Recommendation Systems of Short Video Platforms:

Auditing Algorithms of Short Format Video Platforms to Understand the Rabbit Hole Effect on YouTube Shorts

SEN2331: CoSEM Master Thesis

Christophe Cosse

Delft University of Technology

TU Delft

Recommendation Systems of Short Video Platforms:

Auditing Algorithms of Short Format Video Platforms to Understand the Rabbit Hole Effect on YouTube Shorts

by

Christophe Cosse

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Thursday October 24, 2024 at 10:00 AM.

Student number:	4648269
Project duration:	April, 2024 – September, 2024
First Supervisor:	Savvas Zannettou
Chair:	Aaron Ding
Faculty:	Faculty of Technology, Policy and Management, Delft

This thesis is confidential and cannot be made public until December 31, 2024.

Cover:	Canadarm 2 Robotic Arm Grapples SpaceX Dragon by NASA under CC BY-NC 2.0 (Modified)
Style:	TU Delft Report Style, with modifications by Daan Zwaneveld

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Summary

The rapid rise of short-format video platforms such as TikTok and YouTube shorts in the last 5 years has fundamentally transformed the way that users consume content. These platforms rely more than any other on recommendation systems to provide content to users. These systems analyze user behaviors on certain types of content to learn their interests and deliver content to users directly, limiting direct user inputs. While this innovative approach to content delivery simplifies the user experience and generally increases user satisfaction, there are concerns that they can lead to an increase in rabbit holes. These rabbit holes are defined as situations where users are recommended increasingly narrow content which can reinforce biases and limit diverse perspectives and content.

The scientific body on recommendation systems has proven time and time again that they generate rabbit holes. However, a gap exists in the understanding how exactly these rabbit holes form specifically on Short-Format Video (SFV) Platforms, such as YouTube Shorts or TikTok. While many studies have attempted to audit recommendation systems there is a lack of methodology to effectively prove the importance of different users actions in both influencing the output of the algorithm as well as the generation of rabbit holes. To address this knowledge gap, the main research question of this study is:

"How can the feature importance of user actions in algorithmic recommendation systems on Short-Form Video Platforms, such as YouTube Shorts, be analyzed to understand their role in leading users down a rabbit hole?"

This research question can be divided into two sub-questions, which when answered will provide a comprehensive understanding to the main research question. These sub-questions are as follows:

1. How can a tool be built to collect a large dataset of algorithmic recommendations based on user actions?
2. How can this dataset be analyzed to understand the key user actions that lead to falling into a rabbit hole on YouTube Shorts?

As seen by the sub questions, this research will focus on distinct phases and objectives. The first will be to develop a tool that is able to fully simulate a human user of YouTube Shorts, the SFV platform of interest in the study, with full user interests, actions and decision making. This will permit the study to then focus on the second phase which will be to run this automation tool with varying user personas each interacting with content in different ways and recording all the output of the algorithm. This tool is novel in that it utilizes Large Language Models (LLMs) prompts in order to understand the content of the videos it is watching and decide how to react (positively or negatively) based on the video classification and the user interests. This will allow the researchers to understand how different user actions influence the algorithm and permit it to learn the user interests. It will also permit the researchers to investigate if rabbit holes form on YouTube Shorts as well as the user actions that intensify them. The tool is built to collect a time series of video recommendations with two key data variables for each point: the video category and the reaction of the user (positive or negative).

A time series of 550 videos is collected for the watch, like, share, dislike and skip reactions. The runs for the watch and like are ran at varying probabilities of the reaction happening to moderate their influence. The research then analyses these time series by plotting the running average of videos that positively aligned with the user interests as well as the different video categories recommended to the user. These plots reveal which actions are most important in influencing the algorithm and pertaining to user interests as well as how rabbit holes form. These results are presented below:

- Au automated tool to fully simulate users on YouTube Shorts was successfully developed and can now be used perform algorithmic audits. A full explanation of the development and architecture of the bot/tool is provided in the methodology section in order. The tool itself is also provided to the scientific community on GitHub in order for other researchers to perform more audits.

- Among user interactions, watching videos is identified as the most influential in guiding the YouTube Shorts algorithm. Liking also has a significant impact, though less than watching. In contrast, sharing and disliking videos have minimal influence on the algorithm's output.
- The results also confirms that the YouTube Shorts algorithm has a strong tendency to create rabbit holes by progressively narrowing down content recommendations based on user engagement, particularly through actions like watching and liking videos. It is also shown that the consistency in user actions is key in the intensity of rabbit holes and that to avoid rabbit hole.

From these results, this study aims to advance the understanding of how YouTube Shorts' recommendation algorithms foster algorithmic rabbit holes. The study introduces a novel automated bot for simulating user behavior, offering a new tool for auditing algorithmic processes on short-form video platforms. By providing insights into the mechanics of content personalization, this work contributes to ongoing discussions about algorithm transparency and ethical considerations in digital content curation.

The research on auditing SFV platforms for the rabbit hole effect aligns seamlessly with the overarching theme of the CoSEM program within the Technology Policy Management faculty. By delving into the intricacies of algorithmic recommendation systems on SFV platforms, the study intersects policy and technology within the context of a complex system. The objective is not only to understand the dynamics of these algorithms comprehensively but also to provide a tool for policy makers to analyse and understand the algorithms shaping our world. The results of this thesis can help platform developers optimize algorithms to reduce rabbit holes, guide policymakers in crafting regulations to promote transparency, and inform users on how to engage more critically with content.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

Abbreviation	Definition
SVF	Short Video Format
LLM	Large Language Model
RS	Recommendation Systems
...	

1

Introduction

1.1. Background on YouTube and Recommendation Systems

With billions of daily users, YouTube is the world's largest video-sharing platform and second largest social media platform [32]. YouTube users can upload videos, view, rate, share, add to playlists, report, comment, and subscribe to other users . Since it was created in 2005, YouTube focused mainly on traditional long-form videos that were basically delivered through the user search system—this means that users would search for what they wanted to consume [25].

However, the launch of YouTube Shorts in 2021 marked a shift towards short-form video content delivered similarly as on TikTok and Instagram reels [30]. Indeed, unlike traditional YouTube, where users can actively search and choose videos, YouTube Shorts presents content in a linear, autoplay format. This means users have minimal control over the next video they see, as the platform automatically queues up the next short based on its recommendation algorithm [15].

1.1.1. Understanding the Recommendation Algorithm

The recommendation algorithm of YouTube and especially YouTube Shorts is one of the core pillars of the platform and generally responsible for what content the user will consume [16]. Therefore, the primary goal of this algorithm is to increase user engagement and content consumption by trying to provide users with personalized content that is as engaging as possible. The algorithm achieves this by analyzing user behaviors on the platform, that is to say, it will record how users react to certain types of content (likes, watch time, subscriptions...) to understand the interests of users and provide them with similar content [15].

Nevertheless, the exact inner workings of the YouTube recommendation algorithm are not public, forming what is known as a black box system [22]. This opaque nature of the system presents a challenge for consumers and creators alike who seek to optimize their content consumption and delivery.

1.1.2. Recommendation Systems

The YouTube recommendation algorithm is not unique in the world as it is one of many recommendation systems that have infiltrated most people's content consumption strategies. These systems are now widely used across various digital platforms, from e-commerce sites, which push products they believe you will buy to the front page, to social media networks, which now automatically prioritize user recommended feeds over user subscribed ones [5].

In general, most recommendation systems function in the same way by collecting user interaction data and associating it with the product/content that was interacted with. These interactions are then classified in either positive or negative ways with varying levels of importance, while the products/contents are categorized based on similarity . By analyzing these interactions, the algorithm can predict user preferences and dynamically adjust the content it recommends to increase user engagement.

The specifics of each recommendation system is unique to its platform as the content on the platforms and possible user interactions vary in nature. In the case of YouTube Shorts, the user interactions that the recommendation system track includes which videos are watched to completion, which ones are skipped, liked, disliked, and how often videos are shared [15].

1.2. Research Problem and Objective

1.2.1. Research Problem

The rapid shift from traditional user subscription to algorithmic recommendation systems for content delivery has significantly altered how users consume media and poses a new set of challenges to the social media landscape. Indeed while these systems are designed to enhance user engagement and maintain users interest for as long as possible, they have inadvertently caused negative effects such as rabbit holes and echo chambers [14].

To explain in more detail the issue with these effects, rabbit holes generally refer to situations where users are continually fed more and more narrow content categories based on their viewing history in a bid to keep them engaged for longer. Unfortunately, this tends to lead users down a path of increasingly specific and often radical content as content that would have been considered extreme becomes the new normal and more and more extreme content is required to obtain the same reaction from the user.

Echo chambers, on the other hand, occur when users are exposed to content that mainly only aligns with their own preexisting beliefs. Contrasting perspectives/beliefs are also minimized or excluded leading users think their truth is absolute and everyone agrees with them.

Alas, as one can imagine, both rabbit holes and echo chambers pose significant societal risks. As they present more and more extreme content, the user's Overton window—the range of ideas tolerated in public discourse—shifts towards more extreme positions. Users can be radicalized as they become insulated to contrasting views within homogenous groups that amplify their views without challenge or nuance. This radicalization can have real-world consequences such as undermining healthy public discourse, and increasing polarization [4]. These effects are evident within the American political sphere where Republicans and Democrats now see each other as enemies instead of neighbors [11].

Hence, given these risks, it is critical to understand which specific user behaviors influence recommendation systems and furthermore how these behaviors contribute to the propagation of rabbit holes and echo chambers. By gaining a detailed understanding of these mechanisms, we can potentially offer users scientifically-backed strategies to tailor their interactions with these systems. Users could then consciously mitigate these negative effects and consume media in a healthier manner.

1.2.2. Research Objective

The primary objective of this study is to scientifically and quantitatively determine how different user actions influence YouTube's recommendation algorithm. This involves identifying which behaviors are most likely to lead users into rabbit holes or echo chambers and which actions can help avoid these pitfalls. Specifically, this research aims to:

Analyze the Influence of User Behaviors: Determine how various user interactions, such as liking, disliking, commenting, and sharing, impact the recommendations users receive.

Quantify Behavioral Impacts: Measure the extent to which different actions affect the algorithm's output, identifying which behaviors have the most significant influence.

Develop Guidelines for Users: Provide actionable insights and guidelines for users on how to interact with YouTube in ways that minimize the risk of falling into rabbit holes and echo chambers.

By achieving these objectives, this study seeks to empower users with the knowledge needed to manage their digital environments more effectively, fostering a more balanced and diverse media consumption experience. This research also aims to contribute to the broader understanding of recommendation systems, offering insights that could inform the development of more transparent and user-friendly algorithms.

1.3. Research Questions/Hypotheses

The central research question guiding this study is:

"How can the feature importance of user actions in algorithmic recommendation systems on Short-Form Video Platforms, such as YouTube Shorts, be analyzed to understand their role in leading users down a rabbit hole?"

To address this overarching question, we will investigate two sub-questions that aim to fulfill the intermediary objectives of the project.

The two sub-questions go as follows:

1. How can a tool be built to collect a large dataset of algorithmic recommendations based on user actions?
2. Can this dataset be analyzed to understand the key user actions that lead to falling into a rabbit hole on YouTube Shorts?

The primary goal of the first sub-question is to develop a tool capable of automatically generating a comprehensive dataset of algorithmic recommendations influenced by user interactions. This tool will log a detailed history of recommended videos at specific times for various users, based on different actions performed. It is hypothesized that it is possible to construct an automated system that effectively

simulates user interactions and captures the resulting recommendations, thereby creating a robust dataset for analysis.

The main objective of the second sub-question is to utilize the collected dataset to identify and analyze the specific user actions that trigger the recommendation algorithm to push users into a rabbit hole. This involves determining which interactions are most influential in leading to increasingly niche or narrow content categories. It is hypothesized that the dataset will reveal discernible patterns and key user actions that significantly influence the recommendation algorithm, facilitating a deeper understanding of how users can be led into rabbit holes on YouTube Shorts.

1.4. Scope and Limitations

The scope of this study will be confined to the design and understanding of an automated application that simulates user interactions with YouTube Shorts. This app will perform ”runs,” where a simulated user with predefined possible reactions interacts with the platform. By adjusting the probabilities of each reaction during these runs, the study aims to identify which user actions are most influential in shaping the recommendation algorithm. This targeted approach allows for a focused examination of the algorithm’s behavior in response to different user interactions.

Additionally, the study will explore the formation of rabbit holes, investigating the topics on which they develop based on user interests. While the goal is to determine whether rabbit holes occur and to identify the general themes or topics that trigger them, the study acknowledges the vast complexity of YouTube’s recommendation algorithm. This complexity, coupled with the diversity of content categories, means that the research will not attempt to target or analyze a specific rabbit hole in detail. Instead, the focus will be on understanding the broader patterns and mechanisms at play.

2

Literature Review and Theoretical Perspective

In this section we will review the existing literature on the main topic matters discussed in this Thesis. Firstly we go over the established literature on recommendation systems before delving deeper into Rabbit holes and their link to recommendation systems. The literature review will then shift its attention towards auditing theses algorithmic recommendation systems for transparency as well as user behaviors and their influence on recommendation systems. These claims will be substantiated with a case study of how these algorithmic ally generated rabbit holes have lead to the polarization of politics on social media platforms. Finally, from the results of this literature review, this section will identify the knowledge gap that is to be answered by the research of this thesis.

2.1. Research Strategy

In order to provide a clear and comprehensive state-of-the-art regarding recommendation systems, rabbit holes and analyzing them, the following research strategy was developed. Papers were identified by searching research databases such as Semantic Scholar and Scopus using the following keywords: TikTok, YouTube, algorithm, recommendation systems, rabbit hole, algorithm audit, social media, and personalization. These keywords were combined into different search queries with Boolean operators and quotation marks for specific content to be shown.

2.2. Results

A synthesis of the literature is presented in the 2 tables below. Table 2.1 presents common limitations of the current literature. Table 2.2 shows common future research topics from the existing literature. Subsequently, Appendix B Table B.1 showcases and summarizes the articles selected for this literature review.

Table 2.1: Common and relevant limitations in the current literature

Limitations	**Articles**
Limited to certain events or geographical location	(Klug et al. 2021), (Bandy & Diakopoulos 2020), (Hussein et al. 2020)
Does not explicitly examine algorithm behavior	(Klug et al. 2021), (Dündar & Ranaivoson 2022)
Focuses only on trending videos / Limited video sample size	(Klug et al. 2021), (Dündar & Ranaivoson 2022)
Play count is used as a proxy for algorithmic visibility	(Bandy & Diakopoulos 2020)
Does not explore the negative consequences of algorithmic amplification	(Bandy & Diakopoulos 2020)
Only certain types of content are analysed	(Boeker & Urman 2022), (Liu et al. 2023), (Dündar & Ranaivoson 2022), (Tomlein et al. 2021), (Hussein et al. 2020), (Srba et al. 2023)
Does not consider user demographics	(Boeker & Urman 2022), (Liu et al. 2023)
No comprehensive analysis of TikTok algorithm	(Boeker & Urman 2022)
Focuses on controversial topics	(Cinelli et al. 2021)
No solution to combat echo-chambers	(Cinelli et al. 2021), (Bryant 2020)
Only a literature review	(Bryant 2020)
Preprogrammed agents do not fully capture the complexity of human users	(Tomlein et al. 2021), (Boeker & Urman 2022), (Dündar & Ranaivoson 2022), (Srba et al. 2023)
Focuses only on debunking videos to burst the filter bubble effect	(Srba et al. 2023)

Table 2.2: Common and relevant future topics in the current literature

Future Topics	Articles
Analyse how user interaction changes algorithmically generated content feed	(Klug et al. 2021), (Bandy & Diakopoulos 2020), (Bryant 2020), (Dündar & Ranaivoson 2022)
Isolating algorithmic visibility	(Bandy & Diakopoulos 2020)
Exploring the ethical implication of algorithmic amplification	(Bandy & Diakopoulos 2020), (Cinelli et al. 2021), (Dündar & Ranaivoson 2022), (Woolley & Sharif 2022)
Investigate social media algorithms for filter bubbles	(Boeker & Urman 2022), (Cinelli et al. 2021), (Bryant 2020)
Explore user demographic role	(Boeker & Urman 2022), (Liu et al. 2023), (Cinelli et al. 2021)
How specific content users seek out affects the algorithm	(Boeker & Urman 2022)
Investigate the effectiveness of filter bubble countermeasures	(Boeker & Urman 2022), (Cinelli et al. 2021), (Bryant 2020), (Dündar & Ranaivoson 2022), (Tomlein et al. 2021), (Hussein et al. 2020), (Srba et al. 2023)
Impact of echo chambers on misinformation spread	(Cinelli et al. 2021), (Tomlein et al. 2021), (Hussein et al. 2020), (Srba et al. 2023)
Developing more human-like bot simulations for algorithm audit	(Tomlein et al. 2021), (Srba et al. 2023)
Continuous audit runs to study the longitudinal effects of personalization	(Hussein et al. 2020)

2.2.1. Recommendation systems

The Rise of Recommendation Systems

At the inception of Social Media platforms the vast majority of content user's viewed was provided in users-subscribed content feeds. These types of feeds, also known as chronological or reverse-chronological feeds, presents content to the user in the order it was posted and from sources that the user choose by subscribing, following, or friending other users or pages [5].

These types of feeds have three main key features. Firstly, content is shown in chronological order based on the time of posting. Secondly, users have direct control over the content they see by choosing whom to follow. Thirdly, all the posts from subscribed sources are shown, regardless of perceived relevance or engagement potential [9].

On the other hand, algorithmic feeds use complex algorithms to curate and prioritize content based on various factors, such as user behavior and interactions, engagement metrics and inferred preferences. To be more clear these algorithms analyze how users behave towards content they are shown and then recommends similar or different content based on how the user reacted [5].

By contrast to chronological feeds, algorithmic feeds provide three different key features. Firstly, the feed is tailored to the user's perceived interests. Secondly the algorithm prioritizes content likely to generate higher engagement, such as likes, comment, shares, or views. Finally, users are exposed to a broader range of content, including posts from accounts they don't follow and which the platform believes they might find interesting [18].

While most social media platforms initially launched with chronological feeds, including Facebook in 2005, Twitter in 2007, and Instagram in 2010, they all gradually shifted towards algorithmic recommendation systems throughout the 2010s. At first, the platforms simply included some alternative feeds that were driven by algorithmic recommendations, such as Facebook's "News Feed" in 2009 or Twitter's "While You Were Away" feature in 2015. However, through multiple updates, social media platforms have, slowly but surely, completely transitioned to algorithmic feeds, including Instagram, which switched completely in 2016. Twitter (now known as X) has also switched to an algorithmic recommendation system and sidelines by default the chronological "Following" feed for the algorithmic "For You" Feed [5].

Youtube's Recommendation System

The platform of interest for this study, YouTube Shorts, also makes use of a recommendations system. Although the YouTube Shorts algorithm has not directly been investigated by the literature, we can assume that it employs similar techniques to the classical YouTube algorithm which has been extensively researched.

According to research performed by Covington et al., 2016, YouTube's algorithm utilizes deep neural networks to recommend videos to users. This process is divided into two stages: the candidate generation stage and the ranking stage. The candidate phase involves selecting a subset of videos from a larger library through the use of collaborative filtering techniques. These filters analyze user activity and historical interaction patterns in order to identify potentially relevant content .

Subsequently, the ranking stage orders these candidates using a multi-objective ranking model. This model prioritizes videos based on predicted user engagement, incorporating features such as watch time, likes, and other user interactions [10].

As shown by Liu, Wu, & Resnick, 2023, these two stages happen every time a new video is consumed by the user as continuous real-time data processing and feedback loops are integral to refining these recommendations. This ensures that the model adapts to changing user behaviors and preferences [20].

2.2.2. The Rabbit Hole Effect

As seen in Chapter 1, with the increase in algorithmic recommendation systems, there is concern that algorithmic side effects such as filter bubbles, echo chambers, and rabbit holes will also rise [6]. In this study, we will primarily focus on rabbit holes and this section will explore the existing literature on rabbit holes including their definition, the psychology of why users get stuck in them, and the role of algorithmic recommendation systems in facilitating this effect.

Firstly, lets define what rabbit holes are. According to, Le Merrer et al., 2023, rabbit holes can be defined as the result of advanced personalization mechanisms that create an increasingly narrow content feed [19]. Over time, the feed shifts from mainstream recommendations to highly specialized ones, often triggered by initial user engagement. This shift has frequently been identified by the literature on platforms like YouTube [8, 7, 19, 31]. This process often isolates users in a specific content loop, limiting exposure to diverse perspectives [19] and leading to other negative side effects of recommendation systems such as filter bubbles and echo-chambers [19, 29].

Now that we are able to define rabbit holes, there comes the question of why do users get stuck in them? This can be explained by two psychological concepts: accessibility and immersion. Indeed, According to "Down a Rabbit Hole: How Prior Media Consumption Shapes Subsequent Media Consumption," by Woolley et al., 2022, consecutive consumption of similar media enhances the accessibility of the shared category in the user's mind [31]. By increasing accessibility to certain content categories, the recommendation algorithm also increases the immersion of the user, making the category more appealing to the user. This phenomenon occurs because, the more accessible a category becomes, the more immersive and enjoyable it feels, driving users to seek out similar content [8].

Therefore, it is possible to assume that interruptions can mitigate the rabbit hole effect by reducing the consecutive exposure to similar content. Indeed, Almachnee et al., 2022, showed in their paper that by breaking the immersion and accessibility cycle, the rabbit hole effect was reduced[1].

The psychology of rabbit holes can therefore be directly linked to the functioning of algorithmic recommendation systems. In order to maximize user engagement, these algorithms continuously present content to the user known to be of interest. This process can be directly linked to the concepts of accessibility and immersion as discussed above. Indeed, as the user repeatedly engages with content from a specific category, their immersion is deepened and alternative content becomes less accessible[7, 27].

2.2.3. Methods of Algorithmic Auditing For Rabbit Holes

As seen in the two sections above, rabbit holes are a consequence of the use of recommendation systems for content delivery to users. However these systems, often referred to as "black boxes",

Audit Method	Article
Manual Audit	[12]
Sock Puppet Audit	[24, 5, 14, 16]
Bot Centric Audit	[6, 19, 28, 27, 20]

Table 2.3: Types of Algorithm Audits Performed in Literature

are incredibly complex and often even their creators cannot explain their output [22]. Recognizing the significance of these dynamics, algorithm auditing emerges as a critical practice involving systematic evaluation of recommendation algorithms to ensure fairness, transparency, and accountability, particularly in Short-Form Video (SFV) platforms [3].

Many different types of algorithmic audits exist with different methodologies but the outstanding principle generally remains the same. Provide and record a large number of varied inputs into a algorithm and record the output. By analyzing many of these transformation of the input into the output by these ”black boxes”, the researchers hope to understand which inputs create which outputs, thereby gaining a better insight into the inner-workings of the algorithm [21].

From the literature three main methods of algorithmic auditing emerge as shown in Table 2.3. Firstly, there is Manual auditing with a human user as performed by [12]. This methodology requires a human user to interact with the algorithm. While this methodology is often the easiest to perform it limits the ability of large scale data collection and repeatability. Secondly, there is action less ”sock puppet” auditing where sock puppets are trained on a predefined data-set for the algorithm to learn their interests. The researchers then capture the recommendations that the algorithm provides these sock-puppets. Finally there are bot centric audits where full bots simulate user actions as realistically as possible by watching videos and interacting with content in a specific way. Bot centric audits therefore permit the researchers to understand the path that users are lead down in real time by recommendation systems.

While these methodologies have consistently been successful in demonstrating the role of recommendation systems in the formation of rabbit holes [20], they do have their limits. Apart from [6] none of the audits performed truly encapsulated the behavior of a human user. Mostly the content watched, by the bots and sock-puppets, was either predefined by the training data of the sock puppets or chosen at random from recommendations in the bot centric runs. In reality the process of choosing content from a user is more complicated and aligns more directly with a certain set of interests. This is why [18] advocates for the use of bots, which is complicated by the anti-bot techniques used by platforms.

According to [18] these bots should be able to simulate human behavior accurately by consuming content and most importantly reacting to content in real time.

2.2.4. Results from Algorithmic Audits

The scientific body shows that auditing recommendation systems is not a novel concept. Several studies have used different methods to audit YouTube’s recommendation algorithm for misinformation filter bubbles. Hussein et al. and Papadomou et al. both used a similar method of creating user profiles and tracking the recommendations over time [16, 24]. Tomlein et al. used two different methods: crowdsourcing audits, which involved real users rating the recommendations, and sock-puppet audits, which involved creating fake accounts and simulating user behaviour [29]. Spinelli et al. used a method of comparing the recommendations and search results of different user groups [27]. These studies provide evidence of the existence and properties of misinformation filter bubbles on YouTube.

As common knowledge dictates, on top of a thorough literature review, it is proven that if users engage with certain types of content, recommendation algorithms tend to prioritize and suggest similar content, immersing users in a specific category or topic [31]. This phenomenon, known as the rabbit hole effect, occurs due to the heightened accessibility of the shared category of previous media experiences [29]. For example, studies have shown that watching a series of misinformative videos can strengthen the presence of such content in recommendations [8]. YouTube’s algorithmic recommendations have been criticized for potentially leading users to harmful or extreme content [28]. The algorithm’s influence on users’ content consumption is expected to be influenced by metrics that depend on user engagement, language, location, and video viewing length.

2.3. Identification of Gaps

The existing literature reveals a notable knowledge gap in the realm of Short-Form Video (SFV) platforms, particularly concerning the need for studies that employ a tool with users making real-time decisions. This crucial gap hinders a comprehensive understanding of the dynamics between users and algorithmic recommendation systems [3, 7, 12, 17]. Furthermore, most algorithmic studies have not been performed on SFV platforms highlighting another layer to the knowledge gap.

The primary advantage of employing a tool with real-time decision-making capabilities is the potential to construct a substantial dataset efficiently and with minimal human intervention. This approach allows for the inclusion of diverse user profiles, capturing variations in personalities, preferences, and decision-making processes [18].

The lack of specific research in this area is particularly significant because no study has explicitly addressed the fundamental objective of recommendation systems on SFV platforms – the impact on user engagement. It remains unclear whether these systems are intentionally designed to guide users into content rabbit holes or if such outcomes are inadvertent side effects of the algorithms. Investigating the fundamental objectives of recommendation systems and discerning the intentional or unintentional nature of steering users towards specific content realms constitutes a critical gap in the current literature.

Furthermore, the absence of a dedicated tool to simulate user interactions and assess how algorithmic recommendations influence these simulated users adds another layer to this knowledge gap [28, 29]. Such a tool is essential for systematically studying the interplay between user behaviour and algorithmic decision-making. A simulation tool is necessary for researchers to comprehensively understand how different user personas are affected by the recommendation algorithms in real-time scenarios, impeding insights into the mechanisms shaping user experiences on SFV platforms and previous research mainly focused on using preprogrammed bots that did not fully capture the complexity of a human [6, 12, 28, 29]. Addressing these gaps is paramount for advancing our understanding of the nuanced relationships between users and recommendation systems in the ever-evolving landscape of short-form video content.

3

Research Methodology

The following section will present the Research Methodology that was followed during this Thesis project. Given the complexity of the system that is to be analyzed, a mixed-methods approach is required. This approach combines quantitative and qualitative methods to provide a comprehensive analysis of the phenomena being studied. The different phases of the methodology are presented in Figure 3.1.

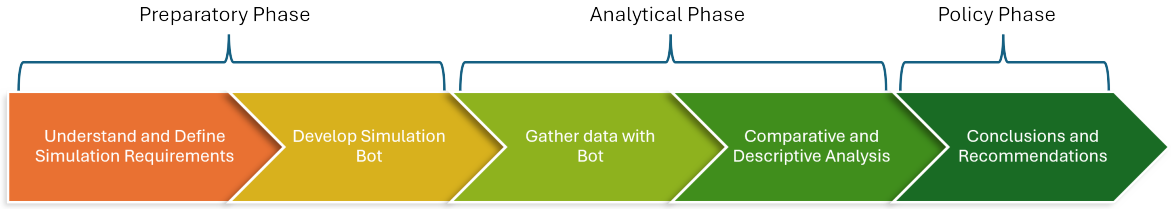

Figure 3.1: Research Methodology Process Diagram

The preparatory phase begins with understanding and defining simulation requirements. This initial step sets up the framework for building a tool capable of simulating users and there-by auditing the YouTube Shorts Algorithm. After building the bot, it needs to be heavily tested and further calibrated to produce reliable results. Once the bot is sufficiently robust, the analytical phase can begin by running the bot with different predetermined configurations so as to understand the affect of different user actions on the algorithm.

The quantitative component of this methodology aims to systematically collect user watch history, focusing on the interactions users make with specific videos. This data provides a robust foundation for understanding the mechanics of content exposure and consumption. The qualitative aspect delves deeper into the reasons behind the formation of ”rabbit holes” on the YouTube Shorts platform, exploring the nuances of user engagement and the potential strategies for avoiding these phenomena.

3.1. Research Design

3.1.1. Why is a bot required?

One of the primary challenges in understanding the YouTube Shorts recommendation algorithm is the necessity of collecting a vast amount of data. Analyzing the algorithm’s behavior requires the accumulation of numerous data points, which entails watching and interacting with potentially hundreds or thousands of consecutive videos to discern any trends in recommendations.

Moreover, this thesis aims to identify the key user actions that have varying degrees of influence on the YouTube Shorts algorithm and the recommendations it generates. Achieving this objective necessitates performing tens of thousands of video-watching and reacting iterations across different user personas and interaction rules. Considering that a single YouTube Short can last up to 60 seconds (with an average duration of [insert actual number here] seconds), manually obtaining this dataset would demand hundreds of hours of labor-intensive effort. Additionally, maintaining the consistent focus required to react according to specific user personas is impractical for a human to sustain over extended periods.

To address these challenges, an automated tool/bot needs to be developed to facilitate data collection without necessitating constant human supervision. This bot would allow the researcher to set up the simulation, run it in the background, and obtain the results after a few hours.

3.1.2. Simulation requirements

The objective of the bot is to simulate a human user consuming content on YouTube Shorts. As Figure 3.2 illustrates, human users begin their interaction with YouTube Shorts by opening the app or website and navigating to the Shorts feed. They are then feed a video by the algorithm which the user starts consuming. The user then quickly (less than 10 seconds generally) decides if the video is relevant or not to his/her interests. If they are the user will watch the video in full and possibly like or share the video. If the video is not of interest, the user will skip the video, and possibly dislike, to access the

next video provided by the algorithm. This loop then repeats until the user decides to stop consuming content.

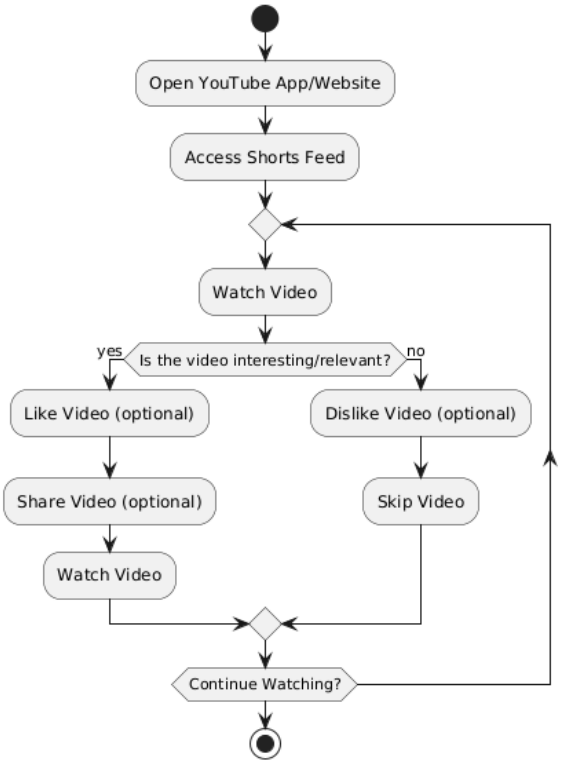

Figure 3.2: Human User Process Flow Diagram

From this Human User Process Flow, three key user actions emerge. Firstly users must understand what type of content they are watching. The user then needs to decide if the content aligns with his/her interests. Finally the user interact with the content in the appropriate manner.

From these key user actions the key features required from the simulation can be defined. First, the bot must be able to mimic the user initialization by starting a session with specific user interests and navigating to the YouTube Shorts feed. Then the bot needs to be able to engage with videos by actually watching them and perform user reactions such as liking or disliking. This can be achieved through web automation tools such as Playwright. After accessing and interacting with videos, the bot needs to be able to understand the type of content it is consuming and then decide based on its user interests if the video pertains to them or not. By forming an opinion like a real user would, the bot simulates human user decision making and is able to react with the appropriate use reaction. Finally the bot needs log all the data of the interaction including the video watched, the content classification of the video and the user action performed on the video by the bot. These features let us create a general overview of the action flow of the bot which is represented in Figure 3.3.

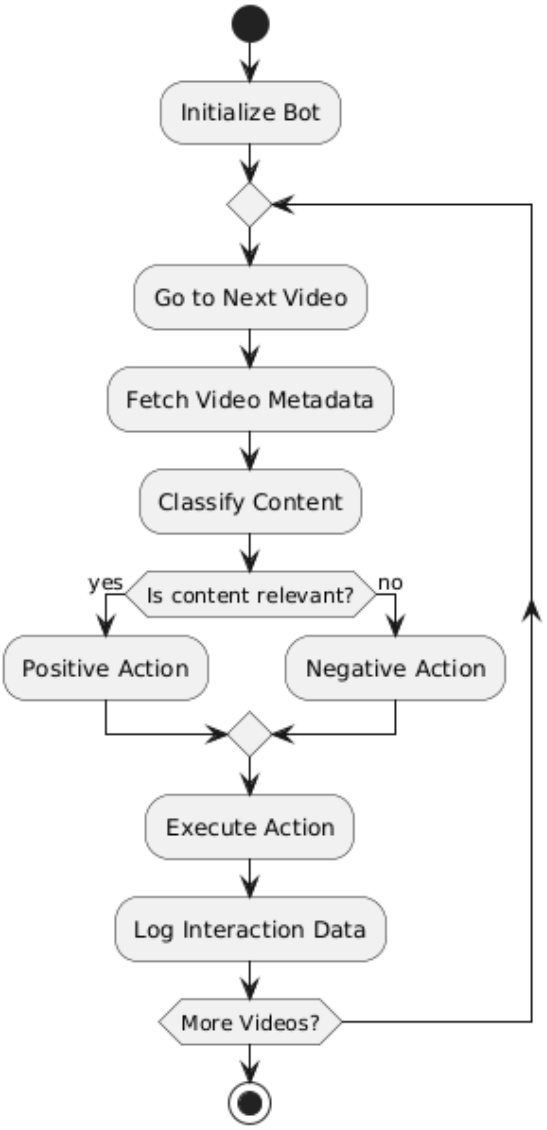

Figure 3.3: Bot Process Flow Diagram

Comparing the bot process flow diagram (Figure 3.3) to the human user process flow diagram (Figure 3.2), we can clearly see that the overarching flow is very similar both consisting of a loop with a decision based on the relevance of the video to the user interests. The main differences between the bot and the human user pertains to their ability to understand the video content and deciding how to react to the video.

It goes without saying that human users mainly utilize their eyes and brains to actually watch the video and understand its content category. Unfortunately this is not really possible with a bot. Indeed, while there have been great advances in video image recognition, these techniques are still relatively novel and very computationally intensive. Indeed, as explained below, one of the user reactions that demonstrates a negative reaction towards a video, and therefore that it is irrelevant to the user, is the action of quickly scrolling to the next video. Therefore, reaction speed is crucial in effectively simulating a human user. This makes it so that computer vision cannot be used. Instead, the bot will instead classify the videos by processing the video metadata, obtained from the YouTube API, through a LLM which are known for their excellent classifying capabilities.

Given that this thesis aims to understand how the individual reactions influence the algorithm, the

actions that the bot will perform will be predetermined at the initialization step. As can be seen in Figure 3.3, the bot will decide if the video is either ”relevant” or ”irrelevant” (if the video pertains to the user interests or not). From this binary ”relevant” or ”irrelevant” decision the bot then acts accordingly to the specified reactions. These reactions are shown below:

Positive Reactions

- **Watching the entire video**: Indicates engagement and interest.
- **Liking the video**: Shows a positive reception.
- **Sharing the video**: Demonstrates a high level of interest and willingness to promote the content.

Negative Reactions

- **Scrolling to the next video quickly**: Implies disinterest or dislike.
- **Disliking the video**: Indicates a negative reception.

By setting up predetermined reactions, the research is able to focus on specific user actions and analyze how they individually or collectively impact the algorithm.

3.1.3. Bot Architecture

Now that we have a good understanding of the requirements for the bot we can proceed with building it. Figure 3.4 shows overarching architecture of the bot with all its components which we will now discuss individually in detail.

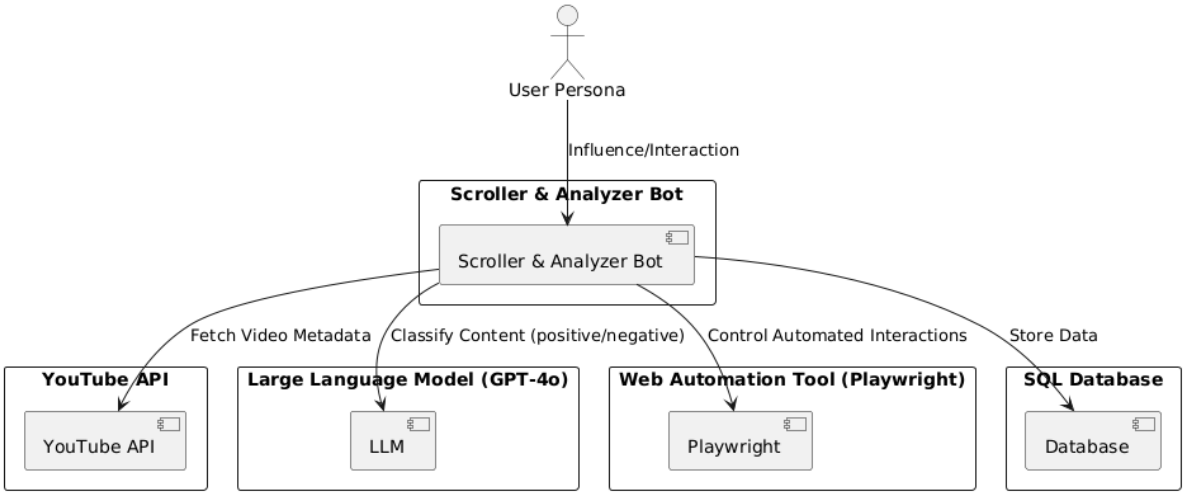

Figure 3.4: Bot Component Diagram

User Persona

The User persona component has three primary functions, firstly to initialize the username, email, password, date-of-birth and sex of the user. Secondly, to set the interests of the simulated user and thirdly to define what interactions the bot should perform. The interests can be any category of content that the user might enjoy such as ”Sports Cars” or ”Apple Technologies” and the user can have as many interests as desired.

The user persona also defines the actions that the bot will perform depending on if the video is of interest (positive) or not (negative). The actions have a probability of happening, this lets us modulate the effect of positive reactions and therefore analyze how the algorithm reacts when the bot likes 10% of relevant videos compared to when it likes 100% of relevant videos.

```
{
    "id": 5,
    "username": "50percentwatcher",
    "email": "percentwatch477@gmail.com",
    "password": "Watch50%!!",
    "dob": "1995-09-02",
    "sex": "Male",
    "interests": [
        "Technology",
        "Science",
        "Engineering",
        "Mathematics",
        "Education",
        "Pets",
        "Motorcycles",
        "Cars",
        "Sports"
    ],
    "reactions": [
        {
            "reaction": "like",
            "probability": 0
        },
        {
            "reaction": "dislike",
            "probability": 0
        },
        {
            "reaction": "watch",
            "probability": 0.5
        },
        {
            "reaction": "share",
            "probability": 0
        },
        {
            "reaction": "skip",
            "probability": 1
        }
    ]
}
```

Figure 3.5: User Persona JSON Code

Figure 3.5 shows the an example user persona coded in JSON. As can been seen from the figure, the user persona indeed as several interests, represented as a list, and the reactions are all defined with the probability of them happening. In this case the persona has a 50% probability of watching relevant videos and a 100% probability to skip irrelevant videos. In essence, this lets the researchers evaluate the behavior of the YouTube Shorts algorithm when a user watches 50% of the videos that are interesting to him and skips all the others, showing disinterest. By defining user personas in this modular way, researchers are able to modulate the types of reactions as well as their probability of happening on content of (dis)-interest to evaluate exactly how different user actions impact the algorithm.

Scroller & Analyzer Bot

The Scroller & Analyzer bot component is the main element of the simulation tool. It contains all the logic of the code, integrating and coordinating the various components. This component is responsible for controlling the flow of the simulation, ensuring that the web automation tool performs interactions correctly, fetching and analyzing video metadata, and logging data into the SQL Database. It acts

Data Type	Description
Title	The title of the video
Description	The description of the video
Tags	List of tags written by the video uploader
Channel	Name of the video uploader
YouTube Category	List of YouTube Categories assigned to the video by YouTube or the uploader
YouTube Topics	List of Topics YouTube assigned to the video

Table 3.1: Metadata obtained from YouTube Data API

as the central hub, orchestrating the entire process to ensure seamless operation and accurate data collection.

YouTube Data API

The YouTube Data API component is essential to the classification of videos. Thanks to its list function [2] it is able to return all the metadata of interest to the Scroller & Analyzer simply by providing the video ID, which itself is obtained by the Web Automation Tool Component. The metadata points of interest are shown in table 3.1.

It is important to note that while the YouTube Data API provides a categories that pertain to the video, these categories are very general and often not accurate to the actual content of the video. Indeed, there are only 15 official YouTube categories and therefore the bot cannot simply rely on this one data point. This is also true to a lesser extent of the YouTube Topics data point which provides a more fine grained description of topic related to the video but does not fully encompass the content.

Large Language Model (GTP-4o)

The Large Language Model (LLM) component has two main functions. Firstly, to classify the video based on the metadata into a specific category and as well to decide based on this category and the user interests if the video is relevant or irrelevant to the user.

LLMs have long been touted for their excellent classification performance [33] and the bot's need to classify videos into categories, it came as a natural choice to utilize a LLM for this function. The main reason for this is that contrary to other machine learning solutions, LLM's are already pre-trained on many billions of parameters and therefore do not require a large training dataset. To perform the classification of videos based on their metadata, the bot prompts an LLM with a classification message and the appropriate metadata, asking the LLM to classify the video into a parent and specific category. The prompt for classification are shown in figures 3.6 and 3.7.

```
"""\
classify the following video in the most similar category based on the following
    title, description, tags, channel name,
youtube category and youtube topics obtained from the youtube data api.


Sometimes the youtube API does not assign the correct category to the video so
    please categorize it better if you deem it neccessary.

the youtube topics are a list of topics that youtube has assigned to the video.

Try not to return generalized categories such as "entertainment" or "lifestyle" be
     a bit more specific by using the provided information.

Return the category name which should be one to two words as well as the parent
    category seperated by a comma.

The parent category is a more general category that the specific category falls
    under and should be one to two words.

Do not return an explanation why you chose a category

If you do not have enough information to categorize return the youtube category

To recap the format that you should return is: specific category, parent category
"""
```

Figure 3.6: System Classification Prompt

```
"""\
title: {title}
description: {description}
tags: {tags}
channel: {channel_name}
youtube category: {youtube_category}
youtube topics: {youtube_topics}
"""
```

Figure 3.7: User Classification Prompt

After obtaining the content classification of the video the LLM is then prompted a second time with the task to decide if the content classification aligns with the user interests. This is done because sometimes the classification result might not be identical to the interests but is still similar enough to qualify as relevant. In cases like this, correctly identifying the video as relevant is possible with the use of an LLM. Figures 3.8 and 3.9 show the prompt for decision making.

```
"""\
Assume that you are a youtube user that has the following user_interests and you
    have just watched the following video with the following title, description,
    tags, channel name, youtube category and youtube topics obtained from the
    youtube data api.

There is also a classification of the video that was made by a model giving the
    specific category and parent category of the video.

Based on the information provided how would the user react to the video based on
    his interests?

Do act strictly based on the user interests and the video classification. Take
    into account that the user is a human and not a robot.

The user might be open to watching videos that are not directly related to his
    interests but can be related in some way.

The possible reactions can we either positive or negative. A reaction can only be
    one of these two.

You should also return the reason why the user would react in that way in a few
    words.

The expected format is a JSON object with the keys "reaction" and "reason" with
    the values being a list of strings and a string respectively.

Example:

{
    "reaction": ["positive"],
    "reason": "The user would like the video because it is about a topic that he
        is interested in"
}

Example 2:

{
    "reaction": ["negative"],
    "reason": "The user would not like the video because it is about a topic that
        he is not interested in"
}

Ensure that the response is in JSON format but do not return code, only the JSON
    object as a string.
"""
```

Figure 3.8: System Decision Making Prompt

```
"""\
user_interests: {interests}
title: {title}
description: {description}
tags: {tags}
channel: {channel_name}
youtube category: {youtube_category}
youtube topics: {youtube_topics}
parent_category: {parent_category}
specific_category: {specific_category}
"""
```

Figure 3.9: User Decision Making Prompt

Model of LLM	Accuracy
Llama 3	66%
GPT-3.5 Turbo	73%
GPT-4o	86%

Table 3.2: Accuracy of Different LLMs in Classifying YouTube Videos Based on Metadata

Several models were evaluated for this task, including GPT-3.5 Turbo, GPT-4o, and local LLMs such as LLama 3 from Meta. Each model was tested on a sample size of 100 videos to assess their classification accuracy and speed. The results indicated that OpenAI's models had the highest accuracy. Despite the latency associated with API calls, GPT-3.5 Turbo demonstrated the highest speed, outperforming local LLMs in this regard.

Table A.1 shows the accuracy of the three different models testing when tasked to classify 100 videos based on metadata. As can be seen GPT-4o had by far the best results with 86% of videos being correctly classified. Therefore, given that the speeds of GPT-4o are comparable to GPT-3.5 Turbo the decision was made to use 4o for the bot.

Web Automation Tool (Playwright)

The web automation tool component performs several crucial functions for the bot architecture. Its primary role is to simulate actual user interactions with YouTube Shorts. This involves automating the process of opening the YouTube Shorts feed, navigating through videos, and performing specific user actions such as liking, disliking, skipping, and sharing. By automating these tasks, the tool ensures consistent and reproducible data collection, mimicking real user behavior accurately.

The tool watches videos to maintain the necessary viewing duration that influences the recommendation algorithm, helping to understand how different watch times affect the algorithm's output. Additionally, it extracts unique video IDs from the content it interacts with, which are then used to fetch metadata from the YouTube Data API. The Web Automation tool is also responsible for the controlled user actions such as liking, disliking, skipping or sharing videos.

Similarly to the LLM component, there are many different tools that could have performed this task, such as Selenium or Katalon Studio, but in the end Playwright was utilized. The reasoning for this choice is explained in Appendix A.

SQL Database

The SQL Database component is used to log all data of interest, ensuring a comprehensive dataset by the end of the process. A SQL Database is utilized to manage Create, Read, Update, and Delete (CRUD) operations, ensuring that all data is saved as the bot progresses. This prevents data loss if the bot crashes halfway through a run, maintaining data integrity and reliability.

3.1.4. Running the bot

Now that we understand the underlying component structure we can now design the sequence of actions that the bot will perform in order to accurately simulate a human user on YouTube Shorts. This

figure shows it

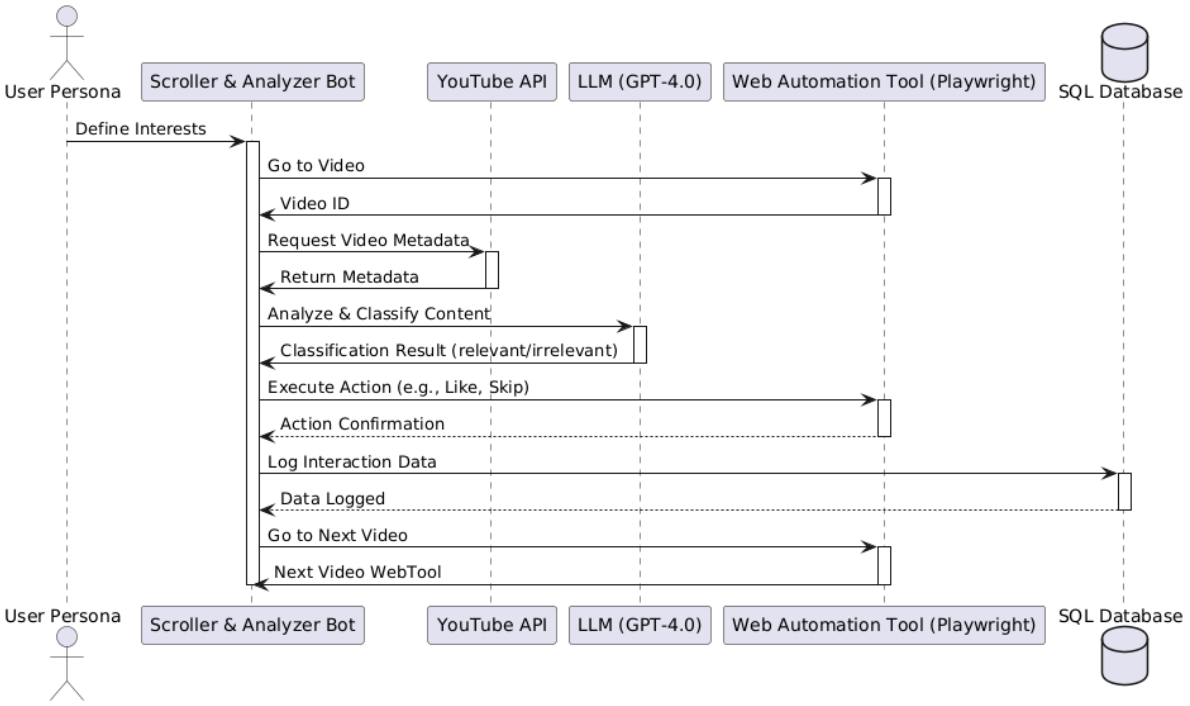

Figure 3.10: Bot Sequence Diagram

3.2. Data gathering

3.2.1. Dataset Structure and Key Data Points

This research aims to understand how YouTube Shorts recommendations evolve over time as simulated users interact with the platform according to predefined personas. To achieve this, it is crucial to collect and analyze a comprehensive dataset that captures both the videos' content and user reactions to them.

Dataset Structure

The structure of the dataset is fundamentally a time series of data points. YouTube Shorts consumption is inherently linear, as only one video can be shown at a time, and users have limited ways to interact with each video. Consequently, the overarching dataset comprises a sequential list of time-indexed data pairs: the video and the associated user reaction.

Video Dataset

Each entry in the video dataset includes detailed metadata about the video. The key data points of interest are as follows:

- **Video Title**: The title of the video.
- **Video Length**: The duration of the video.
- **Number of Likes**: The number of likes the video has received.
- **Number of Dislikes**: The number of dislikes the video has received.
- **Video Category**: A classification that clearly describes the type of video content. This is crucial for analyzing trends and preferences.

User Reaction Dataset

The user reaction dataset documents how the user responded to each video. Reactions are categorized into positive or negative based on the following actions:

Positive Reactions

- **Watching the entire video**: Indicates engagement and interest.
- **Liking the video**: Shows a positive reception.
- **Sharing the video**: Demonstrates a high level of interest and willingness to promote the content.

Negative Reactions

- **Scrolling to the next video quickly**: Implies disinterest or dislike.
- **Disliking the video**: Indicates a negative reception.

3.3. Data Analysis Techniques

After meticulously collecting data through our simulation bot, it is crucial to employ robust data analysis techniques to extract meaningful insights. This section outlines the various methods used to analyze the data gathered from the user interactions with YouTube Shorts. The primary focus of our analysis is to understand the recommendation algorithm's behavior, identify trends and patterns in video recommendations, and evaluate the influence of different user actions.

3.3.1. Dataset preparation

Given that the initial classification was performed by a large language model (LLM), there are instances where similar categories are labeled differently. For example, classifications such as ”Hobby” and ”Hobbies” or ”Travel & Events” and ”Travel,” as well as ”Football” and ”Soccer,” need to be consolidated. Manually grouping these classifications is possible but impractical due to the large number of specific categories, often numbering in the hundreds.

To address this challenge efficiently, we leverage the GPT-4o API to generate a remapping dictionary. Figures 3.11 and 3.12 show the prompts used to obtain the remapping dictionary. In this dictionary, the keys represent the new, consolidated categories, and the values are lists of the original categories that should be grouped under each key. This automated process significantly reduces the time and effort required for manual classification. Once generated, the remapping dictionary is manually reviewed for accuracy and adjusted as necessary. This ensures that the classification variables are consistently and accurately recorded.

```
'''

Here is a list of classifications. Some of the classifications are very similar to
     each other. Can you group similar classifications together?

What is meant by "similar" is that the classifications are related to the same
    topic or category. For example, "Football Highlights" and "Soccer Highlights"
    are similar because they are both related to Football it is just different ways
     of saying the same thing. Furthermore some categories encompass other
    categories for example "Pets and Animals" encompasses "Pets" so they should be
    combined into one category as well.

Example:

{
    "Advertising": ["Advertising"],
    "Art": ["Art", "Performing Arts"],
    "Automotive": ["Automotive", "Autos & Vehicles", "Vehicle", "Vehicles"]
}

Ensure that the response is in JSON format but do not return code, only the JSON
    object as a string.

'''
```

Figure 3.11: System Remapping Prompt

```
'''
classifications: {classifications}
'''
```

Figure 3.12: User Remapping Prompt

In addition to grouping classifications, it is crucial to ensure that all variables are in the correct format. This includes verifying data types, standardizing formats, and handling missing values appropriately.

3.3.2. Descriptive Analysis

To be able to understand the importance of user actions on the outcome of the algorithm, we will perform a descriptive analysis of each data set and then combine these results with being able to obtain cross-dataset analysis, which shows how important each user action is in influencing the algorithm.

To do this, we can first obtain a running average of user perception (relevant or irrelevant) towards videos, which will visualize how good the algorithm is at providing content of interest to the user, depending on how the user is reacting to the videos.

The running average at time x can be represented by the mathematical formula shown in equation 3.1.

$$\text{Running Average at time } x = \frac{1}{100} \sum_{i=x-100}^{x} R_i \tag{3.1}$$

Where:

- R_i is an indicator function that takes the value 1 if the video i is relevant, and 0 irrelevant.
- x is the index of the current video.
- $x - 100$ is the index of the 100th video prior to the current one.

- The sum counts the number of relevant videos between $x - 100$ and x, and dividing by 100 gives the running average.

This equation lets us produce a running average plot of relevant videos for each dataset (run of the bot). We can then combine these graphs across multiple datasets to compare different actions and how they impact the ability of the algorithm to produce relevant videos to the users over time. We can also compare runs where users performed with the same action, such as watching relevant videos and skipping irrelevant ones, but with varying probabilities, such as 10%, 25%, 50%, 75%, and 100%, of actually performing the action.

By analyzing this data, we can determine the importance of these actions. Additionally, this helps to identify which actions users should take to curate their feed in a specific way, thereby avoiding echo chambers and rabbit holes.

3.4. Experimental Setup

Now that a simulation bot has successfully been created, the research can move towards actually utilizing it to analyze and understand the YouTubeShorts recommendation algorithm. In this section, it will firstly explained how to setup the bot to run as required and then how it was setup for the research performed in this thesis.

3.4.1. How to setup the bot

To setup the bot to run as desired there are a two main steps. Firstly, a user must be defined in the users.json file and then a few variables including the llm of choice, and the run configuration need to be defined in the Scroller-AnalyzerV2.py file.

To begin, a user needs to be defined in users.json. This user must follow the same shape and variables as the example persona represented in 3.13. The user profile consists of 4 main parts.

```
{
    "id": 24,
    "username": "watcherAndHalfLiker",
    "email": "WatchAndLike@gmail.com",
    "password": "Watch100%Like50%!!",
    "dob": "1999-02-24",
    "sex": "Female",
    "interests": [
        "Pets",
        "Fashion",
        "Health and Fitness",
        "Beauty",
        "Travel",
    ],
    "reactions": [
        {
            "reaction": "like",
            "probability": 0.5
        },
        {
            "reaction": "dislike",
            "probability": 0
        },
        {
            "reaction": "watch",
            "probability": 1
        },
        {
            "reaction": "share",
            "probability": 0
        },
        {
            "reaction": "skip",
            "probability": 1
        }
    ]
}
```

Figure 3.13: Example User Persona JSON Code Setup

Firstly, the ”id” property which must be a valid integer which is unique among all users. This property is utilized across the entire repository bot for the bot and for analysis and lets the research uniquely identify each run.

Secondly, the ”username” (string), ”email” (string), ”password” (string), ”dob” (string) and ”sex” (string) variables, form the google profile that is linked to this user. This information is critical to the user persona and permits not only the bot to login into YouTube as a real user with a real google account but also forms the basis of the simulated user defining its sex and age. It is also notable to mention that the a google account needs to be created, prior to running the simulation with this user, for which the ”username”, ”email”, ”password”, ”dob” and ”sex” variables are actually correct. This ensures that the bot works correctly.

Thirdly, the ”interests” (list of strings) sets the user interests. These interests define which categories of videos are relevant to the user (and therefore will be reacted to with a positive reaction) and which videos are irrelevant (and therefore will be reaction to with a negative reaction). These interests can be anything so long as they are valid strings and correlate in some way to video categories.

Finally, the ”reactions” (list of reactions) variable defines how the bot will react to videos. The ”reactions” variable must contain all the reactions with their probability of actually occurring in the list. The

”probability” variable for each reaction needs to be a double between 0 and 1 inclusive with 0 indicating a 0% chance of the reaction actually happening and 1 indicating a 100% chance of the reaction actually happening. Setting the ”probability” variable to 0 essentially means that the corresponding reaction is not being tested for its importance in impacting the algorithm. This setup permits the user simulation bot to test any of the 5 reactions it is coded for in anyway that is desired.

It is important to note that for now the bot can only perform these 5 actions (watch, like, share, skip, dislike) but it can be easily modified to add other functionality such as subscribing, commenting or pressing ”do not recommend”. The reasoning behind not implementing this functionality is that the research wanted to focus on direct user actions. Subscribing, commenting or pressing ”do not recommend” are all actions which require a higher amount of user effort to perform then the actions investigated here. It would definitely be of interest for future research to modify the user simulation tool in order to perform analysis of these user actions, but it is not the focus of this research for now as the scope is limited to direct easy to perform, one click/swipe, actions.

Therefore, the example user persona shown in figure 3.13 shows a 24 year old Female user with interests in Pets, Fashion, Health and Fitness, Beauty and Travel. The user will therefore watch 100% of the videos presented which are relevant to these interests and has a 50% chance of liking them as-well. It will also skip 100% of irrelevant videos.

Now that the user(s) with its run configuration has been defined we can proceed to defining the last few variables in the Scroller-AnalyzerV2.py file. Particularly the run that will actually be performed needs to be defined as well as the llm model of choice. Figure 3.14 demonstrates these two variables.

The ’model’ (string) defines which llm model the bot will use to perform the classifications and decision making. There are several options including all the models from Open-AI (which cost money) as well as some local models such as LLama3 (which require powerful computer) and it is up to the researchers to decide which model to use however performance and cost will vary across the models.

Finally, the ’runs’ (list of run objects) defined which user_ids the model will use to perform the run and how many videos each user_id will watch in that run.

```
model = 'gpt-4o'


runs = [
    {'user_id': 28, 'number_videos_to_watch': 450},
    {'user_id': 29, 'number_videos_to_watch': 1000}
]
```

Figure 3.14: LLM and Runs Variables Setup Example

3.4.2. Setup used for this research

User Persona

As mentioned previously, the focus/scope of this research is on how easy to perform user actions impact the recommendations of the algorithm over time. The focus is therefore not on extreme content rabbit holes and the actual content of the rabbit holes formed is not of interest. Therefore the only requirement for the user personas used in the research is that they are realistic. To maintain consistency across all results the same user with the same interests was used for each one of the runs. The user had the following profile:

Attribute	Details		
Sex	Male		
Age	29 years old		
Interests	Technology	Science	Engineering
	Mathematics	Education	Pets
	Motorcycles	Cars	Sports

The actual specific interests of the user is not important as the aim of the research is not to find out how the YouTube Shorts algorithm feeds specific content categories but to overall understand which reactions affect it and if indeed rabbit holes are developed. In this case the user represents a simple 29 year old male with basic male interests.

Run configurations

In order to analyze the feature importance of watching, liking, sharing, disliking and skipping several runs where made by the bot with different setups. In total 14 runs were preformed and kept for analysis. Their specific reactions are presented below:

Table 3.3: Reaction Summary Table

Run	Name	Positive Reactions	Negative Reactions	Probability
1	100percentskipper	\	\	\
2	10percentwatch	Watch	\	10%
3	25percentwatch	Watch	\	25%
4	50percentwatch	Watch	\	50%
5	75percentwatch	Watch	\	75%
6	100percentwatch	Watch	\	100%
7	10percentlike	Like	\	10%
8	25percentlike	Like	\	25%
9	50percentlike	Like	\	50%
10	75percentlike	Like	\	75%
11	100percentlike	Like	\	100%
12	100percentshare	Share	\	100%
13	100percentdislike	\	Dislike	100%
14	100percentall	Watch, Like, Share	Dislike	100%

The process for selecting these 14 runs went as follows. Firstly it was important to provide a baseline for the performance of the algorithm by performing a control run. In this run the user would provide as little feedback as possible to the algorithm so that subsequent runs could be compared to it to see if they meaningfully impacted the recommendations. Therefore in this run the user simply skips every video and does not react in any other way to videos, be they relevant or irrelevant to the user interests. This is done with run 1.

Then, one run was made with a 100% probability for each reaction. This comprises of runs 6, 10, 11, 12 and 13. This provided a good understanding of the impact of each action on the algorithm. These runs were then compared to the control run (run 1). If the impact was deemed to be significant then we would perform additional runs with different probabilities for the same reaction. In this way we can compare the impact of an action if it is performed on 25% of relevant videos compared to 100% of relevant videos.

As shown in Chapter 4 only the like and the watch actions significantly impacted the algorithm when compared to the control so therefore we only ran extra runs with lower probabilities of the user reacting to relevant videos with these two actions which resulted in runs 2, 3, 4, 5, 7, 8, 9, 10.

Finally run 14 was performed where all user actions were performed so as to assess how strongly a user can influence the algorithm. It is of note that the default behavior of the bot is to skip all videos unless the watch reaction is setup. In this way we are able to test the effect of every single reaction on the algorithm without providing unintentional feedback to the algorithm by watching videos.

4

Results

The results section firstly presents a time series analysis of the percentage of relevant videos feed to the user by the YouTube shorts recommendation algorithm. Each time series represents a ”run” made by the scroller & analyzer bot with a specific user profile, which determines if videos are relevant or irrelevant to the user interests, as well as what the specific reactions are.

Each run comprises of a few hundred or more video counts and demonstrates the effect that each reaction has on the output of the YouTube shorts recommendation algorithm, and more specifically how well the algorithm is able to learn the preferences of the user based on the reactions.

The results then aims to demonstrate if rabbit holes are formed by analyzing the specific video categories feed to the user by the YouTube shorts algorithm as a time series.

Finally the results section will conclude on which reactions are the most important in determining the output of the YouTube Shorts algorithm and therefore how to act to escape a rabbit hole.

4.1. Importance of Reaction

4.1.1. Skip (control run)

To be able to understand the effect that each reaction has on the output of the algorithm, it is necessary to first have a control dataset where no reactions were performed at all. Therefore, the first run that was performed was with a user who skipped all the videos presented to him, no matter if they were or interest or not and did not react in any other way.

Figure 4.1 clearly shows that with the specified interests of our user, the running average frequency of relevant videos hovers around 25% the entire time. This proves that if the user does not provide any reactions to the algorithm, it has little way of personalizing content and increasing the relevant video frequency.

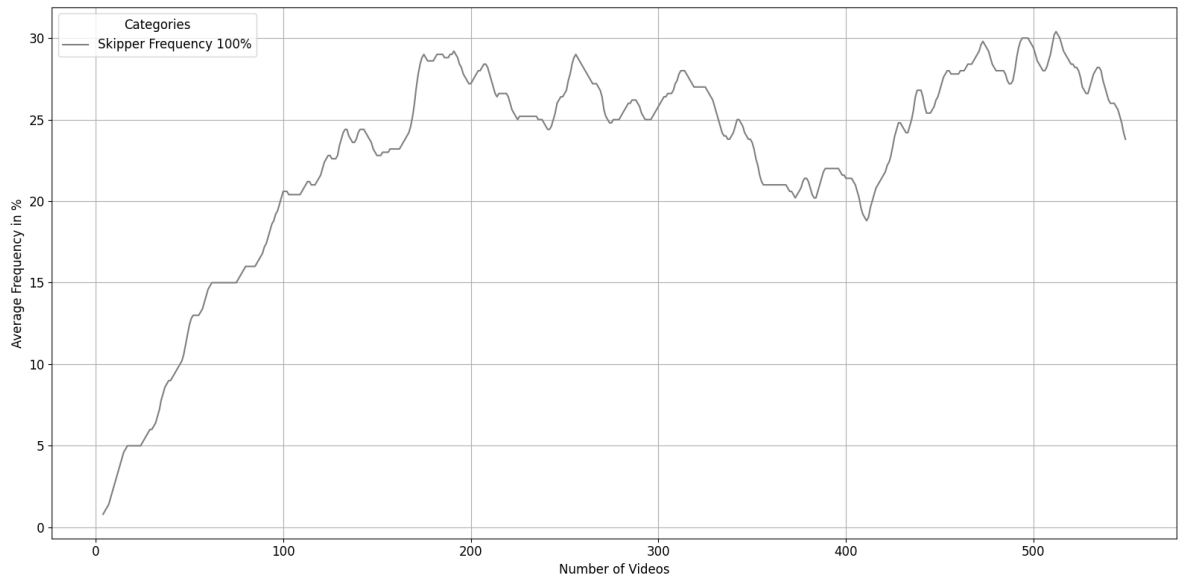

Figure 4.1: Running Frequency of relevant videos with skip reaction (control run)

4.1.2. Watch

In this section we will present the data for the watch reaction of the user. Figure 4.2 presents the average running frequency of relevant videos over the last 100 videos, while figure 4.3 shows the total percentage of relevant videos for all the watch reactions.

Figure 4.2: Running Frequency of relevant videos with watch reaction

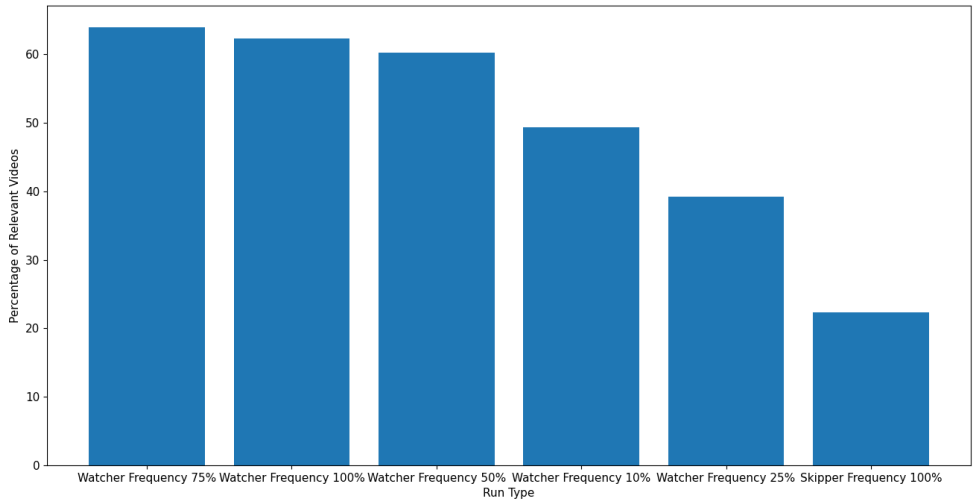

Figure 4.3: Percentage of relevant videos with the watch reaction

Observations

As can be seen in figure 4.2 the higher the probability of the bot actually watching a video of interest the higher the average running frequency of relevant videos. Indeed the 10% (blue) and 25% (yellow) runs consistently have the lowest average running frequency and stay like so the entire length of the run.

In contrast the 50% (green) The 75% (purple) and 100% (red) runs' average running frequencies continuously intersect each other over the length of the run suggesting that above 50% the watch reaction

significantly impacts the YouTube algorithm. These three runs manage to obtain and maintain an average running frequency of around 70% indicating that 70 of the last 100 videos shown to the user were of interest. This is significantly higher than the control ”skipper” run (grey) and demonstrates the high effect that watching has on determining the output of the YouTube algorithm.

To further demonstrate the correlation between the probability of watching a video of interest and the algorithm providing more videos of interest, figure 4.3 clearly shows that the higher the probability of watching a video of interest the higher the percentage of total videos of interest is.

However it is notable that the run with a probability of 75% actually has a slightly higher percentage than the run with a probability of 100%. This could be due simply to randomness or because of the YouTube algorithm trying to diversify the user's feed if he watches to many of the same videos.

Insights

It is clear that the watch reaction has a clear impact on the algorithm as even the runs with lower probabilities (10% and 25%) have a significantly higher percentage of relevant videos than the control ”skipper” run. Furthermore, the higher the probability of reaction the higher the percentage of relevant videos however the increases seem to reduce for probabilities than higher50%.

4.1.3. Like

In this section we will present the data for the like reaction for the user. Figure 4.4 presents the average running frequency of relevant videos over the last 100 videos, while figure 4.5 shows the total percentage of relevant videos for all the like runs.

Figure 4.4: Running Frequency of relevant videos with like reaction

Observations

Figure 4.4 does indeed demonstrate that the like reaction affects the outcome of the algorithm. However it is also clear that only the 75% (violet) and 100% (red) probabilities actually affected the algorithm in any significant way, both achieving an average running frequency of around 50% after watching a thousand videos.

The runs at 10% (blue), 25% (yellow) and 50% (green) do not seem to have any impact on the YouTube Shorts algorithm as their average running frequency is comparable to the control ”skipper” run (grey). This suggests that liking videos is only significant if done very consistently.

It is also of interest to mention that the average running frequency of all the runs for the like reaction seem to have high variability indicating that the algorithm seems to get confused on what the user actually is interested in and can rapidly change the output of the algorithm based on outputs.

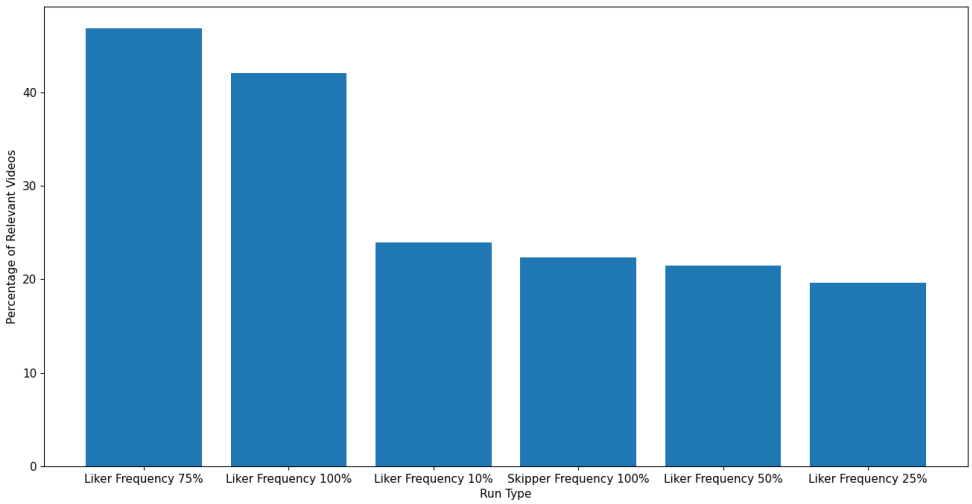

Figure 4.5: Percentage of relevant videos with the like reaction

To further demonstrate the importance of consistently liking videos if the user wishes to tune the algorithm on his preferences, figure 4.5 clearly shows that the higher consistency runs (75 and 100 percent) have around double the percentage of relevant videos throughout their runs than the lower percentages. It seems that the below liking only 50% or less of videos of interest has no impact on the algorithm as the percentages of relevant videos for these runs are within margin of error of the control ”skipper” run. Similarly to the watch reaction, the run with 75% probability has a higher total percentage of relevant videos.

Insights

Liking videos clearly has an impact on the algorithm however the consistency of these likes is crucial in the determining if the impact is significant or not. Indeed liking videos with a 75% probability or higher results in a average running frequency of around 50%, more than double the average running frequency of the control ”skipper” run. In contrast, a liking probability of 50% in lower seems to have no significant impact on the algorithm when compared to the control ”skipper” run with each run finishing at an average running frequency of around 20%.

4.1.4. Share

In this section we will present the data for the share reaction for the user. Figure 4.6 presents the average running frequency of relevant videos over the last 100 videos.

Observations

From figure 4.6, it is clear that the Share reaction does not fundamentally modify the output of the algorithm. Indeed the average running frequency of the share run at 100% (blue) is very similar to that of the control ”skipper” run (grey). The lines continuously criss-cross each other and eventually end at a similar average running frequency of 23-24%.

Insights

Even at 100% probability the share reaction had no significant impact on the output of the algorithm and therefore, can be determined as statistically insignificant Hence, it was decided not to execute extra runs with lower probabilities for the share reaction as the results would have been similar.

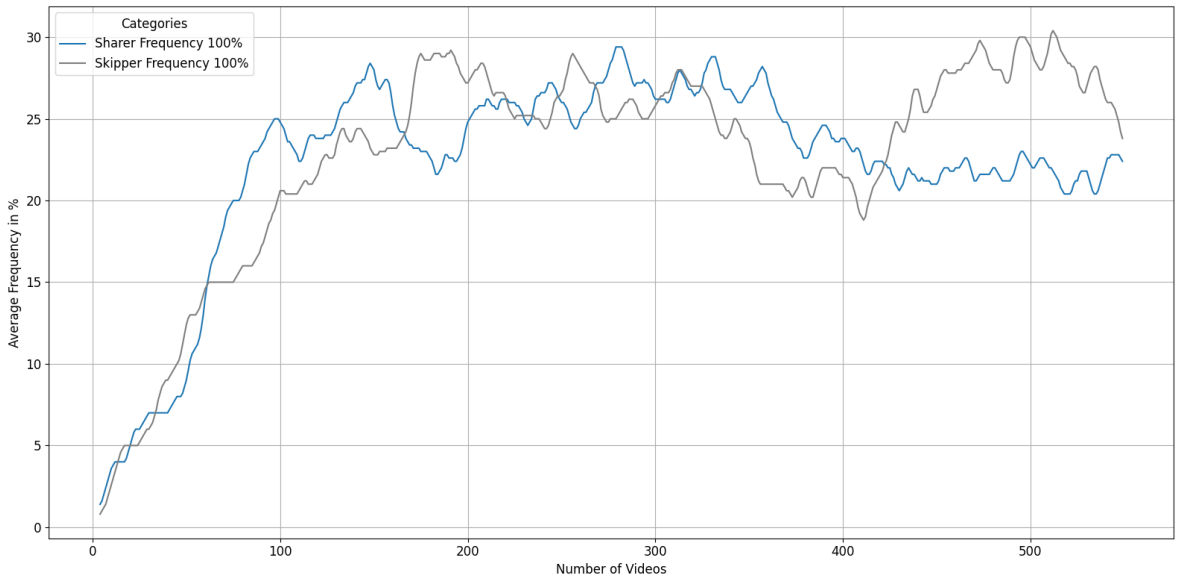

Figure 4.6: Running Frequency of relevant videos with share reaction

4.1.5. Dislike

In this section we will present the data for the dislike reaction for the user. Figure 4.7 presents the average running frequency of relevant videos over the last 100 videos.

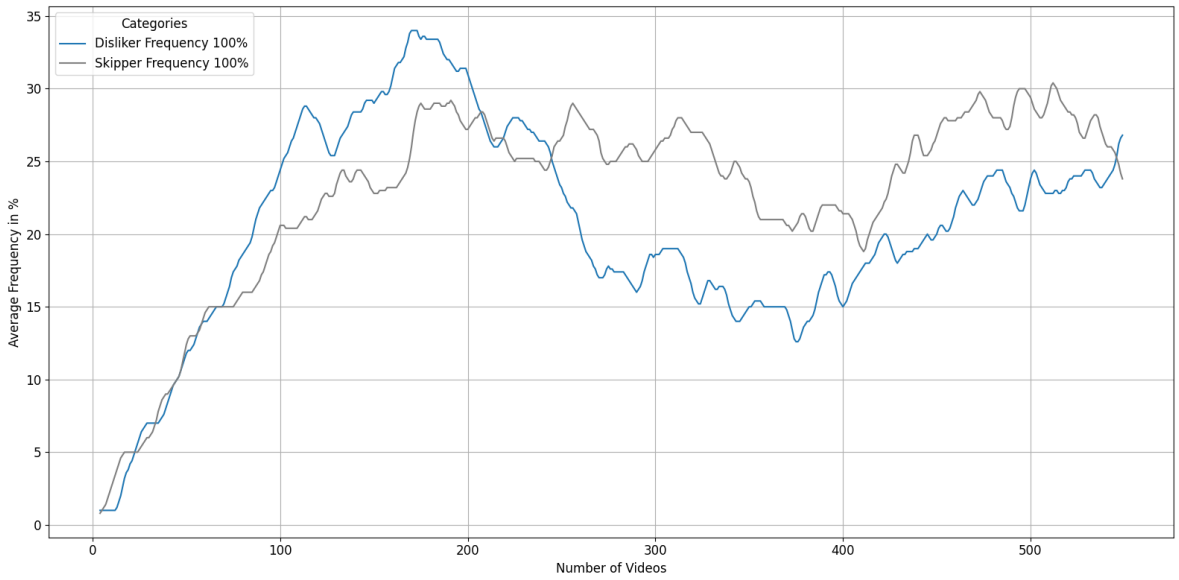

Figure 4.7: Running Frequency of relevant videos with dislike reaction

Observations

Similarly to the share reaction, figure 4.7 demonstrates that the Dislike reaction does not fundamentally modify the output of the algorithm. While the dislike run (blue) initially has a higher average running frequency than the control ”skipper” run (grey) the lines than reserve in their order and the control ”skipper” run actually maintains a higher average running frequency. However both of these frequencies are within margin of error and actually end up at around the same running frequency of 25%.

Insights

Even at 100% probability the dislike reaction had no significant impact on the output of the algorithm and therefore, similarly to the share reaction, it was decided not to execute extra runs with lower probabilities as the results would have been the same.

Similarly to the share reaction, the dislike reaction seems to be statistically insignificant in impacting the output of the YouTube Shorts algorithm. Therefore, it was decided not to execute extra runs with lower probabilities for the dislike reaction as the results would have been similar.

4.1.6. All Actions

In this section we will present the data for the run that performed all possible reactions for the user. Figure 4.8 presents the average running frequency of relevant videos over the last 100 videos.

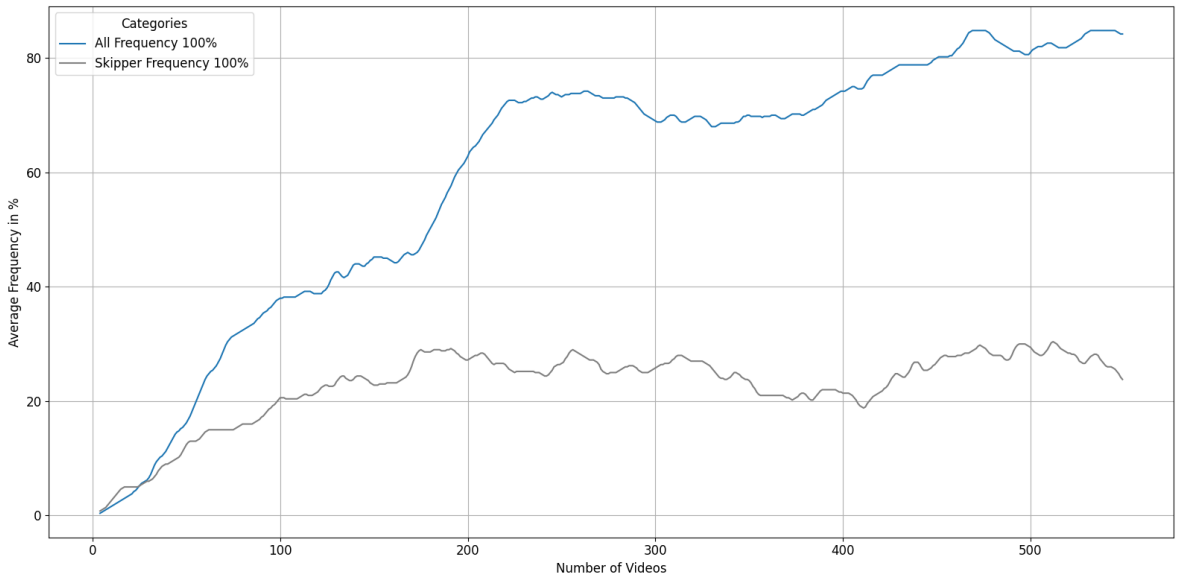

Figure 4.8: Running Frequency of relevant videos for all reactions types

Observations

As shown in figure 4.8, the run where the bot performs every reaction has a very strong effect on the algorithm. Indeed, while the control ”skipper” run stabilizes between 20% and 25% average running frequency. The ”All” run’s average running frequency continuously grows throughout the run and achieves a maximum average running frequency above 80% signalling the highest effect of any of the runs on the output of the YouTube Shorts algorithm.

Insights

4.1.7. Comparison of actions

After presenting each individual reaction in the above subsections, all the reactions can now be compared. Figure 4.9 shows the average running frequency for each reaction type at 100% chance, while figure 4.10 shows the total percentage of relevant videos for all the runs.

Observations

From Figure 4.9 we can observe the importance of each type of reaction on affecting the personalization of the algorithm to the user preferences. The lines on the plot clearly demonstrate that by far and away the watch (purple) reaction is the most important individual reaction that a user perform to influence the algorithm. Indeed the maximum average running frequency of the watch reaction, around 75%, is much greater than the next most important reaction, the like (green), which achieves a maximum of only approximately 50%.

Furthermore the like reaction has quite some variability with the curve of the average running frequency with it going quite significantly downwards after its peak and ends up at 35% towards the end of the

Figure 4.9: Running Frequency of relevant videos for runs at 100%

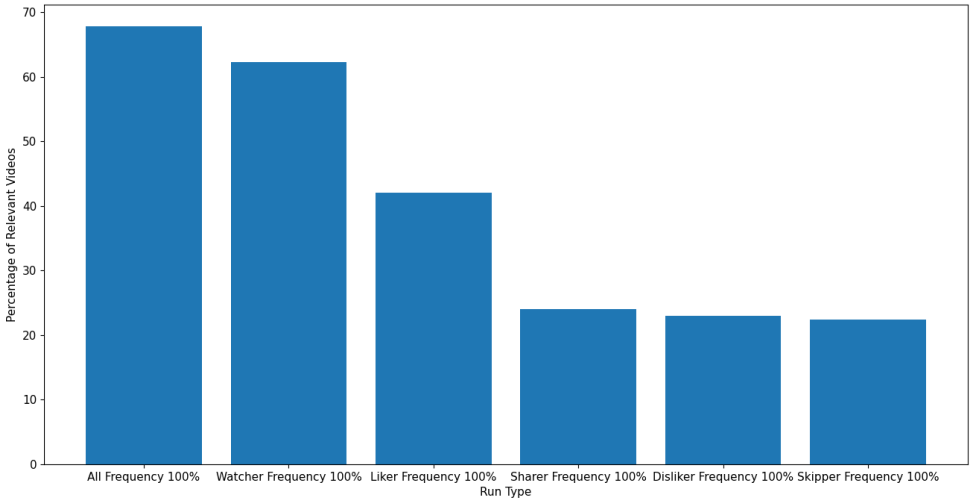

Figure 4.10: Percentage of relevant videos for all reaction types

run. In contrast the watch run maintains its growth throughout the run and finishes very close to its maximum. This indicates that the watch reaction produces not only more positive video reactions but that the algorithm also is able to improve these results as the number of videos watched increases.

When it comes to the other two individual reactions, the share (red) and the dislike (yellow), figure 4.9 clearly demonstrates that they do not have any significant impact on the output of the algorithm with their being constantly intertwined with each other and the control ”skipper” run (grey).

Finally when observing the ”all” run (blue) it is clear that it is by far and away the most effective at producing the most relevant videos. The average running frequency of the ”all” run grows very rapidly and although the average running frequency of the watch run (purple) does manage to get close (and a for short while) surpass the ”all” run, the ”all” run maintains a commendable lead, growing throughout the run and finishing at above 80%.

Figure 4.10 further underscores the observations of the average running frequencies. Indeed the percentage of relevant videos for the Share and Dislike runs are within margin of error of the control ”skipper” run showing the insignificant impact of these two actions on the output of the algorithm. Furthermore it underscores the importance of watch compared to liking as the watch run has nearly double the impact compared to the run in terms of net percentage increase (38% for the watch run compared to 17% for the like) when compared to the control ”skipper” run. Finally, it is clear that the ”all” run produces the best results in terms of total percentage of relevant videos throughout the run.

Insights

From the observations above a few key insights can be extracted. Firstly it is clear that sharing and disliking doesn't have any direct impact on the output of the algorithm. Secondly by far and away watch is the single most important individual reaction when determining the output of the YouTube Shorts algorithm. Liking does have an impact, although its impact is only half of the watch, but it still is significant and this is exemplified in the ”all” run which maintains a higher positive reaction frequency than any other run signifying that the more user inputs the YouTube algorithm has the better it is able to curate the algorithm to the user interests.

4.1.8. Summary of Key Findings: Key user actions

The objective of this section of the results is to understand the influence of different user reactions on the YouTube Shorts algorithm. To achieve this an automated tool powered by AI was utilized to simulate a user with specific interests and react to videos of (dis-)interest in a specific controlled manner. By performing several of these ”runs” each with specific reaction types, we obtained a collection of time-series plots which show the average running frequency of positive video reactions as a function of number of videos watched and reacted to. The analysis of these plots revealed many significant insights into how these reactions shape the personalization of users content feed on YouTube Shorts.

Firstly, a control run needed to be established to be able to compare the results of the reactions. As the ”skipper” run didn't provide any information to the YouTube Algorithm, the average running frequency consistently hovered around 25%, highlighting the algorithm's dependence on user interactions for personalized content.

When users engaged by watching videos, the algorithm showed a marked improvement in delivering relevant content. The data demonstrated that higher probabilities of watching videos of interest led to a substantial increase in the average running frequency of relevant videos. Runs with probabilities of 50%, 75%, and 100% consistently performed better, with the highest achieving a 70% positive reaction frequency. This underscores that watching videos is a potent signal for the algorithm, allowing it to refine its recommendations effectively.

In contrast, the like reaction, while impactful, showed that its influence depended heavily on the consistency of engagement. Only runs with a 75% or higher likelihood of liking videos saw a significant improvement in positive reaction frequencies, reaching around 50%. Lower probabilities had negligible effects, aligning closely with the control run. This suggests that sporadic likes do not provide the algorithm with sufficient data to tailor content effectively.

The share reaction, even at a 100% probability, did not significantly alter the algorithm's output. The running frequency of relevant videos for this action remained similar to the control run, indicating that sharing content does not play a critical role in personalizing video recommendations. Similarly, the dislike reaction showed no meaningful impact on the algorithm, as its results were indistinguishable from the control run. These findings suggest that negative feedback and content sharing are less influential in shaping the algorithm's behavior.

The most compelling results emerged from the run where the user performed all possible reactions. This comprehensive engagement led to the highest improvement in content relevance, with the average running frequency of relevant videos surpassing 80%. This run demonstrated that the combined input from watching, liking, sharing, and disliking videos allows the algorithm to finely tune its recommendations, maximizing user satisfaction.

In conclusion, the results reveals that active engagement, particularly through watching and consistently liking videos, significantly enhances the YouTube Shorts algorithm's ability to deliver relevant

content. While sharing and disliking have minimal direct impact, comprehensive user interactions amplify the algorithm's effectiveness. These insights emphasize the importance of user behavior in driving personalized content experiences on digital platforms.

4.2. Analysis of Rabbit holes

Now that the user actions influence on the algorithm have been thoroughly analyzed; the research analysis turns to understanding the prevalence and formation of rabbit holes on YouTube. To do this, the runs with high percentages of relevant videos will be analyzed based on the video categories shown to the user.

The runs that were selected are presented bellow:

Run Name	Relevant Videos (%)
All Frequency 100%	67.80
Watcher Frequency 75%	63.90
Watcher Frequency 100%	62.34
Watcher Frequency 50%	60.20
Liker Frequency 75%	46.83
Liker Frequency 100%	42.04

Table 4.1: Runs analyzed for rabbit holes

In this section we will present the running frequencies of the video categories, which aligned with the user interests, for each run. In this way we will be able to observe if any video category dominates or not and how the categories of the videos presented to the user change over time.

As a reminder, our user encompasses a wide range of categories so if the YouTube Shorts Algorithm does not push users into rabbit holes than multiple categories should be represented on the graphs and no single category should dominate the feed.

Below the figures are presented and from them we can indeed see that a single category does tend to dominate. Remarkably this category is the same in each of the runs that being the sports category. Each one of the figures demonstrates the strong tendency for the YouTube algorithm to nearly completely personalize the feed to one category. In particular the 100% Watcher Run, shown in Figure 4.13 demonstrates that across the entire run the YouTube algorithm nearly only showed sports videos to the bot encompassing 81.06% of relevant videos.

From the figures, it is evident that the YouTube algorithm tends to personalize the feed heavily towards the sports category, regardless of the variation in the user's interaction frequency. This trend suggests that the algorithm pushes users into a specific content rabbit hole, contradicting the expectation of a diverse feed when a wide range of categories aligns with user interests. The sports category's overwhelming dominance in each run demonstrates the algorithm's strong influence in shaping the content presented to users.

Nevertheless, there are some other points to notice from the figures presented. For example the 50% (Figure 4.14) and 75% (Figure 4.12) watcher runs have much more varied feeds when compared to the 100% watcher (Figure 4.13). Furthermore, as demonstrated in the previous section, the watch reaction is the most important when determining the output of the YouTube Shorts algorithm. Combining these two facts we can see that when the reactions are less consistent it directly shows to the algorithm that the user might have other interests / fatigue with seeing the same content over and over again.

Furthermore, figures 4.15 and 4.16 show that while the like action can form rabbit holes with both 100% and 75% like runs having one dominating category it is not as strong as for the watch plots. Finally, the all run presenting in figure 4.11 shows that a double rabbit hole was formed.

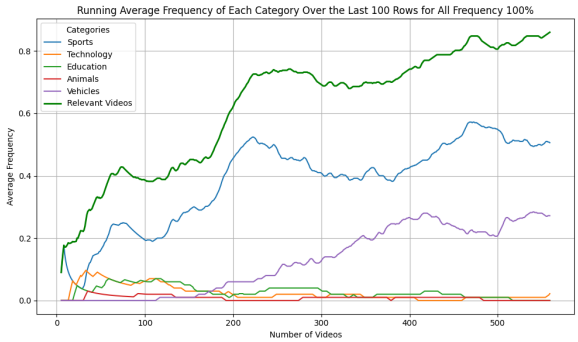

Figure 4.11: Average Running Frequency of Positive Categories: All Frequency 100%

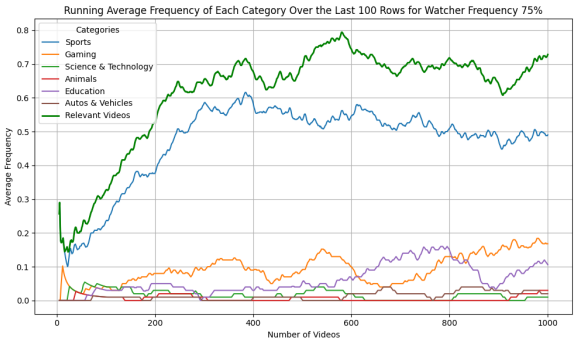

Figure 4.12: Average Running Frequency of Positive Categories: Watcher Frequency 75%

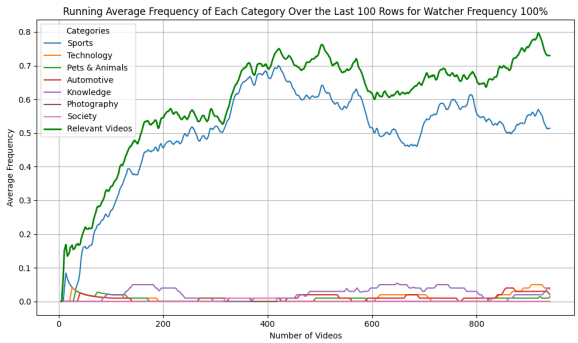

Figure 4.13: Average Running Frequency of Positive Categories: Watcher Frequency 100%

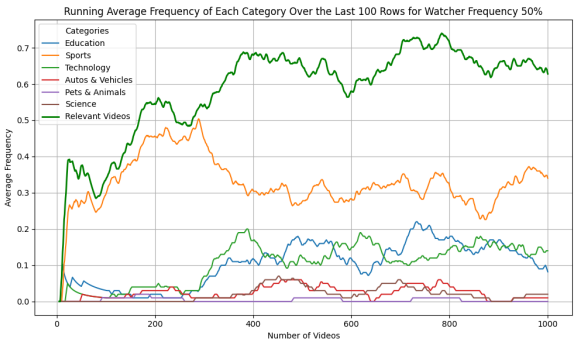

Figure 4.14: Average Running Frequency of Positive Categories: Watcher Frequency 50%

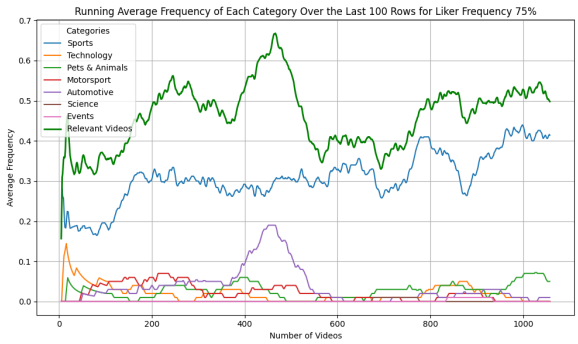

Figure 4.15: Average Running Frequency of Positive Categories: Liker Frequency 75%

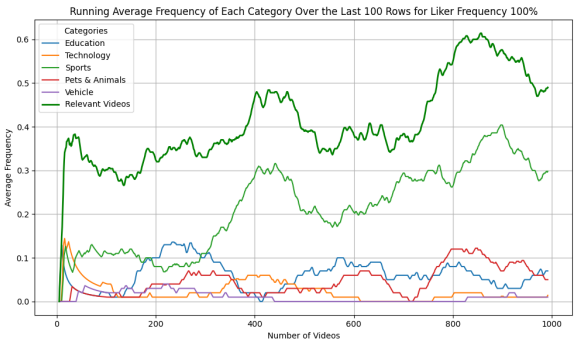

Figure 4.16: Average Running Frequency of Positive Categories: Liker Frequency 100%

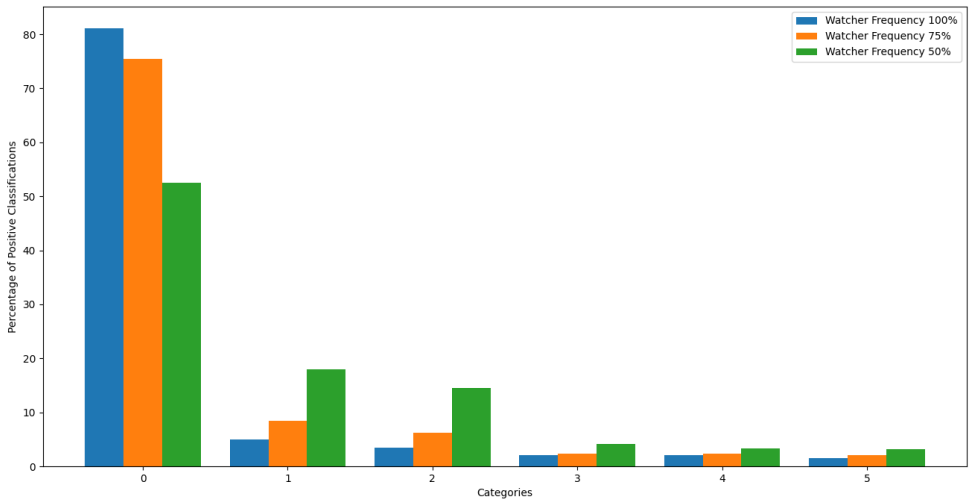

Figure 4.17: Spread of reactions among top 6 categories for each watcher

Indeed when comparing the total category spread across the watch reactions in Figures 4.17 there is a direct correlation between the spread of positive categories and the consistency of relevant videos. The more the bot consistently likes the more narrow the field of interest shown to the user is, indeed for the watcher with 100% probability, the most popular category shown to him represents more than 80% of total ”positive” videos while the next most popular category is only 5% of positive videos. When comparing this to the watcher with 50% probability, its most popular category only represents a bit over 50% of total ”positive” videos, while the next most popular category is a much healthier 20%. This observation appears to confirm the fact that consistency in watching is key in the formation of rabbit holes. Furthermore, it can be observed that the 75% watcher sits nearly perfectly in the middle of the previous two observations, further substantiating the correlation between the consistency of the watch reaction and the formation of rabbit holes.

5

Discussion

The discussion section will firstly focus on interpreting the results presented in chapter 4 in order to answer the two sub-research question and the main research question. The implications of the findings will then be laid out, both in the context of contributions to the scientific community as well as policy recommendations to reduce the formation of rabbit holes on SFV Platforms. Finally, the discussion section will explore the limitations of the study and ethical considerations.

5.1. Interpretation of Results

In this section we will interpret the results in order to answer all three sub and main research questions.

5.1.1. First sub-research question

The main issue with previous methods of algorithm auditing was the lack of full human behavior simulation. Indeed previous methods as seen in [24, 16, 27] did not simulate active decision making of users on how to react to content. Therefore an innovative auditing method was required in order to perform an audit on how the algorithm reacts to user actions on specific content.

As a reminder the first sub-research question went as follows:

How can a tool be built to collect a large dataset of algorithmic recommendations based on user actions?

As seen in the 3 Chapter, the development of this tool followed three steps. The first was to identify the key user actions and user loop that a human engages in when consuming content on YouTube Shorts. This flow diagram was then adapted to fit an automated simulation tool. Following the outline of the flow diagram, components were developed in order to simulate user personas, their decision making process and their actions.

The main components of the tool are as follows:

User Persona: The User Persona component serves as the foundation for simulating diverse user behaviors. It is defined by a set of interests that guide the bot's interactions with content. The persona determines which actions—such as watching, liking, disliking, or sharing—will be performed in response to videos classified as relevant (positive) or irrelevant (negative). This allows the simulation to reflect varied user engagement strategies, mirroring the diverse preferences and interaction patterns of real users.

Classification and User Decision-Making: This component relies on metadata retrieved from the YouTube API to categorize content. A Large Language Model (LLM) is employed to perform text classification based on this metadata, effectively simulating a user's interpretation of the content. The LLM then cross-references the video classification with the predefined user interests, enabling the bot to make decisions on whether to engage positively or negatively with each video. This process ensures that the simulation accounts for the nuanced and context-dependent nature of human decision-making.

User Actions: The User Actions component is designed to simulate the full spectrum of physical interactions a human might have with YouTube Shorts. By leveraging the Playwright web automation framework, the tool is capable of mimicking real user behaviors, such as scrolling through videos, watching content, liking or disliking videos, and sharing them. This automation ensures that the bot accurately replicates the actions a human user would take, allowing for consistent and repeatable simulation of user interactions over extended periods. The realism of these simulated actions is crucial for understanding how specific behaviors influence the platform's recommendation algorithm.

Data Collection: The Data Collection component is responsible for systematically logging each interaction the bot performs, capturing a linear sequence of actions alongside the corresponding video metadata and classification results. To maintain the integrity and reliability of the collected data, a SQL database is used for storage. This database structure ensures that all interactions are recorded accurately, preventing data loss and facilitating detailed analysis. The linear logging of actions provides a clear timeline of user interactions, enabling a comprehensive understanding of the algorithm's response to different user behaviors over time.

By combining all these components together the tool is able to fully simulate a human user interacting with YouTube shorts and log the entire watch and action history of the user. This precisely answers the first sub-research question by producing a dataset of algorithmic recommendations that are based on the recorded user actions on previous content. This lets us fully understand how user actions on specific content impact what specific content the algorithm will provide to the user.

This tool is now publicly available on the following github page, `https://github.com/christophcoss/YouTube_Shorts_Algorithm_Audit_Tool`, for the scientific community to use/further develop in order to analyze more in depth the YouTube Shorts Algorithm. Furthermore a similar approach can be utilized to build a tool for other platforms such as Instagram Reels or TikTok.

5.1.2. Second sub-research question

After successfully building an automated tool to collect a large dataset of algorithmic recommendations based on user actions and answering the first sub-research question we can turn our attention to answering the second sub-research question:

Can this dataset be analyzed to understand the key user actions that lead to falling into a rabbit hole on YouTube Shorts?

The dataset obtained from running the bot build to answer sub-research question 1 is essentially a linear time series of videos with the following data points:

- video id
- video category
- positive or negative alignment to user interests
- remaining metadata

After running the bot multiple times, we obtained multiple datasets each with the same user persona but different actions being performed. As show in the 4 chapter, this permitted the research to compare and identify how different types of user actions influence the algorithm producing running average frequency plots of positive reactions.

It is clear that the most important reaction that lets the algorithm learn user interests is the watch action. Even when the bot only had a 50% chance of watching a video deemed ”positive” it still significantly outperformed all other individual user actions. The like action was also shown to have a strong influence of the output of the YouTube Shorts Algorithm however the other actions such as sharing and disliking didn’t seem to have much of an impact on the algorithm when compared to the control run with no user actions.

The research also explained which actions are most likely to produce rabbit holes. This was done by analyzing the data sets to produce average running frequencies of specific categories. If a single category was much more popular than others then that would indicate a rabbit hole.

Indeed after analyzing these plots it is clear that YouTube Shorts has a strong tendency of producing rabbit holes. Each run that influenced the algorithm showed one or two categories dominating the others. However, it is important to note that the runs that had higher probabilities of performing user actions on positive videos had much strong rabbit holes when compared to runs with lower user action probabilities. This signifies the importance of consistent user feedback in the formation of rabbit holes on YouTube Shorts.

5.1.3. Main research question

Now that both sub-research questions have been successfully answered, the insights gained can be combined to answer the main research question:

”How can the feature importance of user actions in algorithmic recommendation systems on Short-Form Video Platforms, such as YouTube Shorts, be analyzed to understand their role in leading users down a rabbit hole?”

The research conducted has demonstrated that it is indeed possible to analyze the feature importance of the YouTube Shorts recommendation algorithm to a significant extent, allowing for a deeper under-

standing of how user actions influence the content that is recommended. By designing and implementing the automated bot described in response to the first sub-research question, a robust dataset was generated that captured the intricacies of algorithmic responses to varied user behaviors. This dataset provided a comprehensive view of how different user actions, particularly watching and liking videos, guide the algorithm in shaping the recommendation stream. The tool effectively replicated the decision-making processes of real users, which was critical in accurately modeling and understanding the algorithm's behavior.

Through careful analysis of the data, as discussed in the second sub-research question, the key mechanisms by which users are directed into rabbit holes were identified. The most significant finding is the algorithm's propensity to increasingly narrow down the content recommendations when it detects consistent user engagement with specific categories of videos. This behavior was especially pronounced when the bot exhibited a high probability of watching or liking videos aligned with its predefined interests. In such scenarios, the algorithm progressively funneled content into more specialized and repetitive categories, creating the so-called rabbit holes. This pattern was evident across multiple bot runs, reinforcing the conclusion that consistent and targeted user actions significantly contribute to the formation of rabbit holes on YouTube Shorts.

The implications of these findings are far-reaching, particularly in the context of algorithm transparency and user control. While the research confirms that analysing the feature importance of the recommendation system is feasible, it also highlights the complexity and opacity of these algorithms. The results underscore the importance of developing methods for users and researchers to better understand and, ideally, influence the recommendation algorithms that shape their online experiences. Furthermore, the ethical considerations raised by this research cannot be ignored. The ability of algorithms to lead users into increasingly narrow content streams raises questions about user autonomy and the potential for manipulative practices in content curation.

In conclusion, this research has not only provided a method for recording and analyzing the YouTube Shorts recommendation algorithm but has also shed light on the underlying processes that drive users into rabbit holes. These insights pave the way for future research aimed at enhancing transparency and empowering users in their interactions with algorithmic recommendation systems on Short-Form Video Platforms.

5.2. Implications of Findings

In this section we will discuss the implications of the findings of this study in terms of contributions to the scientific body, policy recommendations for reducing rabbit holes on SFV Platforms and finally recommendations for future works based on this study.

5.2.1. Contributions to the Scientific Body

This research performed in this study has resulted in three main contributions to the scientific body.

1. **Advancement In Algorithmic Understanding:** The first contribution is that of providing a detailed analysis of how specific user interactions influence the YouTube Shorts algorithm. There isn't much previous works performed on understanding the YouTube Shorts algorithm specifically with most studies focusing on the main YouTube Platform itself. This study provides researchers with better knowledge on how the algorithm works and more specifically cements the tendency of recommendation systems, particularly YouTube's, of leading users down rabbit holes.
2. **Methodological Contributions:** The second key contribution is that of the auditing tool that was developed to perform the research. Few tools made to audit a SFV platform existed publicly prior to this research and the development and documentation of its design proves that it is possible to simulate users that react specifically based on the content the tool is shown. This level of simulation hasn't been achieved before and this leads into the last contribution.
3. **Foundation for Further Research:** The development of the tool and the findings obtained with it can serve as a foundation for further research. Many questions regarding the behavior of the YouTube Shorts algorithm are yet to be answered as will be discussed in section 5.2.3. However all these questions could be answered with the use of this tool and its modality and customization allows other researchers to tailor its behavior for specific issues. Furthermore, the documentation

of the design of the tool lets future researchers develop their own tool for other SFV platforms such as TikTok or Instagram Reels.

Given the novel algorithmic-audit methodology of this research, it is difficult to compare the results of this study directly with previous works. However, parallels can be drawn from Boeker et Al. (2022)'s study "*An Empirical Investigation of Personalization Factors on TikTok*" and this thesis [6]. Similarly to this thesis, Boeker et Al. (2022) focused on the algorithm of an SFV Platform, in their case TikTok, and they also found that watching the videos and liking strongly impacted the algorithm. Interestingly however they found that watching only had a marginally higher impact on the recommendations of the algorithm compared to liking. This is contrast to the results of this thesis that shows that watching is much more important then liking for the YouTube Algorithm. Interestingly, the results from Boeker et Al. (2022) also showed that the follow user action (akin to subscribing on YouTubeShorts) was the most influential in determining the output of the recommendation algorithm. Therefore, it would be of great interest to include the "subscribe" user action in future research performed with the Scroller&Analyzer on YouTube Shorts, in order to fully compare the algorithms of these two SFV Platforms.

5.2.2. Policy Recommendations: Reducing Rabbit Holes on SFV Platforms

One of the objectives of this study, as discussed in chapter 1, was to outline policy recommendations and guidelines for users to mitigate the development of rabbit holes on YouTube Shorts. Below we present several policy recommendations that would improve user control of algorithmic recommendations in an effort to limit rabbit holes.

Recommendations to regulators

Algorithmic recommendation regulation already broadly exists in the European Union. The Digital Services Act (DSA), implemented by the EU in 2024, imposes several regulations on algorithmic recommendations systems particularly on "Very Large Online Platforms"(VLOPs) which include platforms like YouTube Shorts and TikTok [23]. These regulations are based on three key points that are relevant to this thesis: transparency and explanation, user control and algorithmic accountability. Furthermore, the AI act, adopted by the EU in 2024 and set for implementation in 2026, contains further regulations for AI systems such as algorithmic recommendation systems [13]. However, as has been observed in this thesis, YouTube Shorts currently doesn't comply with these guidelines. As a result, the following recommendations are proposed for policymakers.

First, the need for enhanced **algorithmic transparency** is reinforced by the results of this study. Both the DSA and AI Act require platforms to provide users with clear insights into how recommendation systems function. However, as observed with YouTube Shorts, no explicit explanation is offered as to why a particular video is recommended. Users are left without clarity on how their actions influence the content they receive, which stands in direct contrast to the transparency requirements outlined in the DSA. Although users are able to delete their watch history, this measure does not provide any meaningful insight into how their behavior impacts future recommendations. Therefore, it is recommended that platforms be required to provide accessible explanations of how specific user interactions, such as likes or watch time, affect the recommendations shown. Providing such transparency would better equip users to engage critically with platform algorithms.

Second, the current lack of **user control** over YouTube Shorts' recommendation algorithm highlights another significant area where platform practices diverge from regulatory intent. While users can clear their entire watch history to "reset" the algorithm, this is far from an intuitive solution. It functions more as a workaround than a genuine tool for user empowerment, as it indiscriminately erases all historical data rather than allowing users to selectively adjust the algorithm based on their current preferences. To bridge this gap, platforms should implement more targeted tools that allow users to modify the recommendation system in real-time. For example, a simple chatbot interface could facilitate easier customization by enabling users to specify topics of interest or content they wish to avoid. This approach would allow for a more nuanced control mechanism, far exceeding the current all-or-nothing option of clearing one's entire watch history.

Finally, there is a pressing need to enhance **algorithmic accountability** on platforms like YouTube Shorts, particularly with respect to how recommendation systems are audited. Although the DSA and AI Act both stress the importance of ongoing audits and accountability, current platform implementations

fall short in offering users direct feedback or options for reviewing and adjusting how the algorithm interprets their preferences. The findings of this thesis demonstrate how users can become trapped in increasingly narrow content loops or ”rabbit holes,” with little opportunity to intervene or correct the algorithm’s course. To address this, it is recommended that platforms introduce user-friendly tools for personal algorithmic audits, enabling users to view and adjust their content streams in a more structured manner. Such tools could, for example, display a breakdown of the categories or themes that the algorithm prioritizes based on past interactions, offering users the opportunity to modify or reset these preferences as needed.

In summary, while the introduction of the DSA and AI Act has established a foundational framework for regulating algorithmic recommendation systems, further efforts are required to ensure platforms like YouTube Shorts are fully compliant. By improving transparency, enabling greater user control, and fostering enhanced accountability mechanisms, platforms can better align with regulatory goals, while simultaneously addressing the challenges identified in this study.

Recommendations to users

While these policies recommendations would go a long way to reduce rabbit holes and provide users with more control over their algorithmic recommendations; until they are implemented, and properly enforced alongside the DSA and AI act, the task of limiting the formation of rabbit holes currently falls in the hands of the individual user. Therefore, users who seek to limit rabbit holes on YouTube Shorts should follow these guidelines:

1. **Engage with diverse content:** Users should actively explore a wide range of content across different topics and genres to avoid narrowing their recommendations. This helps the algorithm recognize a broader set of preferences and prevents it from focusing too heavily on specific content categories.
2. **Vary your engagement behavior:** To avoid reinforcing repetitive recommendations, users should mix their interactions, such as liking, skipping, or simply watching to completion across different content types. This disrupts the feedback loop that drives narrow content exposure.
3. **Monitor your feed regularly:** Users should periodically review their feed to ensure a variety of content is being recommended. If the feed becomes too focused on one genre, users can reset the algorithm by deliberately engaging with new or different content.
4. **Avoid passive scrolling:** Mindless consumption reinforces the algorithm’s biases. Instead, users should consciously decide what to watch and skip, especially when noticing repetitive patterns in recommended content.
5. **Reset your watch history strategically:** If recommendations become too repetitive, users should consider clearing specific parts of their watch history to help reset the algorithm’s learning without wiping all data. This targeted approach can refocus the content being recommended.

On the contrary if users seek to curate their algorithm for specific type of content, forming an intentional rabbit hole, they should prioritize instantly skipping any videos that are not of interest and fully watching all videos that are. Liking the videos of interest could be beneficial as well but the main actions required are skipping and watching.

5.2.3. Future recommendations

Finally this research will outline recommendations of possible future research based on this work.

1. **Investigate importance of watch time:** While this study has outlined the importance of the ”watch” interaction, this was done by always watching the entirety of a video. Therefore it is unclear what percentage of a video needs to be watched for that interaction to be considered positive. Future research could modify the tool slightly to watch only a portion of the videos length and compare the results when watching 10%, 50% or 100% of the video. This would highlight how quickly a user needs to swipe up in order to indicate to the algorithm that they are not interested in that content type.
2. **Investigate other user actions:** While the study as focused on many user actions such as watching and liking, it has not investigated all the actions. Some of the actions left to investigate

are commenting, subscribing to a channel or utilizing the ”not interested” feature. These actions should be investigated as well in order to gain a clearer picture of the importance of different user actions and how users can better train their recommendation algorithm.

3. **Investigate algorithmic re-calibration:** One other aspect of the algorithm that has not been investigated is its ability to adapt to changes in user interests. Future research could utilize the tool to perform a run where the user interests change halfway through in order to investigate how quickly the algorithm is able to adapt to the new interests.
4. **Investigate other SFV platforms:** Finally, this study has investigated the YouTube Shorts platform and while it can be argued that other SFV platforms utilize similar recommendations sytems, it is important to investigate them directly. The processes used to develop the tool outlined in the 3 chapter can be applied to develop similar tools for other platforms and investigate them as well. This would further deepen the scientific communities understanding of SFV platforms' recommendation systems and enlighten better policy recommendations that would work across all recommendation systems.

5.3. Limitations of the Study and Ethical Considerations

While the outlined research approach presents a robust methodology for investigating algorithmic recommendation systems on SFV platforms, there are notable limitations and ethical issues to discuss.

Firstly, the runs themselves contain some limitations in that they were performed over a short period of time, May - June 2024, and therefore the results collected only reflect the algorithmic performance during that period of time. It is known that algorithmic recommendation systems are continuously updated and therefore the results presented in this study might not be accurate in the future. Furthermore, the runs were all performed with the same user persona which while providing consistency to compare the effects of different reactions on the algorithm, limit the research into how different personas and user interests affect the formation of rabbit holes.

Secondly, the use of LLMs for classifying and simulating user decision-making introduces accuracy, bias, and interpretability complexities. These models are known to occasionally hallucinate and can sometimes provide varying results given the same prompt twice. As discussed in Appendix A, the model of choice for this study, GTP-4o, provides 85% accuracy when classifying videos, which can make the tool react differently from how a human user would.

Additionally, the automated data collection tools used to simulate user behavior on YouTube Shorts, while effective, present challenges. These tools must be constantly updated and maintained due to frequent changes in the platform's interface, which can lead to inconsistencies in data collection. This variability could affect the reliability of the results over time, as the algorithm's responses might shift in ways that are difficult to capture accurately.

Content manipulation risks also emerge as a significant ethical consideration. The study's methodology, particularly in simulating user interactions, has the potential to influence the algorithm in ways that might unfairly impact content creators. Moreover, the research operates within a framework that lacks the explicit consent of the platform being audited. While external audits of this nature are common, they raise ethical questions about conducting research without the platform's knowledge or agreement.

Furthermore, the thesis focused on only five specific user actions—watch, like, share, dislike, and skip. While these actions were chosen for their ease of use and minimal effort required from users, there are other user actions that could also influence the algorithm, such as the ”do not recommend” button, subscribing to channels, or commenting on videos. These actions, while likely important in shaping algorithmic recommendations, were not included in the scope of this research due to their higher user cost in terms of time and effort. The study prioritized actions that are more frequently performed by users, allowing for a more practical understanding of how common interactions shape recommendation patterns.

Finally, this research did not specifically address the formation of extreme content rabbit holes, such as those leading to alt-right or ”red pill” communities. While these types of rabbit holes are significant and have been linked to broader concerns around polarization and radicalization, the focus of this thesis

was on understanding general rabbit hole dynamics on SFV platforms. Investigating the pathways that lead to extreme content would require a distinct approach and ethical considerations.

6

Conclusion

This research contributes to the expanding field of algorithmic auditing by investigating the influence of recommendation systems on short-form video (SFV) platforms, with a specific focus on YouTube Shorts. Through the development of a novel tool that simulates user behavior and gathers large datasets, this study uncovers critical insights into the ”rabbit hole” effect, wherein users are led to increasingly narrow content recommendations based on their interactions. By analyzing user actions such as watching, liking, disliking, and sharing, the research reveals how these behaviors shape the content delivered by YouTube Shorts' recommendation algorithm.

The main research question—”How can the feature importance of user actions in algorithmic recommendation systems on Short-Form Video Platforms, such as YouTube Shorts, be analyzed to understand their role in leading users down a rabbit hole?”—has been answered by demonstrating that watching and liking videos significantly influence the algorithm, leading to more focused content suggestions, while disliking or sharing have minimal impact. This prioritization of engagement metrics highlights the algorithm's role in amplifying content that aligns with user preferences, reinforcing immersion into specific categories. The study thereby confirms the existence of rabbit holes, which intensify with consistent user actions, raising concerns about the creation of echo chambers and filter bubbles that limit exposure to diverse viewpoints and may contribute to societal polarization.

From a scientific perspective, this research advances our understanding of how recommendation algorithms function, particularly on SFV platforms, and fills a gap in the study of algorithmic biases in short-form content curation. The introduction of a user-simulation tool provides a new method for auditing algorithms, offering broader applications for researchers investigating algorithmic transparency and accountability.

On a societal level, the implications are significant. The study underscores the risks associated with rabbit holes, including the potential for content radicalization and diminished public discourse. By demonstrating how user interactions can deepen algorithmic biases, the research suggests the need for greater regulatory oversight. Proper enforcement of the Digital Services Act (DSA) and the AI Act is critical to ensuring that platforms are held accountable for the content they recommend and to preventing algorithmic systems from exacerbating social division. Policymakers could leverage these findings to promote algorithmic transparency and ensure a more diverse range of content, helping mitigate the harmful effects of automated content curation.

Finally, this thesis aligns closely with the Complex Systems Engineering and Management (CoSEM) program's focus on addressing socio-technical challenges. The study exemplifies how interdisciplinary approaches—integrating technology, policy, human behavior, and ethics—are essential in managing complex systems like digital content platforms. By bridging technical insights with broader societal concerns, this work reflects CoSEM's core objective: to design sustainable, balanced solutions for large-scale socio-technical problems.

In summary, this research advances both scientific understanding and societal awareness of recommendation algorithms, while offering practical tools and insights for future audits. The developed tool, alongside the findings, lays a foundation for further exploration of algorithm-driven platforms and their broader impacts, providing valuable guidance for users, researchers, and policymakers alike.

References

[1] Renata Almachnee and Mary Cozzie. “Social Media and Dopamine: Studying Generation Z and Dopamine Levels”. en. In: *Journal of Student Research* 11.4 (Nov. 2022). ISSN: 2167-1907. DOI: 10.47611/jsrhs.v11i4.3649. URL: https://www.jsr.org/hs/index.php/path/article/view/3649 (visited on 12/20/2023).

[2] *API Reference | YouTube Data API*. en. URL: https://developers.google.com/youtube/v3/docs (visited on 12/20/2023).

[3] Jack Bandy and Nicholas Diakopoulos. *#TulsaFlop: A Case Study of Algorithmically-Influenced Collective Action on TikTok*. en. Issue: arXiv:2012.07716 arXiv:2012.07716 [cs]. Dec. 2020. URL: http://arxiv.org/abs/2012.07716 (visited on 09/19/2023).

[4] Pablo Barberá. “Social Media, Echo Chambers, and Political Polarization”. en. In: *Social Media and Democracy*. Ed. by Nathaniel Persily and Joshua A. Tucker. 1st ed. Cambridge University Press, Aug. 2020, pp. 34–55. ISBN: 978-1-108-89096-0 978-1-108-83555-8 978-1-108-81289-4. DOI: 10.1017/9781108890960.004. URL: https://www.cambridge.org/core/product/identifier/9781108890960%23CN-bp-3/type/book_part (visited on 09/13/2024).

[5] Nathan Bartley et al. “Auditing Algorithmic Bias on Twitter”. en. In: *13th ACM Web Science Conference 2021*. Virtual Event United Kingdom: ACM, June 2021, pp. 65–73. ISBN: 978-1-4503-8330-1. DOI: 10.1145/3447535.3462491. URL: https://dl.acm.org/doi/10.1145/3447535.3462491 (visited on 11/10/2023).

[6] Maximilian Boeker and Aleksandra Urman. “An Empirical Investigation of Personalization Factors on TikTok”. en. In: *Proceedings of the ACM Web Conference 2022*. arXiv:2201.12271 [cs]. Apr. 2022, pp. 2298–2309. DOI: 10.1145/3485447.3512102. URL: http://arxiv.org/abs/2201.12271 (visited on 09/19/2023).

[7] Lauren Valentino Bryant. “The YouTube Algorithm and the Alt-Right Filter Bubble”. en. In: *Open Information Science* 4.1 (June 2020). Number: 1, pp. 85–90. ISSN: 2451-1781. DOI: 10.1515/opis-2020-0007. URL: https://www.degruyter.com/document/doi/10.1515/opis-2020-0007/html (visited on 09/26/2023).

[8] Annie Y. Chen et al. “Subscriptions and external links help drive resentful users to alternative and extremist YouTube channels”. en. In: *Science Advances* 9.35 (Sept. 2023). Number: 35, eadd8080. ISSN: 2375-2548. DOI: 10.1126/sciadv.add8080. URL: https://www.science.org/doi/10.1126/sciadv.add8080 (visited on 11/10/2023).

[9] Matteo Cinelli et al. “The echo chamber effect on social media”. en. In: *Proceedings of the National Academy of Sciences* 118.9 (Mar. 2021). Number: 9, e2023301118. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.2023301118. URL: https://pnas.org/doi/full/10.1073/pnas.2023301118 (visited on 09/26/2023).

[10] Paul Covington, Jay Adams, and Emre Sargin. “Deep Neural Networks for YouTube Recommendations”. en. In: *Proceedings of the 10th ACM Conference on Recommender Systems*. Boston Massachusetts USA: ACM, Sept. 2016, pp. 191–198. ISBN: 978-1-4503-4035-9. DOI: 10.1145/2959100.2959190. URL: https://dl.acm.org/doi/10.1145/2959100.2959190 (visited on 08/02/2024).

[11] Carroll Doherty et al. “FOR RELEASE AUGUST 9, 2022”. en. In: ().

[12] Pınar Dündar and Heritiana Ranaivoson. “Science by YouTube: an Analysis of YouTube’s Recommendations on the Climate Change Issue”. en. In: *Observatorio (OBS*)* 16.3 (Sept. 2022). Number: 3. ISSN: 1646-5954. DOI: 10.15847/obsOBS16320222061. URL: https://obs.obercom.pt/index.php/obs/article/view/2061 (visited on 09/26/2023).

[13] *EU AI Act: first regulation on artificial intelligence*. en. Aug. 2023. URL: https://www.europarl.europa.eu/topics/en/article/20230601STO93804/eu-ai-act-first-regulation-on-artificial-intelligence (visited on 09/18/2024).

[14] Muhammad Haroon et al. "Auditing YouTube's recommendation system for ideologically congenial, extreme, and problematic recommendations". In: *Proceedings of the National Academy of Sciences* 120.50 (Dec. 2023). Publisher: Proceedings of the National Academy of Sciences, e2213020120. DOI: 10.1073/pnas.2213020120. URL: https://www-pnas-org.tudelft.idm.oclc.org/doi/10.1073/pnas.2213020120 (visited on 07/19/2024).

[15] *How to Make the YouTube Shorts Algorithm Work for You (2024)*. URL: https://riverside.fm/blog/youtube-shorts-algorithm (visited on 07/19/2024).

[16] Eslam Hussein, Prerna Juneja, and Tanushree Mitra. "Measuring Misinformation in Video Search Platforms: An Audit Study on YouTube". en. In: *Proceedings of the ACM on Human-Computer Interaction* 4.CSCW1 (May 2020). Number: CSCW1, pp. 1–27. ISSN: 2573-0142. DOI: 10.1145/3392854. URL: https://dl.acm.org/doi/10.1145/3392854 (visited on 09/26/2023).

[17] Daniel Klug et al. "Trick and Please. A Mixed-Method Study On User Assumptions About the TikTok Algorithm". en. In: *13th ACM Web Science Conference 2021*. Virtual Event United Kingdom: ACM, June 2021, pp. 84–92. ISBN: 978-1-4503-8330-1. DOI: 10.1145/3447535.3462512. URL: https://dl.acm.org/doi/10.1145/3447535.3462512 (visited on 09/19/2023).

[18] Tobias D. Krafft, Marc P. Hauer, and Katharina A. Zweig. "Why Do We Need to Be Bots? What Prevents Society from Detecting Biases in Recommendation Systems". en. In: *Bias and Social Aspects in Search and Recommendation*. Ed. by Ludovico Boratto et al. Vol. 1245. Series Title: Communications in Computer and Information Science. Cham: Springer International Publishing, 2020, pp. 27–34. ISBN: 978-3-030-52484-5 978-3-030-52485-2. DOI: 10.1007/978-3-030-52485-2_3. URL: http://link.springer.com/10.1007/978-3-030-52485-2_3 (visited on 11/10/2023).

[19] Erwan Le Merrer, Gilles Trédan, and Ali Yesilkanat. "Modeling Rabbit-Holes on YouTube". In: *Social Network Analysis and Mining* 13.1 (Dec. 2023). Publisher: Springer, p. 100. DOI: 10.1007/s13278-023-01105-9. URL: https://hal.science/hal-03620039 (visited on 07/18/2024).

[20] Alexander Liu, Siqi Wu, and Paul Resnick. *How to Train Your YouTube Recommender to Avoid Unwanted Videos*. en. Issue: arXiv:2307.14551 arXiv:2307.14551 [cs]. Aug. 2023. URL: http://arxiv.org/abs/2307.14551 (visited on 09/19/2023).

[21] Danaë Metaxa et al. "Auditing Algorithms: Understanding Algorithmic Systems from the Outside In". en. In: *Foundations and Trends® in Human–Computer Interaction* 14.4 (2021). Number: 4, pp. 272–344. ISSN: 1551-3955, 1551-3963. DOI: 10.1561/1100000083. URL: http://www.nowpublishers.com/article/Details/HCI-083 (visited on 11/10/2023).

[22] Jessica Ochmann, Sandra Zilker, and Sven Laumer. "The Evaluation of the Black Box Problem for AI-Based Recommendations: An Interview-Based Study". en. In: *Innovation Through Information Systems*. Ed. by Frederik Ahlemann, Reinhard Schütte, and Stefan Stieglitz. Vol. 47. Series Title: Lecture Notes in Information Systems and Organisation. Cham: Springer International Publishing, 2021, pp. 232–246. ISBN: 978-3-030-86796-6 978-3-030-86797-3. DOI: 10.1007/978-3-030-86797-3_16. URL: https://link.springer.com/10.1007/978-3-030-86797-3_16 (visited on 11/08/2023).

[23] Harsh Vardhan Pachisia. *The EU's Digital Services Act takes on "The Algorithm"*. en-US. Jan. 2024. URL: https://chicagopolicyreview.org/2024/01/01/the-eus-digital-services-act-takes-on-the-algorithm/ (visited on 09/18/2024).

[24] Kostantinos Papadamou et al. ""It Is Just a Flu": Assessing the Effect of Watch History on YouTube's Pseudoscientific Video Recommendations". en. In: *Proceedings of the International AAAI Conference on Web and Social Media* 16 (May 2022), pp. 723–734. ISSN: 2334-0770, 2162-3449. DOI: 10.1609/icwsm.v16i1.19329. URL: https://ojs.aaai.org/index.php/ICWSM/article/view/19329 (visited on 12/19/2023).

[25] Rahul Rana. *History of YouTube - How it All Began & Its Rise*. en-US. May 2024. URL: https://www.vdocipher.com/blog/history-of-youtube/ (visited on 07/19/2024).

[26] Ivan Sekulić et al. *Reliable LLM-based User Simulator for Task-Oriented Dialogue Systems*. en. arXiv:2402.13374 [cs]. Feb. 2024. URL: http://arxiv.org/abs/2402.13374 (visited on 10/05/2024).

[27] Larissa Spinelli and Mark Crovella. "How YouTube Leads Privacy-Seeking Users Away from Reliable Information". en. In: *Adjunct Publication of the 28th ACM Conference on User Modeling, Adaptation and Personalization*. Genoa Italy: ACM, July 2020, pp. 244–251. ISBN: 978-1-4503-7950-2. DOI: 10.1145/3386392.3399566. URL: https://dl.acm.org/doi/10.1145/3386392.3399566 (visited on 12/19/2023).

[28] Ivan Srba et al. "Auditing YouTube's Recommendation Algorithm for Misinformation Filter Bubbles". en. In: *ACM Transactions on Recommender Systems* 1.1 (Mar. 2023). Number: 1 arXiv:2210.10085 [cs], pp. 1–33. ISSN: 2770-6699. DOI: 10.1145/3568392. URL: http://arxiv.org/abs/2210.10085 (visited on 09/26/2023).

[29] Matus Tomlein et al. "An Audit of Misinformation Filter Bubbles on YouTube: Bubble Bursting and Recent Behavior Changes". en. In: *Fifteenth ACM Conference on Recommender Systems*. arXiv:2203.13769 [cs]. Sept. 2021, pp. 1–11. DOI: 10.1145/3460231.3474241. URL: http://arxiv.org/abs/2203.13769 (visited on 09/26/2023).

[30] Caroline Violot et al. "Shorts vs. Regular Videos on YouTube: A Comparative Analysis of User Engagement and Content Creation Trends". en. In: *ACM Web Science Conference*. Stuttgart Germany: ACM, May 2024, pp. 213–223. ISBN: 9798400703348. DOI: 10.1145/3614419.3644023. URL: https://dl.acm.org/doi/10.1145/3614419.3644023 (visited on 07/19/2024).

[31] Kaitlin Woolley and Marissa A. Sharif. "Down a Rabbit Hole: How Prior Media Consumption Shapes Subsequent Media Consumption". en. In: *Journal of Marketing Research* 59.3 (June 2022). Number: 3, pp. 453–471. ISSN: 0022-2437, 1547-7193. DOI: 10.1177/00222437211055403. URL: http://journals.sagepub.com/doi/10.1177/00222437211055403 (visited on 09/26/2023).

[32] *YouTube Statistics 2024 [Users by Country + Demographics]*. en-US. July 2024. URL: https://www.globalmediainsight.com/blog/youtube-users-statistics/ (visited on 07/19/2024).

[33] Shiwei Zhang, Mingfang Wu, and Xiuzhen Zhang. "Utilising a Large Language Model to Annotate Subject Metadata: A Case Study in an Australian National Research Data Catalogue". en. In: ().

A

Technology Stack

Now that the key data points are determined and we understand the human-YouTube interactions and how to simulate them, we can decide on the technology stack upon which the user simulation tool will be built.

Programming language: Python

Python was selected as the primary programming language for the simulation tool for several compelling reasons. Firstly, Python is renowned for its simplicity and readability, which significantly reduces the complexity of writing and maintaining code. This is particularly beneficial for a project involving multiple components and integrations, as it ensures that the codebase remains manageable and accessible. Additionally, Python boasts a vast ecosystem of libraries and frameworks that can streamline various aspects of development, from web automation to data analysis and machine learning. The language's versatility makes it an ideal choice for developing a comprehensive simulation tool that requires integration with external APIs, real-time data processing, and decision-making capabilities based on machine learning models.

Web automation tools to interact with YouTube: Playwright

Playwright was chosen as the web automation tool to interact with YouTube for several reasons. Playwright, a more modern alternative to Selenium, offers advanced features that simplify the automation process. One of the key advantages of Playwright is its support for action recording, which can be directly translated into code. This feature makes the automation setup more intuitive and less error-prone, allowing for a smoother development experience. Furthermore, Playwright supports capturing screenshots of specific elements on a webpage, which is particularly useful for taking screen captures of videos. These screenshots can provide additional context for decision-making processes, enhancing the tool's ability to simulate human-like interactions accurately.

Classification model: GPT-4o API

To perform the classification, a large language model (LLM) driven solution was deemed the most effective, particularly due to its exceptional capabilities in natural language processing (NLP) and classification tasks. Unlike human users, the classification model cannot gather information about the video through vision. Therefore, it relies on the video metadata obtained from the YouTube Data API. This metadata primarily consists of textual information, underscoring the necessity of NLP.

Model of LLM	Accuracy
Llama 3	66%
GPT-3.5 Turbo	73%
GPT-4o	86%

Table A.1: Accuracy of Different LLMs in Classifying YouTube Videos Based on Metadata

Several models were evaluated for this task, including GPT-3.5 Turbo, GPT-4o, and local LLMs such as LLama 3 from Meta. Each model was tested on a sample size of 100 videos to assess their classification accuracy and speed. The results indicated that OpenAI's models had the highest accuracy. Despite the latency associated with API calls, GPT-3.5 Turbo demonstrated the highest speed, outperforming local LLMs in this regard.

Table A.1 shows the accuracy of the three different models testing when tasked to classify 100 videos based on metadata. As can be seen GPT-4o had by far the best results with 86% of videos being correctly classified. Therefore, given that the speeds of GPT-4o are comparable to GPT-3.5 Turbo the decision was made to use 4o.

User reaction decision making: GTP-4o API

Similarly to the classification model, the user reaction model must be capable of making decisions based on the context of textual information. In this case, the relevant contextual information includes the classification result of a video as well as the user persona and interests.

User persona decision-making has been successfully implemented using LLMs in previous research [26]. For this task, we employed GPT-4o. Although this task could be performed by a less powerful and cheaper LLM, GPT-4o was chosen due to its superior performance. Utilizing GPT-4o ensures consistency in model performance across the application, which is crucial for maintaining reliability and accuracy in the simulation tool.

Video Metadata Collection: YouTube Data API

To obtain the necessary metadata for performing natural language processing (NLP) classification of videos, the YouTube Data API is utilized. This API is chosen due to its cost-free access, exceptional speed, and ability to retrieve the required metadata with a single API call. The specific metadata of interest for classification includes:

- **Title**: The title of the video.
- **Description**: The description of the video.
- **Tags**: The tags associated with the video.
- **Channel**: The name of the channel uploading the video.
- **YouTube Category**: One of fifteen predefined YouTube categories.
- **YouTube Topics**: One to three of fifty predefined YouTube topics.

Database: SQL-Lite

To store the data, we chose an SQL database, as the data can be sequentially stored one at a time as a new row in a table as the bot worked. This would ensure CRUD principles and that data would not be lost in the case that the bot crashed before completing all of its iterations.

SQL-Lite, in particular, was chosen as it is very lightweight, possesses all the functionality required for the project, and does not require an extensive setup process.

B

Literature Review Results

Table B.1: List of articles and their focus

Article Name	Reference	Focus/Main Talking Points
Trick and Please. A Mixed-Method Study On User Assumptions About the TikTok Algorithm	(Klug et al. 2021)	Examines user assumptions about the TikTok algorithm and how to make videos trend. It uses mixed methods including qualitative interviews and quantitative data analysis.
#TulsaFlop: A Case Study of Algorithmically-Influenced Collective Action on TikTok	(Bandy & Diakopoulos 2020)	Measure the influence of TikTok's algorithmic recommender system on the visibility of call-to-action videos related to the Tulsa rally and provide a critical analysis of the role of algorithms in shaping human knowledge practices and political ramifications.
An Empirical Investigation of Personalization Factors on TikTok	(Boeker & Urman 2022)	Investigate the influence of user behaviour and characteristics on content distribution on TikTok using a sock-puppet auditing technique to collect data and proposing countermeasures to filter bubbles.
How to Train Your YouTube Recommender to Avoid Unwanted Videos	(Liu et al. 2023)	Investigate the effects of different strategies for removing unwanted recommendations on YouTube focusing on their efficacy for specific topics.
The echo chamber effect on social media	(Cinelli et al. 2021)	Present a conceptual framework and empirical analysis of echo chambers on social media identifying their presence and impact on users' interactions and information consumption and suggesting potential solutions to mitigate their negative effects.
The YouTube Algorithm and the Alt-Right Filter Bubble	(Bryant 2020)	Explore how the YouTube algorithm fuels radicalism within the alt-right by creating filter bubbles and emphasizes the need for greater transparency and accountability in algorithm programming to counteract harmful content.
Science by YouTube: An Analysis of YouTube's Recommendations on the Climate Change Issue	(Dündar & Ranaivoson 2022)	Investigate if YouTube's recommendation system forms filter bubbles regarding climate change using an experimental analysis of suggested videos. It aims to enhance science communication in addressing climate change while contributing to the filter bubble literature.

Continued on next page

Table B.1 – continued from previous page

Article Name	Reference	Focus/Main Talking Points
An Audit of Misinformation Filter Bubbles on YouTube: Bubble Bursting and Recent Behavior Changes	(Tomlein et al. 2021)	Investigate the impact of watching misinformative videos on YouTube and the effectiveness of watching debunking videos in bursting the misinformation filter bubble. The paper also explores the ethical considerations in researching misinformative content.
Measuring Misinformation in Video Search Platforms: An Audit Study on YouTube	(Hussein et al. 2020)	Audit study of YouTube's search and recommendation algorithms to examine the promotion and recommendation of misinformative content for certain search topics and watch histories. It also discusses the negative implications of this promotion and suggests interventions to mitigate the problem.
Auditing YouTube's Recommendation Algorithm for Misinformation Filter Bubbles	(Srba et al. 2023)	Investigate the behavior of YouTube's personalization algorithm in creating and bursting misinformation filter bubbles. Reports on the effectiveness of watching debunking videos in improving the situation and discusses ethical considerations in auditing algorithms.
Down a Rabbit Hole: How Prior Media Consumption Shapes Subsequent Media Consumption	(Woolley & Sharif 2022)	Explore the "rabbit hole effect" in media consumption where prior choices influence subsequent preferences. It analyses how consumers gravitate towards content similar to their previous choices driven by genre categorization. The paper investigates the mechanisms and implications of this phenomenon for marketers and consumers.
Subscriptions and external links help drive resentful users to alternative and extremist YouTube channels	(Chen et al. 2023)	Focuses on analysing the influence of subscriptions and external links on YouTube users' exposure to alternative and extremist channels particularly among viewers with high resentment levels.
"It Is Just a Flu": Assessing the Effect of Watch History on YouTube's Pseudoscientific Video Recommendations	(Papadamou et al. 2022)	Detect and characterize pseudoscientific content on YouTube while assessing the impact of a user's watch history on pseudoscientific video recommendations.
How YouTube Leads Privacy-Seeking Users Away from Reliable Information	(Spinelli & Crovella 2020)	Explores the nature of YouTube recommendations highlighting the tension between user privacy and extreme recommendations and the impact on public opinion formation and information diversity.
Metadata Based Classification and Analysis of Large Scale Web Videos	(Algur & Bhat, 2015)	Classify web videos into categories using metadata attributes like length, ratings, age, and comments. Propose an effective classification model leveraging data mining algorithms (Random Tree and J48), compare accuracy of these models, and address challenges due to insufficient metadata. Highlight importance of metadata in improving video search and information retrieval.

Continued on next page

Table B.1 – continued from previous page

Article Name	Reference	Focus/Main Talking Points
Social Media and Dopamine: Studying Generation Z and Dopamine Levels	(Almachnee & Cozzie, 2022)	Investigate the influence of social media on dopamine levels in Generation Z, leveraging data from 200 respondents. Analyze how social media usage affects dopamine fluctuations, compare findings with existing research, and identify significant impacts of social media on dopamine regulation within this demographic.
Auditing Algorithmic Bias on Twitter	(Bartley et al., 2021)	Implement a methodology to audit Twitter's algorithmic curation. Use matched pairs of bots to compare personalized and chronological timelines, revealing biases in content exposure. Identify recency, popularity, and exposure biases, showing that algorithmic curation skews information users see, amplifies popular content, and distorts perceptions of friend activity. Highlight implications for understanding social media influence.
Predictors of Social Media Self-Control Failure: Immediate Gratifications, Habitual Checking, Ubiquity, and Notifications	(Du et al., 2019b)	Investigate predictors of social media self-control failure (SMSCF) among daily users. Analyze factors like habitual checking, perceived ubiquity, and notifications, identifying their impact on SMSCF. Results show habitual checking, ubiquity, and notifications increase SMSCF, while immediate gratifications do not predict it. Highlight implications for managing media use and suggest interventions targeting changeable factors like notifications.
Auditing YouTube's recommendation system for ideologically congenial, extreme, and problematic recommendations	(Haroon et al., 2023)	Audit YouTube's recommendation system for bias towards ideologically aligned, extreme, and problematic content. Identify patterns, measure algorithmic influence, and discuss potential impacts on viewer perception and polarization.
YouTube Video Classification based on Title and Description Text	(Kalra et al., 2019)	Develop a method for classifying YouTube videos using titles and descriptions. Utilize natural language processing techniques to improve content categorization, enhancing searchability and recommendation accuracy.
Why Do We Need to Be Bots? What Prevents Society from Detecting Biases in Recommendation Systems	(Krafft et al., 2020)	Investigate societal and technical barriers to detecting biases in recommendation systems. Explore the necessity of automated methods (bots) to audit algorithms, highlighting challenges and suggesting improvements for transparency and fairness.
Modeling Rabbit-Holes on YouTube	(Le Merrer et al., 2023)	Analyze the phenomenon of "rabbit-holes" on YouTube. Develop models to understand how YouTube's recommendation algorithm contributes to users' deep dives into specific content themes, assessing patterns and implications for user engagement and behavior.
Auditing Algorithms: Understanding Algorithmic Systems from the Outside In	(Metaxa et al., 2021)	Explore methods for auditing algorithms to understand their decision-making processes from an external perspective. Discuss various approaches, challenges, and implications for transparency, accountability, and ethical considerations in algorithmic systems.

Continued on next page

Table B.1 – continued from previous page

Article Name	Reference	Focus/Main Talking Points
Cutting Through the Comment Chaos: A Supervised Machine Learning Approach to Identifying Relevant YouTube Comments	(Möller et al., 2023)	Apply supervised machine learning to filter and identify relevant comments on YouTube. Develop models to manage and streamline comment sections, enhancing user experience by prioritizing meaningful and engaging interactions.
RECOMMENDATION OF EFFECTIVENESS OF YOUTUBE VIDEO CONTENTS BY QUALITATIVE SENTIMENT ANALYSIS OF ITS COMMENTS AND REPLIES	(Nawaz et al., 2019)	Use qualitative sentiment analysis on YouTube comments and replies to evaluate the effectiveness of video content. Assess how sentiment data can inform content creators about audience reception and engagement.
The Evaluation of the Black Box Problem for AI-Based Recommendations: An Interview-Based Study	(Ochmann et al., 2021)	Investigate the “black box” problem in AI-based recommendation systems through interviews. Explore user perceptions, transparency issues, and the challenges in understanding AI decision-making processes in recommendation algorithms.
A Short Video Classification Framework Based on Cross-Modal Fusion	(Pang et al., 2023)	Develop a classification framework for short videos using cross-modal fusion techniques. Combine audio-visual data to enhance the accuracy and efficiency of video categorization, improving content retrieval and recommendation systems.
History of YouTube - How it All Began & Its Rise	(Rana, 2024)	Trace the history and evolution of YouTube from its inception to its rise as a global video-sharing platform. Highlight key milestones, technological advancements, and the impact on digital media consumption.
Metadata extraction and classification of YouTube videos using sentiment analysis	(Rangaswamy et al., 2016b)	Implement sentiment analysis to extract and classify metadata from YouTube videos. Use sentiment data to improve video categorization and enhance recommendation systems by understanding viewer reactions.
How YouTube Leads Privacy-Seeking Users Away from Reliable Information	(Spinelli & Crovella, 2020c)	Analyze how YouTube’s algorithms may direct privacy-seeking users towards unreliable information. Investigate the mechanisms behind these recommendations and their impact on user trust and information quality.
Effects of Short Video Addiction on the Motivation and Well-Being of Chinese Vocational College Students	(Ye et al., 2022)	Study the impact of short video addiction on the motivation and well-being of Chinese vocational college students. Assess how excessive consumption affects academic performance, mental health, and daily life activities.