	0.9333
ROC-AUC	80	0.9299	0.9388	0.9512	0.9366	0.9378
ROC-AUC	90	0.9338	0.9395	0.9547	0.9413	0.9419
ROC-AUC	100	0.9416	0.9418	0.9576	0.9439	0.9456
PR-AUC	1	0.6020	0.8138	0.6214	0.8155	0.7560
PR-AUC	10	0.7720	0.8625	0.8380	0.8582	0.8588
PR-AUC	20	0.8426	0.8857	0.8713	0.8738	0.8750
PR-AUC	30	0.8771	0.8972	0.8983	0.8817	0.8837
PR-AUC	40	0.8951	0.9082	0.9121	0.8878	0.8960
PR-AUC	50	0.9091	0.9122	0.9249	0.8951	0.9035
PR-AUC	60	0.9228	0.9254	0.9338	0.9047	0.9176
PR-AUC	70	0.9315	0.9222	0.9391	0.9169	0.9217
PR-AUC	80	0.9378	0.9322	0.9442	0.9267	0.9267
PR-AUC	90	0.9406	0.9331	0.9483	0.9317	0.9314
PR-AUC	100	0.9464	0.9356	0.9516	0.9347	0.9367

Table 6.6: ROC-AUC and PR-AUC metrics for movie posters in a simulated cold start environment

Evaluation Metrics	Percentage	Recommenders				
		ALS*	BPR*	FM	WD	AINT
Precision@10	1	0.0076	0.0160	0.0054	0.0157	0.0145
Precision@10	10	0.0150	0.0178	0.0135	0.0164	0.0165
Precision@10	20	0.0219	0.0243	0.0197	0.0212	0.0206
Precision@10	30	0.0310	0.0277	0.0289	0.0212	0.0206
Precision@10	40	0.0391	0.0339	0.0331	0.0218	0.0239
Precision@10	50	0.0480	0.0391	0.0417	0.0249	0.0264
Precision@10	60	0.0588	0.0508	0.0485	0.0330	0.0375
Precision@10	70	0.0714	0.0540	0.0623	0.0417	0.0419
Precision@10	80	0.0769	0.0649	0.0679	0.0455	0.0439
Precision@10	90	0.0900	0.0735	0.0824	0.0529	0.0508
Precision@10	100	0.0994	0.0812	0.0892	0.0543	0.0591
Recall@10	1	0.0738	0.1530	0.0508	0.1513	0.1406
Recall@10	10	0.1059	0.1267	0.0938	0.1151	0.1152
Recall@10	20	0.1208	0.1364	0.1083	0.1195	0.1169
Recall@10	30	0.1390	0.1286	0.1348	0.1010	0.1001
Recall@10	40	0.1560	0.1324	0.1260	0.0850	0.1000
Recall@10	50	0.1672	0.1369	0.1405	0.0862	0.0945
Recall@10	60	0.1815	0.1524	0.1398	0.1046	0.1117
Recall@10	70	0.1905	0.1517	0.1659	0.1217	0.1256
Recall@10	80	0.1910	0.1567	0.1623	0.1169	0.1129
Recall@10	90	0.1949	0.1609	0.1753	0.1216	0.1149
Recall@10	100	0.2001	0.1601	0.1744	0.1178	0.1198

Table 6.7: Precision@10 and Recall@10 metrics for movie posters in a simulated cold start environment

Evaluation Metrics	Percentage	Recommenders				
		ALS*	BPR*	FM	WD	AINT
Map@10	1	0.0288	0.0613	0.0147	0.0668	0.0638
Map@10	10	0.0484	0.0695	0.0456	0.0631	0.0634
Map@10	20	0.0785	0.0857	0.0724	0.0775	0.0767
Map@10	30	0.0954	0.0986	0.0969	0.0775	0.0756
Map@10	40	0.1228	0.1082	0.1046	0.0757	0.0783
Map@10	50	0.1396	0.1185	0.1259	0.0792	0.0835
Map@10	60	0.1618	0.1499	0.1555	0.1021	0.1180
Map@10	70	0.1914	0.1584	0.1696	0.1273	0.1262
Map@10	80	0.2023	0.1896	0.1883	0.1437	0.1374
Map@10	90	0.2234	0.2033	0.2173	0.1580	0.1504
Map@10	100	0.2499	0.1998	0.2252	0.1612	0.1576
NDCG@10	1	0.0396	0.0836	0.0237	0.0879	0.0826
NDCG@10	10	0.0708	0.0921	0.0651	0.0843	0.0846
NDCG@10	20	0.1074	0.1165	0.0975	0.1043	0.1026
NDCG@10	30	0.1344	0.1320	0.1330	0.1037	0.1017
NDCG@10	40	0.1706	0.1494	0.1448	0.1024	0.1099
NDCG@10	50	0.1972	0.1668	0.1748	0.1110	0.1180
NDCG@10	60	0.2299	0.2065	0.2083	0.1449	0.1635
NDCG@10	70	0.2638	0.2207	0.2345	0.1787	0.1782
NDCG@10	80	0.2775	0.2549	0.2543	0.1964	0.1879
NDCG@10	90	0.3045	0.2756	0.2898	0.2193	0.2095
NDCG@10	100	0.3276	0.2736	0.2985	0.2237	0.2218

Table 6.8: Map@10 and NDCG@10 metrics for movie posters in a simulated cold start environment

Bibliography

Beheshti, A., S. Ghodratnama, M. Elahi, and H. Farhood (2022), *Social Data Analytics*, CRC Press. 1.1

Brodersen, K. H., C. S. Ong, K. E. Stephan, and J. M. Buhmann (2010), The balanced accuracy and its posterior distribution, in *2010 20th international conference on pattern recognition*, pp. 3121–3124, IEEE. 3.6.1

Burke, R. (2000), Knowledge-based recommender systems, *Encyclopedia of library and information systems*, *69*(Supplement 32), 175–186. 2.1

Cheng, H.-T., L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, et al. (2016), Wide & deep learning for recommender systems, in *Proceedings of the 1st workshop on deep learning for recommender systems*, pp. 7–10. 3.4

Deldjoo, Y., M. Elahi, P. Cremonesi, F. Garzotto, P. Piazzolla, and M. Quadrana (2016a), Content-based video recommendation system based on stylistic visual features, *Journal on Data Semantics*, *5*, 99–113, doi:10.1007/S13740-016-0060-9. 1.1, 2.2, 2.3, 4.1.1

Deldjoo, Y., M. Elahi, P. Cremonesi, F. Garzotto, and P. Piazzolla (2016b), Recommending movies based on mise-en-scene design, in *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pp. 1540–1547. 2.1

Deldjoo, Y., M. Elahi, M. Quadrana, and P. Cremonesi (2018), Using visual features based on MPEG-7 and deep learning for movie recommendation, *International Journal of Multimedia Information Retrieval*, *7*(4), 207–219, doi:10.1007/S13735-018-0155-1. 2.1

Elahi, F. B. M. M., et al. (2019), Cold start solutions for recommendation systems. 1.1, 2.1, 2.2

Elahi, M., M. Braunhofer, T. Gurbanov, and F. Ricci (2018), User preference elicitation, rating sparsity and cold start. 2.2

Elahi, M., R. Hosseini, M. H. Rimaz, F. B. Moghaddam, and C. Trattner (2020), Visually-aware video recommendation in the cold start, in *Proceedings of the 31st ACM Conference on Hypertext and Social Media*, pp. 225–229. 1.2

Elahi, M., D. Jannach, L. Skjærven, E. Knudsen, H. Sjøvaag, K. Tolonen, Ø. Holmstad, I. Pipkin, E. Throndsen, A. Stenbom, et al. (2021a), Towards responsible media recommendation, *AI and Ethics*, pp. 1–12. 1.2

Elahi, M., F. Bakhshandegan Moghaddam, R. Hosseini, M. H. Rimaz, N. El Ioini, M. Tkalcic, C. Trattner, and T. Tillo (2021b), Recommending videos in cold start with automatic visual tags, in *Adjunct Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization*, pp. 54–60. 1.2

Ge, M., C. Delgado-Battenfeld, and D. Jannach (2010), Beyond accuracy: evaluating recommender systems by coverage and serendipity, in *Proceedings of the fourth ACM conference on Recommender systems*, pp. 257–260. 3.6.1

Hasler, D., and S. E. Suesstrunk (2003), Measuring colorfulness in natural images, in *Human vision and electronic imaging VIII*, vol. 5007, pp. 87–95, International Society for Optics and Photonics. 3.1.2

Hastie, T., R. Tibshirani, and J. Friedman (2001), The elements of statistical learning. springer series in statistics, *New York, NY, USA*. 3.6.1

Hazrati, N., and M. Elahi (2021), Addressing the new item problem in video recommender systems by incorporation of visual features with restricted boltzmann machines, *Expert Systems*, *38*(3), e12,645. 1.1, 2.1

Hazrati, N., M. Elahi, and F. Ricci (2020), Simulating the impact of recommender systems on the evolution of collective users' choices, in *Proceedings of the 31st ACM conference on hypertext and social media*, pp. 207–212. 2.1

Hu, Y., Y. Koren, and C. Volinsky (2008), Collaborative Filtering for Implicit Feedback Datasets. 2.1, 3.4

Kvifte, T., M. Elahi, and C. Trattner (2021), Hybrid recommendation of movies based on deep content features. 1.1, 1.2

Lops, P., M. d. Gemmis, and G. Semeraro (2011), Content-based recommender systems: State of the art and trends, *Recommender systems handbook*, pp. 73–105. 3.4.1

Lü, L., M. Medo, C. H. Yeung, Y.-C. Zhang, Z.-K. Zhang, and T. Zhou (2012), Recommender Systems. 3.6.1

Moghaddam, F. B., M. Elahi, R. Hosseini, C. Trattner, and M. Tkalcic (2019), Predicting Movie Popularity and Ratings with Visual Features, *2019 14th International Workshop on Semantic and Social Media Adaptation and Personalization, SMAP 2019*, doi:10.1109/SMAP.2019.8864912. 2.2

Moosburger, M. (2017), Colour labelling of art images using colour palette recognition, *Bachelorarbeit/bachelor thesis*. 3.1.2

Rendle, S. (2010), Factorization machines, in *2010 IEEE International conference on data mining*, pp. 995–1000, IEEE. 3.4

Rendle, S., C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme (2012), Bpr: Bayesian personalized ranking from implicit feedback, *arXiv preprint arXiv:1205.2618*. 3.4

Rimaz, M. H., M. Elahi, F. Bakhshandegan Moghadam, C. Trattner, R. Hosseini, and M. Tkalčič (2019), Exploring the power of visual features for the recommendation of movies, in *Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization*, pp. 303–308. 2.2, 2.3

Schedl, M., H. Zamani, C. W. Chen, Y. Deldjoo, and M. Elahi (2018), Current challenges and visions in music recommender systems research, *International Journal of Multimedia Information Retrieval*, *7*(2), 95–116, doi:10.1007/S13735-018-0154-2. 3.6.1

Song, W., C. Shi, Z. Xiao, Z. Duan, Y. Xu, M. Zhang, and J. Tang (2019), Autoint: Automatic feature interaction learning via self-attentive neural networks, in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 1161–1170. 3.4

Van Meteren, R., and M. Van Someren (2000), Using content-based filtering for recommendation, in *Proceedings of the machine learning in the new information age: MLnet/ECML2000 workshop*, vol. 30, pp. 47–56. 2.1

Vovk, V. (2015), The fundamental nature of the log loss function, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9300*, 307–318, doi:10.1007/978-3-319-23534-9_20. 3.6.1

Weisstein, E. W. (), Fisher's Exact Test – from Wolfram MathWorld. 3.6.2

Zhao, L., Z. Lu, S. J. Pan, Q. Yang, and W. Xu (2016), Matrix factorization+ for movie recommendation., in *IJCAI*, pp. 3945–3951. 1.1, 2.2, 2.3

Zhou, T., L. Lü, and Y. C. Zhang (2009), Predicting missing links via local information, *European Physical Journal B*, *71*(4), 623–630, doi:10.1140/EPJB/E2009-00335-8. 3.6.1

Zhou, T., Z. Kuscsik, J.-G. Liu, M. Medo, J. R. Wakeling, and Y.-C. Zhang (2010), Solving the apparent diversity-accuracy dilemma of recommender systems, *Proceedings of the National Academy of Sciences*, *107*(10), 4511–4515. 3.6.1