**VIDEO RECOMMENDATION SYSTEM**

**A group project report of partial fulfilment of requirements for the minor project (DS1504)**

**By**

**Hemant Mangal (22106042)**

**Bhupesh Joshi (22106051)**

**Sartaj Singh (22106054)**

**Pushkar (22106058)**

**Under the supervision of**

**Dr. Kanu Goel & Dr. Satnam Kaur**



**Department of Computer Science and Engineering**

**Punjab Engineering College (Deemed to be University)**

**Chandigarh**

**DECLARATION**

**We hereby declare that the group project report VIDEO RECOMMENDATION SYSTEM, submitted for partial fulfillment of the requirements for the Minor Project (DS 15004), is a bonafide work done by us under the supervision of Dr. Kanu Goel & Dr. Satnam Kaur.**

**This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the sources.**

**We also declare that we have adhered to the ethics of academic honesty and integrity and have not misrepresented or fabricated any data idea fact or source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained.**

**This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.**

**Hemant Mangal (22106042)**

**Bhupesh Joshi (22106051)**

**Sartaj Singh (22106054)**

**Pushkar (22106058)**

**(i)**

**PUNJAB ENGINEERING COLLEGE (DEEMED TO BE UNIVERSITY)**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**CHANDIGARH**

**2023 - 24**

****

**CERTIFICATE**

**This is to certify that the report entitled VIDEO RECOMMENDATION SYSTEM submitted by Hemant Mangal, Bhupesh Joshi, Sartaj Singh and Pushkar to the Punjab Engineering College (Deemed to be University), Chandigarh in partial fulfillment of requirements for the Minor Project (DS 1504) in Computer Science & Engineering (Data Science) is a bonafide record of the project preliminary work carried out by them under our guidance and supervision.**

**This report in any form has not been submitted to any other University or Institute for any purpose.**

**Dr. Kanu Goel Dr. Satnam Kaur**

**(Project Mentor) (Project Mentor)**

**Assistant Professor Assistant Professor**

**Dept. of CSE Dept. of CSE**

**Punjab Engineering College Punjab Engineering college**

**(Deemed to be university) (Deemed to be university)**

**Chandigarh Chandigarh**

**(ii)**

**Acknowledgement**

We would like to take this opportunity to thank our college Punjab Engineering College (Deemed to be University), Chandigarh, and the Department of Computer Science and Engineering for allowing us to work on this project. We are immensely grateful to our project mentors Dr. Kanu Goel (Assistant Professor, Department of Computer Science & Engineering) & Dr. Satnam Kaur(Assistant Professor, Department of Computer Science & Engineering) whose continuous guidance, technical support, and moral support at times of difficulty helped us to achieve milestones in the given time.

They have been a great source of knowledge. We are also thankful to the Committee for evaluation who gave their valuable feedback and guidelines for the betterment and completion of this project. We convey our deep sense of gratitude to all teaching and non-teaching staff for their constant encouragement, support, and selfless help throughout the project work.

It is a great pleasure to acknowledge the help and suggestions we received from the Department of Computer Science and Engineering.

We wish to express our profound thanks to all those who helped us gather information about the project. Our families too have provided moral support and encouragement several times.

Hemant Mangal

Bhupesh Joshi

Sartaj Singh

Pushkar

**(iii)**

**Abstract**

Video recommendation systems play a crucial role in delivering personalized and engaging content to users. This paper presents our approach to building an effective video recommendation system, focusing on content-based filtering for video recommendation system. By analyzing features such as video titles, descriptions, and categories, we generate a curated list of videos tailored to the user's specific interests and viewing patterns. This approach ensures that the recommended videos are relevant and aligned with the user's preferences, enhancing their overall experience.

Our system is designed to handle challenges such as managing large video datasets and accurately capturing user interests based on available content information. Content-based filtering allows the system to recommend videos even for new or less common users by relying solely on video attributes rather than user interactions. This makes the system more adaptable and robust, especially when dealing with sparse user data.

In future work, we plan to develop and integrate an advanced ranking mechanism to further refine these recommendations. The ranking system will evaluate the generated candidates using a broader set of features, such as user behavior patterns, historical interactions, and contextual data. This will ensure that the most engaging and personalized content appears at the top of the recommendation list. Our ultimate goal is to continuously enhance the recommendation process, making it more precise, adaptive, and aligned with evolving user preferences.

**(iv)**

**Contents**

Acknowledgement (iii)

Abstract (iv)

List of figures (vi)

List of abbreviations (vii)

1. Introduction ……………………………………………………… 1

1.1 Types of Chatbot ……………………………………………….

2. Background & Literature review 4

2.1Deep Neural Networks for YouTube Recommendations………………. 4

2.2 Netflix Recommender System: Algorithms, Innovation. ………………… 4

3. Proposed Work

3.1 Methodology ………………………………………………. 6

3.2 Requirement Analysis ………………………………………… 7

3.3 Design Phase ………………………………………………. 7

4. Implementation details

4.1 Technology Stack ………………………………………………. 9

4.1.1 HTML ………………………………………………. 9 (v)

4.1.2 Tailwind CSS ………………………………………… 10

4.1.3 Typescript ………………………………………………. 10

4.1.4 React.js ………………………………………………. 11

4.1.5 MS Visual Studio Code …………………………………12

4.2 Login and Signup page …………………………………………… 13

5. Results and discussion

5.1 Response Generated ………………………………………….. 17

6 .Conclusions and future work

6.1 Music Curation for Events and Activities ….………………….. 20

6.2 Marketing and Customer Engagement ………………………….. 20

6.3 Future Potential ……………………………………….….. 21 7.

References ………………………………………………. 22

**List of figures**

**List of Tables**

**List of Abbreviations**

**INTRODUCTION**

Video recommendation systems have become essential tools for guiding users through vast and ever-expanding digital content libraries. By offering personalized suggestions, these systems enhance user engagement and satisfaction, helping users discover content aligned with their preferences and viewing habits. With the rapid growth of video-sharing platforms and streaming services, the challenge lies in efficiently analyzing large datasets and accurately predicting what users might want to watch next.

**Collaborative Filtering:**  
Traditional recommendation approaches often rely on collaborative filtering, which analyzes user interactions to identify patterns. Collaborative filtering compares user behavior, such as watch history and ratings, to find similarities between users or items. However, these methods can struggle with new or infrequent users and require extensive historical data to generate meaningful suggestions.

**Hybrid Filtering:**  
Hybrid filtering combines both collaborative and content-based approaches to overcome these limitations, leveraging user interaction data alongside video attributes for more comprehensive recommendations. This combination ensures a more robust recommendation process by addressing the weaknesses of each individual approach.

**Content-Based Filtering:**  
In contrast, content-based filtering offers a powerful alternative by leveraging intrinsic video features such as titles, descriptions, and categories. This technique can provide relevant recommendations solely based on the attributes of the videos, making it particularly effective for new users or niche content where user interaction data may be limited.

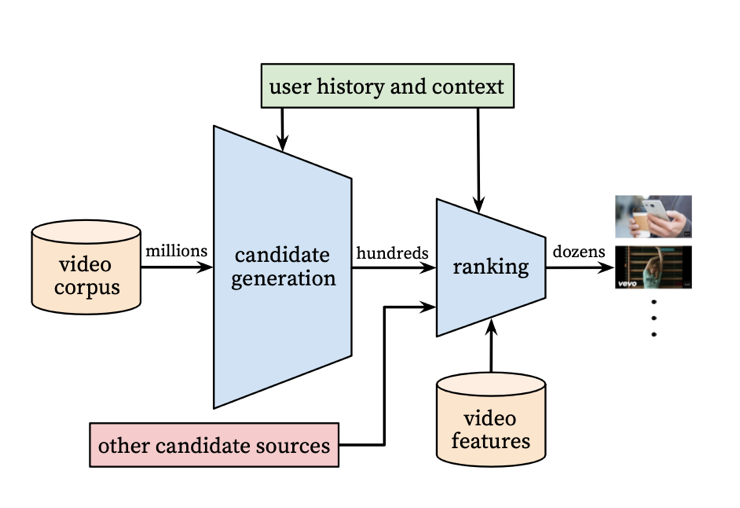
**Proposed System:**  
Our proposed system focuses on candidate generation using content-based filtering to create an initial pool of relevant videos. This method ensures that the recommendations are grounded in the actual content characteristics, allowing for more precise and contextually relevant suggestions.  
As part of our future work, we aim to enhance this system by integrating an advanced ranking mechanism. This next stage will involve evaluating the generated candidates using additional features, including user behavior patterns and contextual data, to prioritize the most engaging and personalized content.  
This paper explores the design and implementation of our content-based recommendation system, addressing key challenges such as handling large datasets and ensuring recommendation relevance. We also outline our future plans for integrating a ranking model to further refine the recommendation process, aiming to create a dynamic and user-centric system that continuously adapts to evolving preferences.

**Background & Literature review**

**Deep Neural Networks for YouTube Recommendations**

Video recommendation systems are essential for helping users discover content tailored to their preferences. Traditional methods like collaborative filtering and matrix factorization have been widely used to analyze user interactions and make recommendations. However, these approaches often struggle with challenges like scalability, sparse user data, and dynamic content updates. To address these limitations, deep learning techniques have revolutionized recommendation systems by enabling the processing of large-scale data and capturing complex user preferences​.

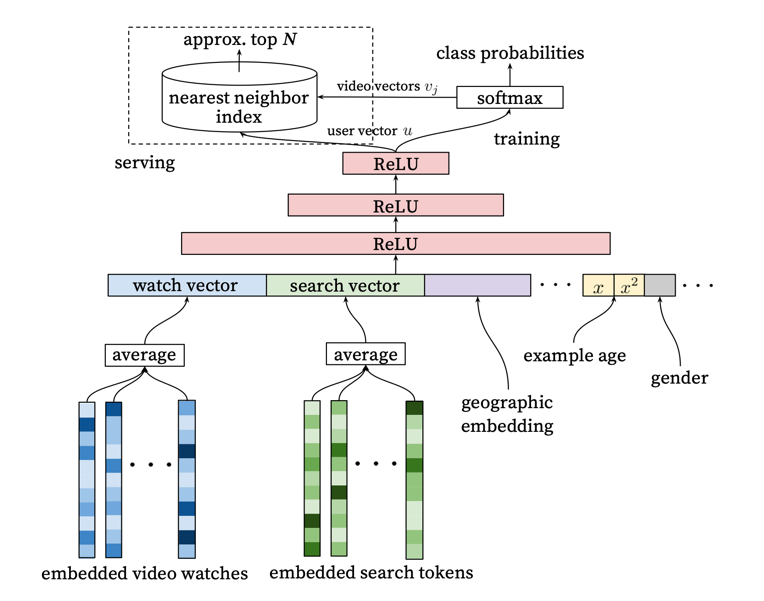
The YouTube recommendation system, one of the most advanced in the world, adopts a **two-stage architecture**. The process begins with **candidate generation**, which reduces a vast pool of videos (millions) to a smaller set (hundreds) of relevant candidates. This is achieved using user activity history, such as watch and search data, which are embedded into dense vectors to represent user preferences. These embeddings are processed through a neural network to generate recommendations. The next stage, **ranking**, further refines this list to a few videos (dozens) that are displayed to the user. This stage assigns scores to videos using features like demographics, video metadata, and user engagement signals, ensuring the top-scored videos are most relevant​.



Recommendation system architecture demonstrating the candidate videos are retrieved and ranked before presenting only a few to the user.

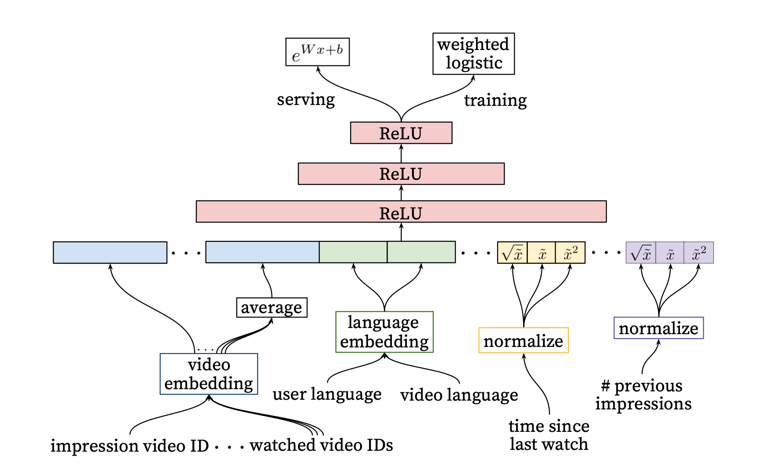
**Figure below** demonstrates the architecture of the **candidate generation model**, which uses a deep neural network. User history, such as watched video IDs and search queries, is embedded into dense vectors. These vectors are combined with demographic features like geographic location and device type. The network processes these inputs through multiple layers of **Rectified Linear Units (ReLU)** to model interactions and produce a shortlist of videos. The hierarchical structure enables the system to capture nonlinear relationships, significantly improving prediction accuracy.

page



Deep candidate generation model architecture showing embedded sparse features concatenated with dense features. Embeddings are averaged before concatenation to transform variable sized bags of sparse IDs into fixed-width vectors suitable for input to the hidden layers. All hidden layers are fully connected. In training, a cross-entropy loss is minimized with gradient descent on the output of the sampled softmax. At serving, an approximate nearest neighbor lookup is performed to generate hundreds of candidate video recommendations.

Once the candidate videos are identified, the **ranking model** assigns scores to these videos based on additional features. This model incorporates both categorical data, like video IDs, and continuous data, like time since the user’s last watch. Features are normalized to ensure uniform scaling, and embeddings are used to convert categorical data into dense vectors. These inputs pass through multiple ReLU layers, and the final score is determined using logistic regression. The ranking stage prioritizes videos by expected watch time, ensuring engaging and relevant recommendations. Importantly, this stage also accounts for user behaviors, such as skipping videos or preferring fresh content, to refine the recommendations further​.



Deep ranking network architecture depicting embedded categorical features (both univalent and multivalent) with shared embeddings and powers of normalized continuous features. All layers are fully connected. In practice, hundreds of features are fed into the network.

page

In summary, modern video recommendation systems leverage deep learning to address scalability, dynamic data updates, and user engagement challenges. By combining candidate generation and ranking stages with advanced neural network architectures, these systems provide personalized, accurate, and timely recommendations. Future work should explore integrating real-time feedback and addressing ethical concerns such as bias and privacy.

The Netflix Recommender System: Algorithms, Business Value, and Innovation

**How Content-Based Video Recommendation Works**

Content-based video recommendation systems analyze attributes of videos, such as titles, descriptions, genres, and cast members, to suggest relevant content to users. These systems focus on matching content features to user preferences rather than relying on interactions with other users.

1. **Feature Extraction:**  
   Textual data, such as video descriptions and metadata, are converted into structured representations. Methods like Term Frequency-Inverse Document Frequency (TF-IDF) are commonly used to weigh the importance of words in video descriptions. This enables the system to assess the similarity between different videos.
2. **Similarity Measurement:**  
   The system calculates the similarity between videos using metrics like cosine similarity. Higher similarity scores indicate stronger relevance between videos. This approach ensures that recommended videos share characteristics with those the user has previously enjoyed.
3. **Algorithm Training and Evaluation:**  
   Machine learning models are trained on historical data to predict user preferences. These models learn from attributes such as genres, keywords, and user ratings to refine recommendations. A/B testing is often used to compare different versions of the algorithm, ensuring that changes improve user engagement and retention.

**Validity and Performance Assessment**

The validity of content-based recommendation systems is evaluated through a combination of offline experiments and real-world A/B testing:

1. **Offline Experiments:**  
   These involve testing the algorithm against historical data to predict user behavior. For example, an algorithm is evaluated based on how well it ranks videos that users have previously watched. While this method allows rapid iteration, it assumes that past user behavior accurately predicts future actions, which may not always be true.
2. **A/B Testing:**  
   In A/B tests, different groups of users are exposed to various algorithm versions to measure real-time performance. Metrics like watch time, user retention, and click-through rates are analyzed. Tests typically run for several months to capture long-term effects. Successful algorithms show significant improvements in these metrics compared to the control group.
3. **Challenges in Testing:**
   * **Sample Size:** Ensuring statistical significance requires large sample sizes, especially when measuring subtle improvements. For instance, detecting a 0.1% increase in retention might require millions of users per test group.
   * **Bias Control:** Algorithms must account for presentation bias, where highly recommended videos are more likely to be watched simply because they are prominently displayed. Techniques like randomizing recommendations or adjusting for exposure frequency help mitigate this issue.

**Key Findings and Practical Implications**

Content-based systems have demonstrated effectiveness in recommending niche and long-tail content that might be overlooked by collaborative filtering approaches. They provide reliable suggestions even for new users with limited interaction history. However, overspecialization can occur, where recommendations become too narrow, limiting content diversity. Future research focuses on integrating richer data sources and improving personalization techniques to address these limitations.

In summary, content-based video recommendation systems offer a robust framework for delivering personalized content, validated through rigorous offline and real-world testing. Continuous improvements in feature extraction and algorithm evaluation are crucial for enhancing system performance and user satisfaction.

**Proposed Work**

DATA COLLECTION:

The data for this project was collected through a Python-based automated process that utilized the YouTube Data API v3 and the YouTubeTranscriptApi library. The objective was to gather a diverse and detailed dataset from YouTube videos across various categories, including Film & Animation, Music, Gaming, and more. The collection process began with identifying video categories using predefined category IDs, ensuring a wide representation of content types. For each category, the script retrieved metadata for videos, including essential information such as video titles, descriptions, channel names, view counts, like counts, comment counts, and video durations. A specific focus was placed on videos with durations ranging from 3 to 15 minutes, as this range typically captures concise, viewer-friendly content. This filtering criterion helped maintain consistency and relevance across the dataset.

Once the videos were identified and filtered, the script extracted their transcripts to provide deeper insights into the content. The YouTubeTranscriptApi library enabled this by accessing available subtitles for each video, which could be used for further linguistic, semantic, or thematic analysis. The script employed multithreading, utilizing Python’s ThreadPoolExecutor module, to efficiently handle the transcript extraction process for multiple videos simultaneously. This optimization ensured that the data collection was both time-efficient and capable of processing large volumes of video data.

The final dataset was meticulously structured, with all the collected data stored in a well-organized JSON format. This format allows for easy integration into various data analysis tools and workflows, ensuring the dataset is ready for immediate exploration and visualization. The combination of video metadata and transcript content offers a rich resource for analyzing trends, audience engagement, and content strategies within different YouTube categories. By automating the entire process, this approach ensures scalability, reliability, and replicability, making it ideal for large-scale studies of YouTube's dynamic content ecosystem. This systematic and detailed collection process provides a robust foundation for understanding user preferences, video performance, and emerging patterns in digital media consumption.

DATA PREPARATION:

**Data Preprocessing and Preparation for Research Paper**

**1. Introduction**

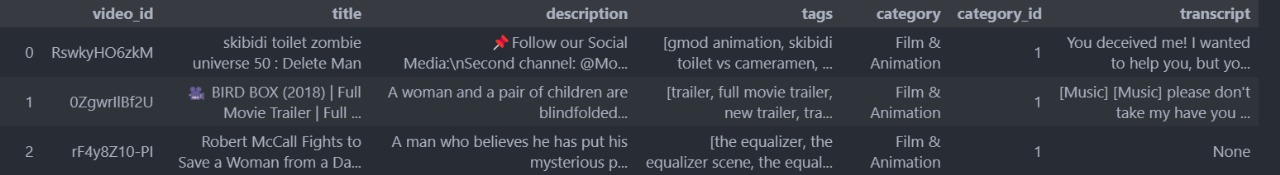
This study involves analyzing YouTube video metadata, focusing on preprocessing text data for a video recommendation system. Key steps include data cleaning, language detection, text normalization, and generating word embeddings using advanced models like BERT and spaCy.

**2. Dataset Description**

The dataset contains the following fields:

* **Video ID**: Unique identifier for each video.
* **Title**: Video title.
* **Description**: Video description.
* **Tags**: Keywords associated with the video.
* **Category**: The category to which the video belongs.
* **Category ID**: Numeric ID for the video category.
* **Transcript**: Text transcript of the video content.
* **Likes**
* **Views**
* **Comment count**

**Sample Data (Before Preprocessing):**



**3. Preprocessing Steps**

**Step 1: Handling Missing Data**

* **Objective**: Remove rows with missing essential data.
* **Action**: Dropped rows where video\_id, title, description, category, or transcript were null.

**Step 2: Removing Emojis**

* **Objective**: Standardize text by removing non-textual symbols.
* **Action**: Applied a regular expression to strip out emojis from title, description, and transcript.

**Step 3: Language Detection and Filtering**

* **Objective**: Ensure uniform language (English) across the dataset.
* **Action**: Used the langdetect library to filter out non-English content, ensuring only English text was retained.

**Step 4: Text Cleaning and Normalization**

* **Objective**: Prepare text for analysis by removing noise and standardizing format.
* **Actions**:
  + Converted text to lowercase.
  + Removed special characters, numbers, and punctuations.
  + Removed common English stopwords using the NLTK library.

**Step 5: Text Preprocessing for Specific Columns**

* **Objective**: Apply text cleaning consistently to key fields.
* **Columns Processed**: title, description, and transcript.
  + Each text field was cleaned using the steps above, ensuring consistency.

**Step 6: Post-Cleaning Validation**

* **Objective**: Ensure data quality after preprocessing.
  + Removed rows with empty title or transcript.
  + Filtered out entries without tags to maintain relevance.

**4. Embedding Generation**

**Approach**: To prepare for video recommendations based on user preferences, embeddings were generated using two methods:

1. **spaCy Embeddings**:
   * **Model**: en\_core\_web\_md.
   * **Process**: Transformed user input into a vector representation using spaCy’s pre-trained model.
2. **BERT Embeddings**:
   * **Model**: bert-base-uncased.
   * **Process**:
     + Combined title, description, and tags into a single text field (combined\_text).
     + Tokenized the combined text using the BERT tokenizer.
     + Generated embeddings from the pooled output of the BERT model.

**Example**:

df['combined\_embedding'] = df['combined\_text'].progress\_apply(generate\_embedding)

**5. Output and Next Steps**

* **Processed Data**: Saved to preprocessed\_dataset.csv.
* **Future Work**: Utilize generated embeddings to build and refine the recommendation system, focusing on improving candidate generation and ranking processes.

**Sample Data (Before Preprocessing):**



This preprocessing pipeline ensures high-quality, standardized, and ready-to-use data for downstream tasks like content-based video recommendation.