AI ASSISTED FRAMEWORK FOR VIDEO RECOMMENDATION OF SOCIAL NETWORKING PLATFORMS

Under the Guidance of Dr. Kanu Goel and Dr. Satnam Kaur

PRESENTED BY

Hemant Mangal (22106042)
Bhupesh Joshi (22106051)
Sartaj Singh (22106054)
Pushkar (22106058)

PROBLEM STATEMENT

Recommending relevant videos efficiently remains a challenge due to sparse user data and the cold start problem. Our pipeline addresses these limitations through a hybrid approach that combines content-based filtering (TF-IDF and BERT embeddings) with clustering and neural network-based recommendation models (LSTM/GRU).

OUR OBJECTIVE

01 Personalized Video Recommendations

Develop a system that provides tailored video suggestions using metadata analysis.

02 Semantic Understanding with Embeddings

Leverage TF-IDF and BERT to capture deep semantic relationships in video metadata.

03 Dynamic Feedback Integration

Utilize user feedback (likes/dislikes) to refine and update recommendations in real time.

04 Scalable and Efficient Architecture

Design a system capable of handling large datasets and adapting to new videos dynamically.

TECH STACK

Data Collection

YouTube API for metadata and transcripts.

Feature Extraction

TF-IDF and BERT for embeddings.

Models

LSTM and GRU for embedding predictions.

Tools

Python, TensorFlow and NLTK

NLTK

Clustering

K-Means for video grouping based on similarity.

DATA AQUISITION

Data Collection

Data collected using YouTube Data API v3, focusing on attributes like video IDs, titles, descriptions, tags, categories, likes, and transcripts.

Transcript Analysis

Transcripts processed using YouTubeTranscriptApi for text analysis.

Filtering Criteria

Videos filtered by duration (3–15 minutes) to maintain viewer engagement.

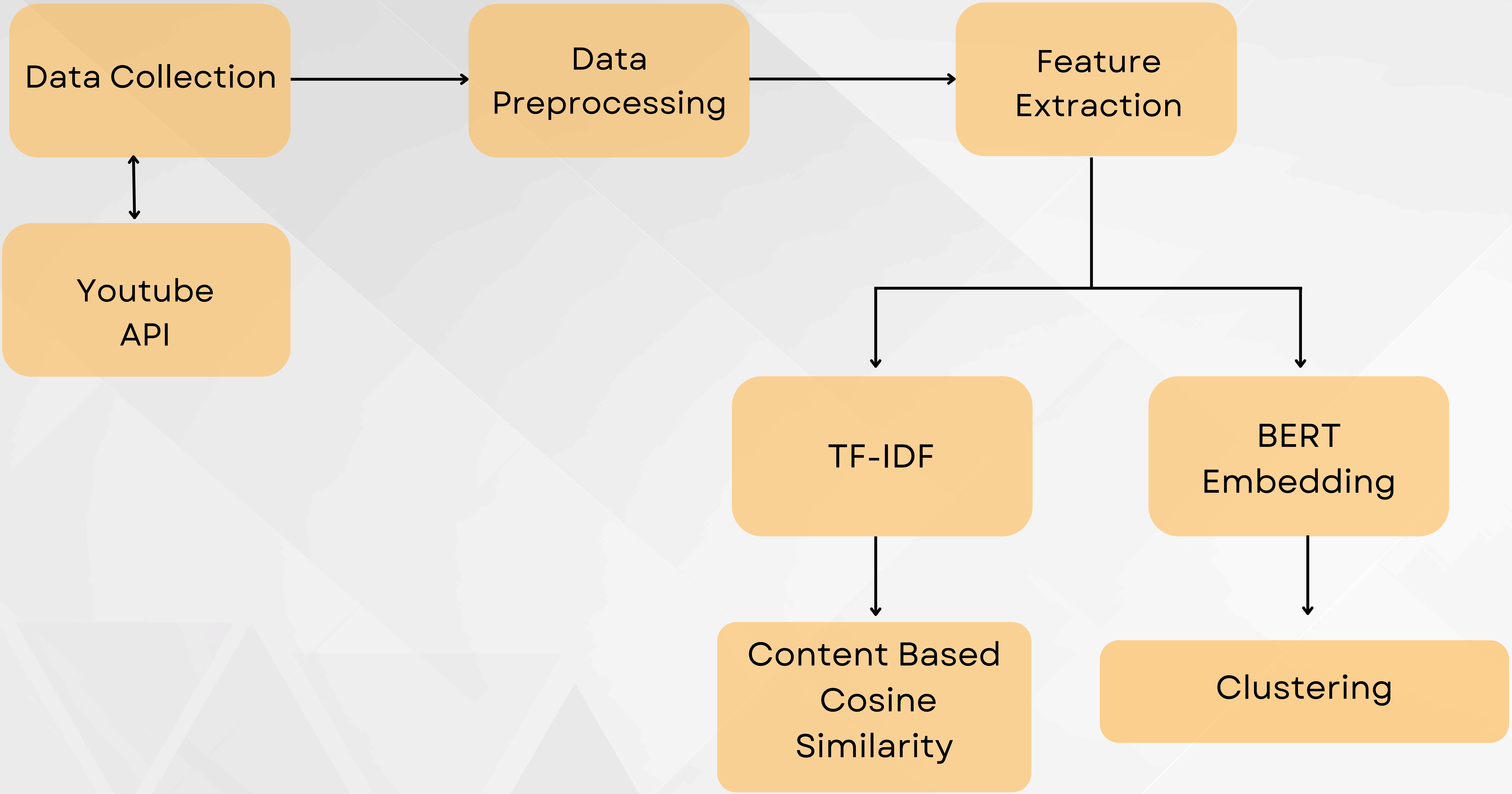
Performance Optimization

Multitheading enabled faster data extraction across large datasets.

DATA PREPROCESSING

Data preprocessing involved cleaning and standardizing video metadata, removing inconsistencies, and preparing text data using tokenization, stemming, and embeddings to ensure high-quality inputs for the recommendation system.

- Cleaned metadata using text preprocessing techniques (tokenization, stemming, stopword removal).
- Non-English entries filtered using langdetect.
- BERT embeddings generated for deep semantic analysis of textual data.
- TF-IDF scores weighted across metadata fields for feature representation.



Methodology

FEATURE SELECTION

TF-IDF:

Weighted importance of title, description, and tags (50%, 30%, 20%).

BERT Embeddings

Captures context and semantic relationships between metadata elements.

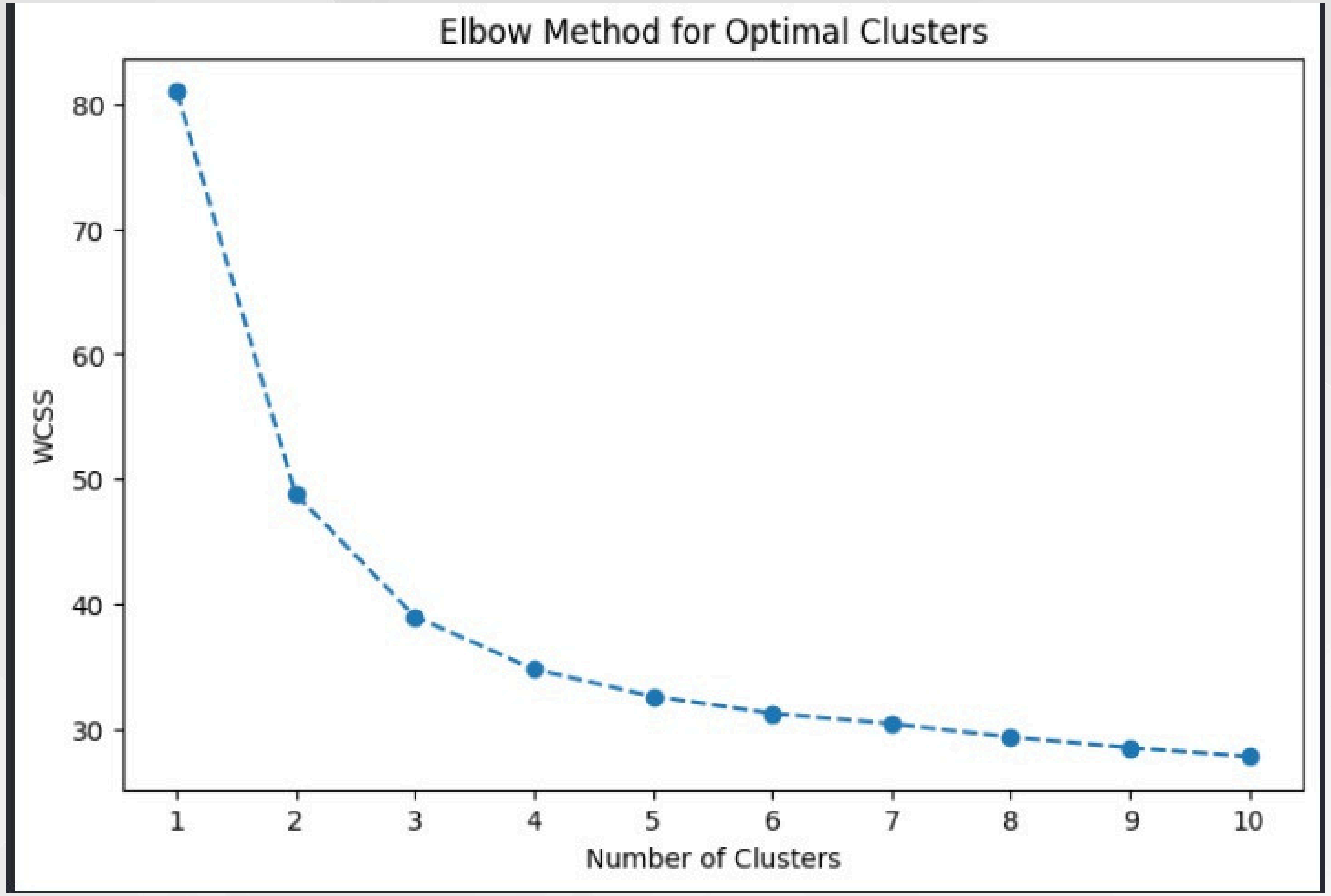
Cosine similarity

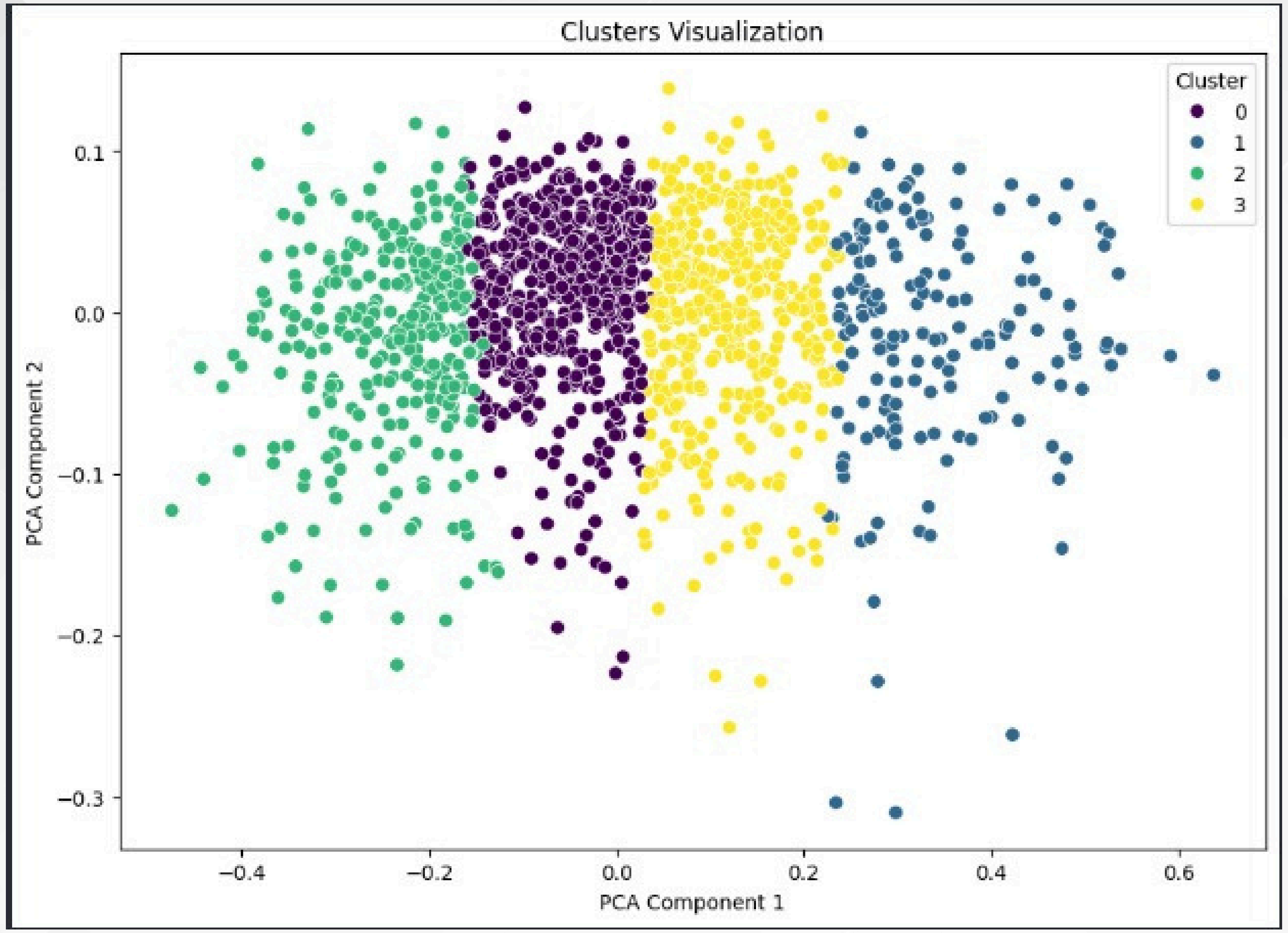
It is used for clustering and ranking videos based on content relevance

```
df[['video_id','combined_embedding']]
```

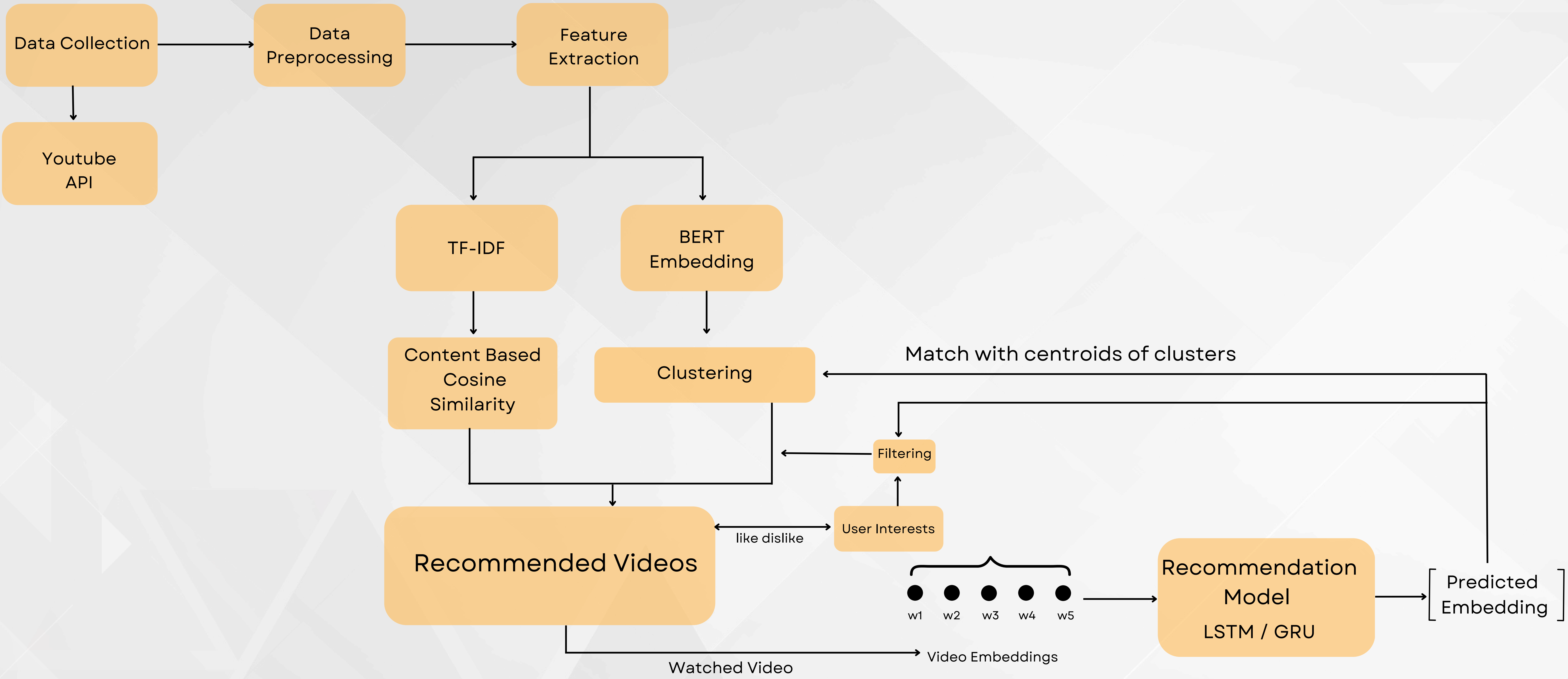
✓ 0.0s

	video_id	combined_embedding
0	UrRZ-kKnBEw	[-0.85304976, -0.57584673, -0.8508247, 0.62368...
1	i2nyb8AYfR4	[-0.7067046, -0.37589747, -0.41218412, 0.30357...
2	h3qWScaLB9I	[-0.67882466, -0.6013942, -0.8749499, 0.492596...
3	nG3NSE7U3M8	[-0.72070384, -0.5049586, -0.61066866, 0.48221...
4	bk7McNUjWgw	[-0.7762446, -0.443919, -0.5427689, 0.4812273,...
...	...	...
1472	UA-Tk9qlG9A	[-0.7893694, -0.43548054, -0.327786, 0.5024284...
1473	25SQ71zi8wA	[-0.8028755, -0.3428881, -0.042129423, 0.58514...
1474	MKRGgX5rYbY	[-0.6665717, -0.44774604, -0.8272868, 0.514428...
1475	MrTxRn9MNWM	[-0.7678555, -0.5277248, -0.89222556, 0.598504...
1476	Ecc7SHqymPo	[-0.7903542, -0.5647549, -0.895528, 0.60848737...


Elbow Method for Optimal Clusters
80
70
60
50
40
30
WCSS
1
2
3
4
5
6
7
8
9
10
Number of Clusters


Clusters Visualization
Cluster
0
1
2
3
0.1
0.0
−0.1
−0.2
−0.3
PCA Component 2
−0.4
−0.2
0.0
0.2
0.4
0.6
PCA Component 1

ARCHITECHTURE

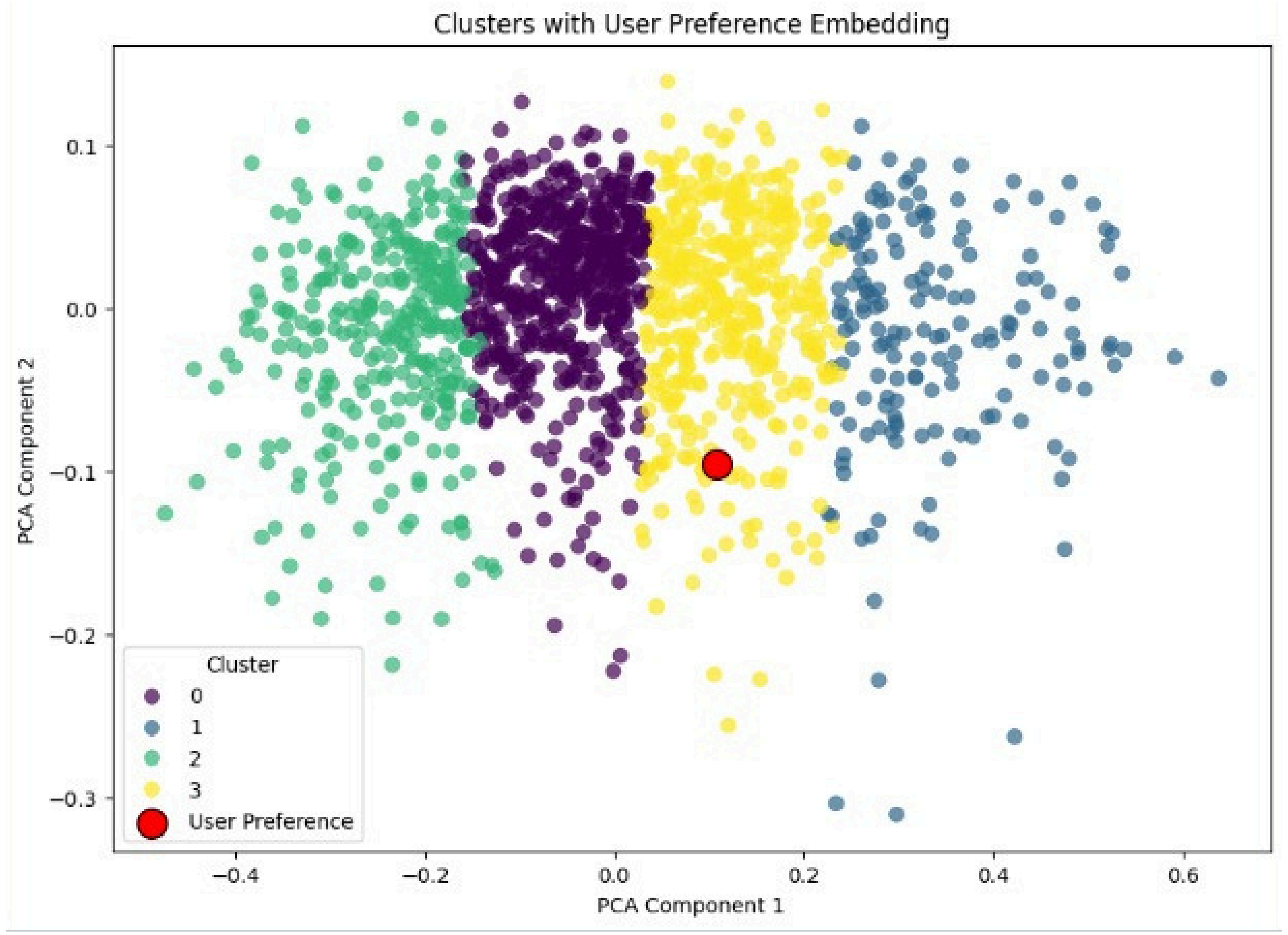


Model Training

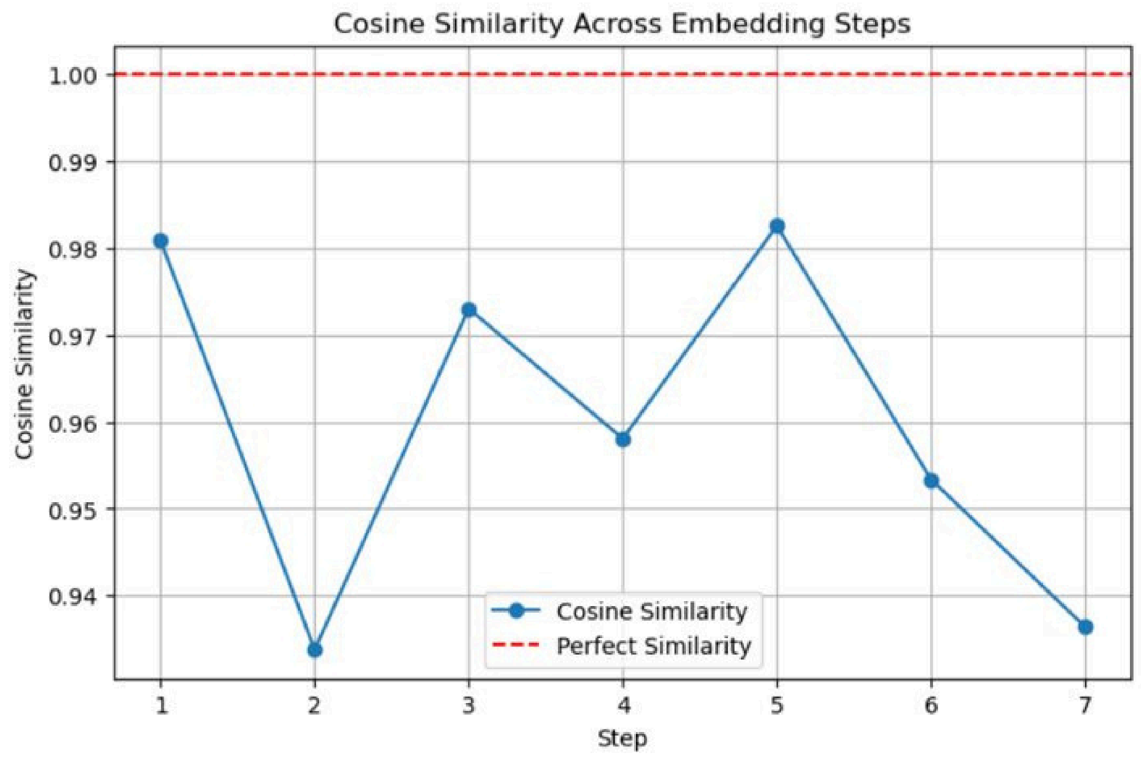
```
['M7pMtxicaVU', '28Du9nIQwBM', 'UrRZ-kKnBEw', 'rF4y8Z10-PI', 's5UarLR8SmE', 'jyD3uKKxI0g', '0ZgwrI1Bf2U', 'Ecc7SHqymPo']
Embeddings shape after extraction: (9, 768)
Embeddings shape: (9, 768)
Initial embedding: [-0.80943596 -0.47267896 -0.27824128  0.52083296  0.11249123 -0.18286908
  0.70619047  0.40365812  0.12661737 -0.99996591]
Step 1: Loss = 0.3132
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 734ms/step
Embedding after Step 1: [ 0.00023709 -0.14852361 -0.11374148  0.0654068   0.12000379  0.00099999
  0.10392206  0.06711336 -0.23685373 -0.06853326]
Epoch 2
Step 1: Loss = 0.2913
Step 2: Loss = 0.2745
Step 3: Loss = 0.2593
Step 4: Loss = 0.2442
Step 5: Loss = 0.2289
Step 6: Loss = 0.2137
Step 7: Loss = 0.1990
Step 8: Loss = 0.1851
Epoch 3
Step 1: Loss = 0.1722
Step 2: Loss = 0.1605
Step 3: Loss = 0.1499
Step 4: Loss = 0.1404
Step 5: Loss = 0.1319
Step 6: Loss = 0.1243
Step 7: Loss = 0.1175
Step 8: Loss = 0.1113
Initial embedding: [-0.80943596 -0.47267896 -0.27824128  0.52083296  0.11249123 -0.18286908
  0.70619047  0.40365812  0.12661737 -0.99996591]
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 183ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 58ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 61ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 87ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 59ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 55ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 79ms/step
Final updated embedding shape: (768,)
Final updated embedding (sample): [-0.90446583 -0.43673611 -0.19331596  0.52154935  0.08588572 -0.15654305
...
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 79ms/step
Final updated embedding shape: (768,)
Final updated embedding (sample): [-0.90446583 -0.43673611 -0.19331596  0.52154935  0.08588572 -0.15654305
  0.71923721  0.32072594  0.2402307  -1.08023406]
```

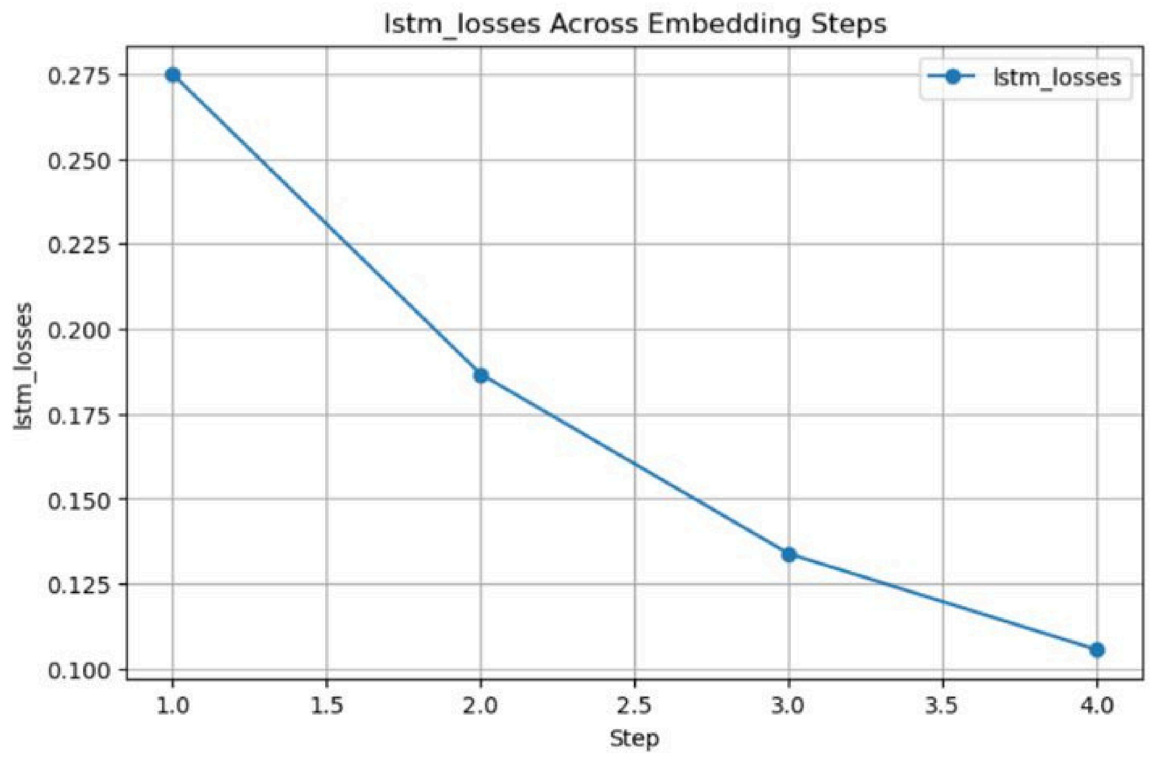
```
✓ 21.3s
['jyD3uKKxI0g', 's5UarLR8SmE', '0ZgwrI1Bf2U', 'UrRZ-kKnBEw', '28Du9nIQwBM', 'jc0ZcR8ozjE', 'Ecc7SHqymPo', 'rF4y8Z10-PI']
Embeddings shape after extraction: (9, 768)
Embeddings shape: (9, 768)
Initial embedding: [-0.80943596 -0.47267896 -0.27824128  0.52083296  0.11249123 -0.18286908
  0.70619047  0.40365812  0.12661737 -0.99996591]
Step 1: Loss = 0.2969
Epoch 2
Step 1: Loss = 0.2854
Step 2: Loss = 0.2744
Step 3: Loss = 0.2630
Step 4: Loss = 0.2507
Step 5: Loss = 0.2377
Step 6: Loss = 0.2243
Step 7: Loss = 0.2109
Step 8: Loss = 0.1979
Epoch 3
Step 1: Loss = 0.1854
Step 2: Loss = 0.1738
Step 3: Loss = 0.1631
Step 4: Loss = 0.1533
Step 5: Loss = 0.1444
Step 6: Loss = 0.1363
Step 7: Loss = 0.1289
Step 8: Loss = 0.1222
Initial embedding: [-0.80943596 -0.47267896 -0.27824128  0.52083296  0.11249123 -0.18286908
  0.70619047  0.40365812  0.12661737 -0.99996591]
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 440ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 122ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 57ms/step
Final updated embedding shape: (768,)
Final updated embedding (sample): [-0.94881514 -0.48972443 -0.37056927  0.59662903  0.14898405 -0.17151255
...
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 57ms/step
Final updated embedding shape: (768,)
Final updated embedding (sample): [-0.94881514 -0.48972443 -0.37056927  0.59662903  0.14898405 -0.17151255
  0.83380765  0.36086831  0.14566104 -0.9942729 ]
```

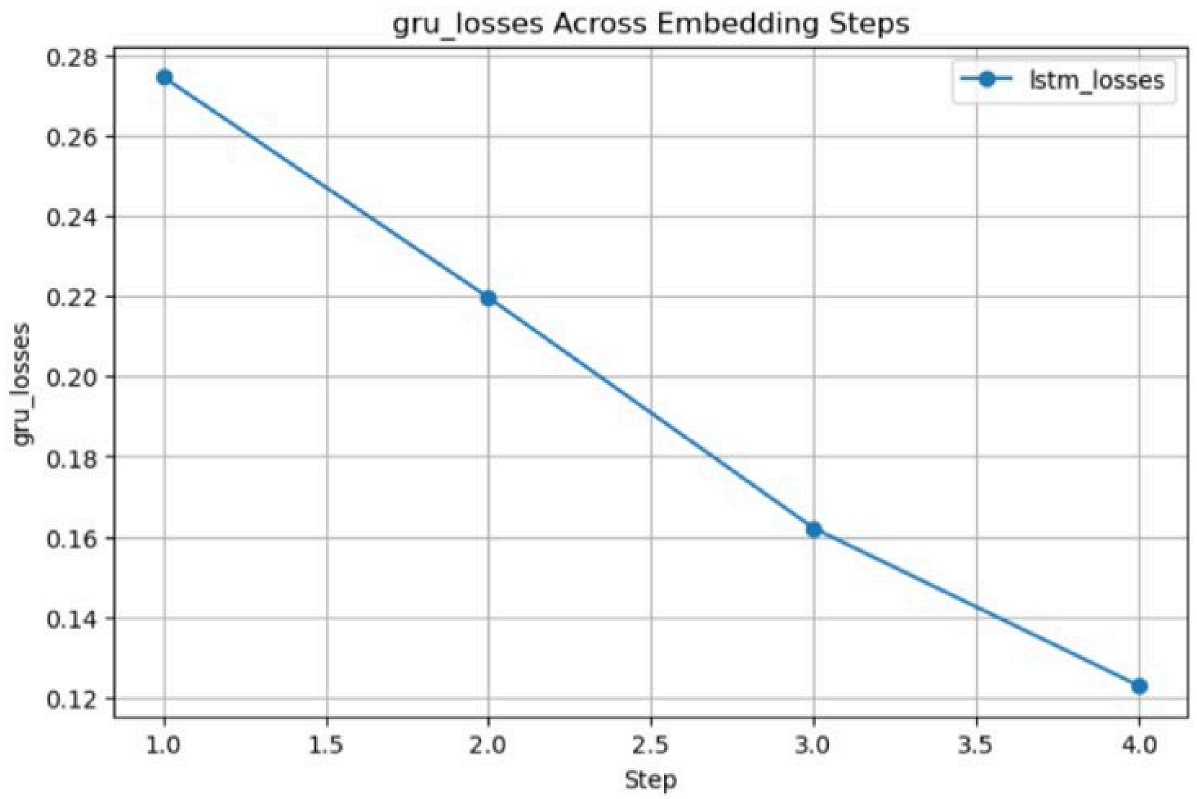
Results

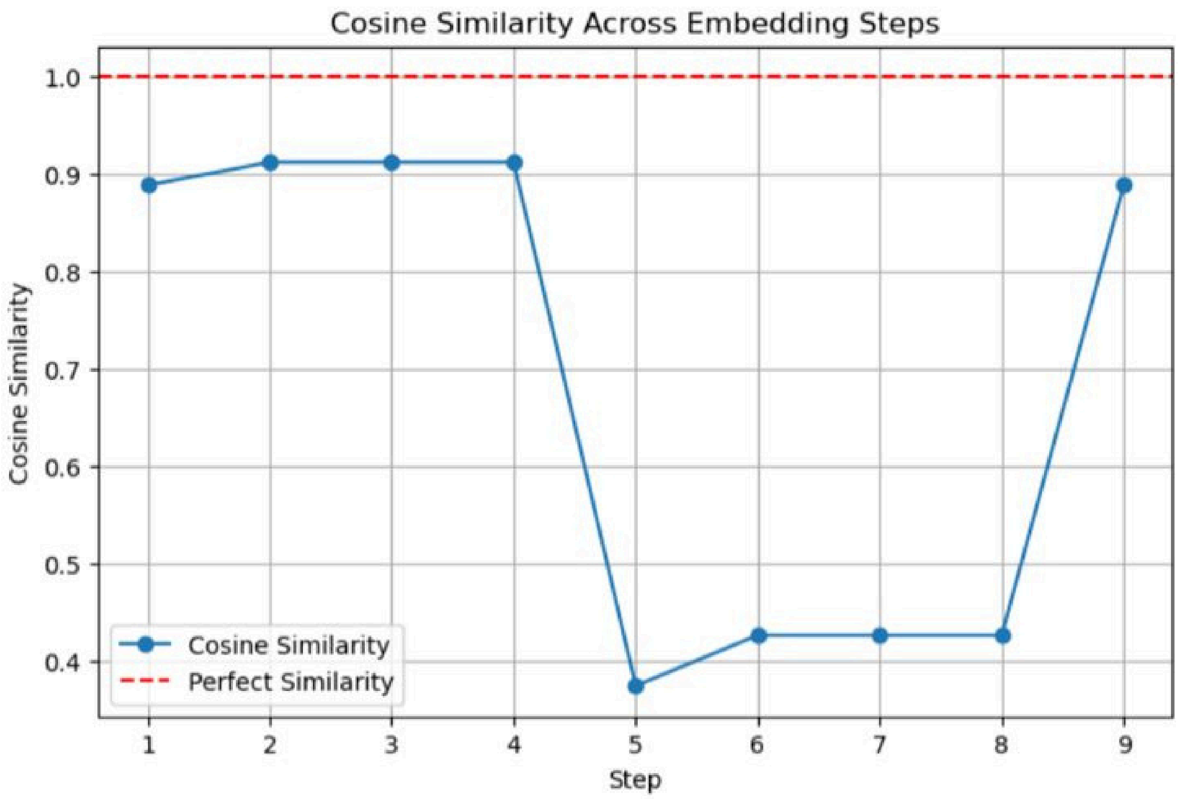

Clusters with User Preference Embedding
PCA Component 2
PCA Component 1
0.1
0.0
−0.1
−0.2
−0.3
−0.4
−0.2
0.0
0.2
0.4
0.6
Cluster
0
1
2
3
User Preference

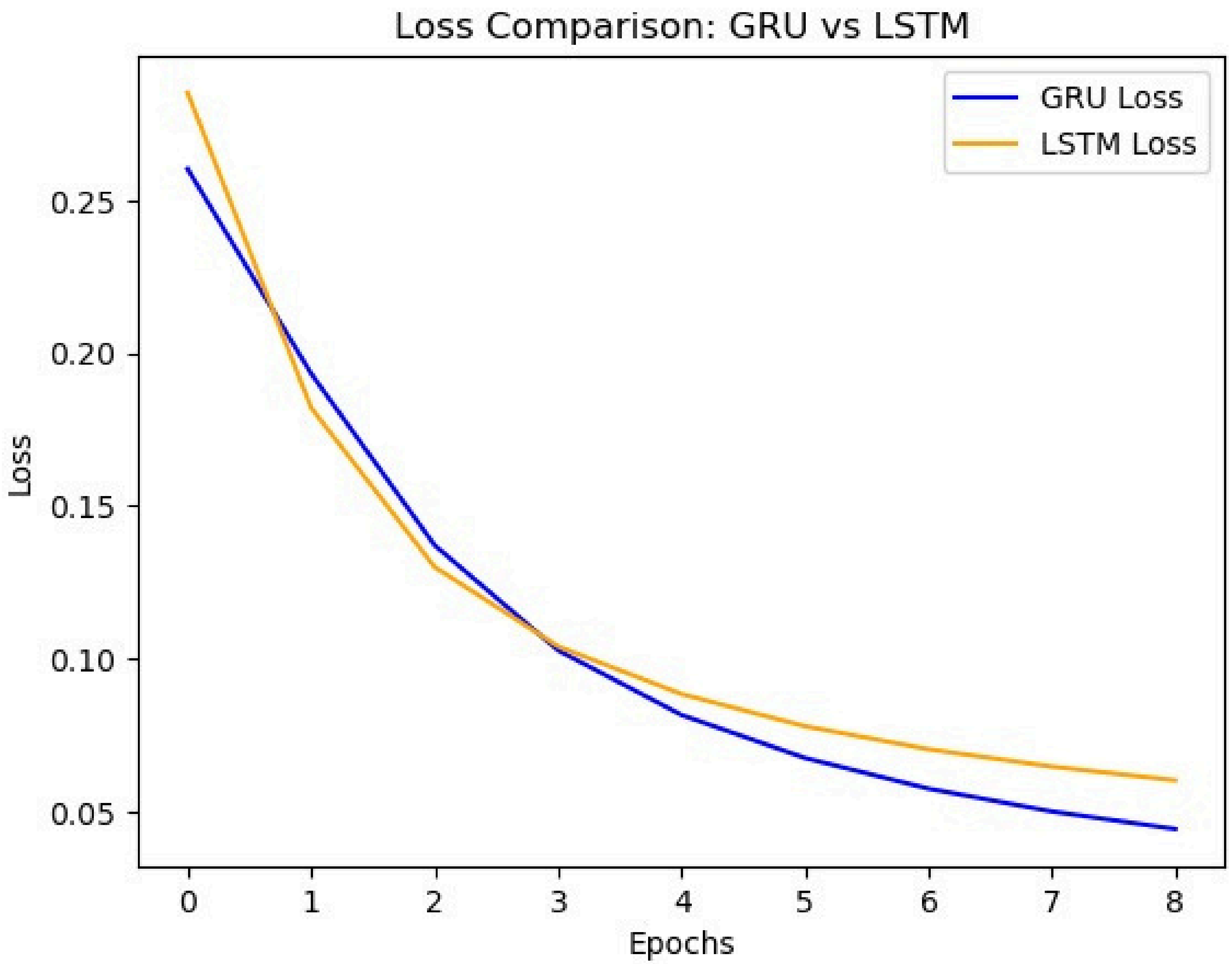
Model Evaluation











Loss Comparison: GRU vs LSTM
GRU Loss
LSTM Loss
Loss
0.25
0.20
0.15
0.10
0.05
0
1
2
3
4
5
6
7
8
Epochs

FUTURE SCOPE

Context-Aware Ranking Models

Develop advanced ranking models integrating contextual data and user behavior.

Hybrid Filtering Techniques

Explore hybrid filtering systems combining collaborative and content-based techniques.

Dynamic Feedback Integration

Incorporate real-time feedback to dynamically update user preferences.

Enhanced Evaluation Metrics

Investigate methods like Manhattan Distance or advanced sequence alignment metrics for improved evaluation.

NOVELTY

1.**Advanced Embedding Techniques**: Integration of BERT embeddings for deep semantic understanding of video metadata, surpassing traditional methods like TF-IDF or Word2Vec.
2. **Clustering for Efficiency**: Embedding-based clustering narrows the candidate pool, reducing computational overhead and aligning recommendations with user preferences.
3. **Dynamic User Preferences**: Real-time updates to user embeddings based on interactions ensure evolving personalization.
4. **LSTM vs. GRU Analysis**: Comparative evaluation of LSTM and GRU models highlights trade-offs between accuracy and efficiency for sequential data modeling.
5. **Hybrid Feature Representation**: Combines TF-IDF for initial weighting and BERT for contextual understanding, balancing computational cost and accuracy.
6. **Cold-Start Solution**: Content-based filtering handles sparse user data, making the system effective for new users or niche content.
7. **Automated Data Pipeline**: Efficient data collection via APIs, robust preprocessing, and normalization ensure high-quality feature extraction.
8. **Cluster-Based Dynamic Ranking**: Refines recommendations within clusters based on relevance and popularity scores.
9. **Visualization of Clusters**: PCA-based visualizations validate embedding clustering and user alignment.
10. **Future-Ready Design**: Plans to integrate collaborative filtering, advanced ranking models, and hybrid techniques for scalability and improved performance.

This combination of advanced methods, personalization, and efficiency sets your system apart from traditional recommendation approaches.

REFERENCES

1. Deep Neural Networks for YouTube Recommendations Paul Covington, Jay Adams, Emre Sargin Google Mountain View, CA {pcovington, jka, msargin}@google.com . https://research.google.com/pubs/archive/45530.pdf
2. The Netflix Recommender System: Algorithms, Business Value, and Innovation CARLOS A. GOMEZ-URIBE and NEIL HUNT, Netflix, Inc. https://scholar.google.co.in/scholar_url?url=https://dl.acm.org/doi/pdf/10.1145/2843948&hl=en&sa=X&ei=E1VNZ96vMKOx6rQPwdys2AM&scisig=AFWwaeZ-zjVIomh-YnXSNviKmWCo&oi=scholarr
3. Video Content-Based Advertisement Recommendation System using Classification Technique of Machine Learning To cite this article: R C Konapure and L M R J Lobo 2021 J. Phys.: Conf. Ser. 1854 012025 https://iopscience.iop.org/article/10.1088/1742-6596/1854/1/012025/pdf#:~:text=Content%2Dbased%20advertising%20helps%20to,a%20relevant%20advertisement%20to%20it
4. Research on Collaborative Filtering Personalized Recommendation Algorithm Based on Deep Learning Optimization Guo Wei-wei, Liu Feng Heilongjiang University of Technology, JiXi,158100, China gwwguoweiwei@163.com https://ieeexplore.ieee.org/document/8806589

5. Artificial intelligence in recommender systems Qian Zhang1 ·Jie Lu1 · Yaochu Jin2
https://link.springer.com/article/10.1007/s40747-020-00212-w
6. Social Recommendation for Social Networks Using Deep Learning Approach: A Systematic Review, Taxonomy, Issues, and Future Directions
MUHAMMADALRASHIDI 1,ALI SELAMAT 1,2,3,4, (Member, IEEE), ROLIANA IBRAHIM1, (Member, IEEE), AND ONDREJ KREJCAR 3,4
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10128133&tag=1
7. Content Based Recommendation System on Netflix Data Dr. Deepti Sharma1, Dr. Deepshikha Aggarwal2*, Dr. Archana B. Saxena3 1,2*,3Professor.
https://www.researchgate.net/publication/378337926_Content_Based_Recommendation_System_on_Netflix_Data/fulltext/65eca65bb7819b433bf204d3/Content-Based-Recommendation-System-on-Netflix-Data.pdf?origin=publication_detail&_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6InB1YmxpY2F0aW9uIiwicGFnZSI6InB1YmxpY2F0aW9uRG93bmxvYWQiLCJwcmV2aW91c1BhZ2UiOiJwdWJsaWNhdGlvbiJ9fQ&__cf_chl_tk=sSAuZxi7Oli29CuWfgDvGAfwhsoBOxzFLPeKk347NEM-1733245145-1.0.1.1-lk96Br65Tv2jlICNxxsv9FDQAOqUSeS_nok6pyMHmvw
8. Pazzani, M. J., & Billsus, D. (2007). Content-Based Recommendation Systems.https://link.springer.com/chapter/10.1007/978-3-540-72079-9_10

THANK YOU