AN IDP PROJECT REPORT

on

"TEDTALKS RECOMMENDATION USING CONTENT-BASED FILTERING"

Submitted

By

221FA04270

Vishnu Vardhan.K

221FA04312

Lakshmi Narayana.S

221FA04325

Adithya.V

221FA04425

Charishma.S

Under the guidance of

Mr.Sourav Mondal

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING VIGNAN'S

FOUNDATION FOR SCIENCE, TECHNOLOGY AND RESEARCH

(Deemed to be UNIVERSITY) Vadlamudi, Guntur.

ANDHRA PRADESH, INDIA, PIN-522213.

VIGNAN'S
FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH
(Deemed to be University) - Estd. u/s 3 of UGC Act 1956

CERTIFICATE

This is to certify that the Field Project entitled **"TEDTALKS Recommendation Using Content-Based Filtering"** that is being submitted by 221FA04270 (Vishnu Vardhan.K), 221FA04312(Lakshmi Narayana.S),221FA04325(Adithya.V), 221FA04425(Charishma.S),for partial fulfilment of Field Project is a bonafide work carried out under the supervision of Mr.Sourav Mondal , Assistant Professor ,Department of CSE.

Mr. Sourav Mondal
Guide name& Signature

Assistant Professor,
CSE

Dr. S. V. Phani Kumar
HOD,CSE

Dr.K.V. Krishna Kishore
Dean, SoCI

VIGNAN'S
FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH
(Deemed to be University) - Estd. u/s 3 of UGC Act 1956

DECLARATION

We hereby declare that the Field Project entitled "**TEDTLKS RECOMMENDATION USING CONTENT-BASED FILTERING"** that is being submitted by 221FA04270 (Vishnu Vardhan.K), 221FA04312(Lakshmi Narayana.S), 221FA04325(Adithya.V) and 221FA04425(Charishma.S) in partial fulfilment of Field Project course work. This is our original work, and this project has not formed the basis for the award of any degree. We have worked under the supervision of Mr.Sourav Mondal, Assistant Professor, Department of CSE.

By

221FA04270(Vishnu Vardhan.K),

221FA04312(Lakshmi Narayana.S),

221FA04325(Adithya.V),

221FA04425(Charishma.S)

Date : 1-4-2025

ABSTRACT

With the exponential growth of online content, personalized recommendation systems have become essential in enhancing user experience. This paper presents a content-based filtering approach to recommend TED Talks to users based on their preferences. The system leverages Natural Language Processing (NLP) techniques to analyze TED Talk transcripts, descriptions, and metadata, extracting key features such as topics, keywords, and speaker information. Using Term Frequency-Inverse Document Frequency (TF-IDF) and cosine similarity, the system matches users' past preferences with relevant TED Talks. Unlike collaborative filtering, which relies on user interactions, content-based filtering ensures personalized recommendations even for new or niche users. Experimental results demonstrate improved relevance in recommendations, showcasing the effectiveness of content-based methods for TED Talk personalization. This study highlights the potential of machine learning in enhancing content discovery, offering an efficient solution for personalized learning and entertainment.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

1.INTRODUCTION

1.1 Background and Significance of TEDTalks Recommendation using content-based filtering

Background:

TED (Technology, Entertainment, and Design) Talks are influential presentations covering a wide range of topics, including science, technology, business, psychology, and personal development. With thousands of TED Talks available, it becomes essential to recommend relevant talks to users based on their interests and preferences.Content-based filtering recommends TED Talks by analyzing their textual and metadata features, such as titles, descriptions, tags, speaker names, and transcripts. It extracts features using TF-IDF (Term Frequency-Inverse Document Frequency) or word embeddings (Word2Vec, BERT, etc.) to represent the textual content numerically. The system then computes cosine similarity or other distance metrics to find talks with similar content and recommends them based on a user's viewing history or preferences. This approach ensures personalized recommendations without relying on other users' behavior, making it effective even with a small user base.

FIGURE 1.1-TEDTALKS

Significance:

The significance of the TED Talks recommendation system using content-based filtering lies in its ability to enhance user engagement by providing personalized and relevant recommendations based on talk content rather than user behavior. This approach ensures that users discover insightful talks aligned with their interests, improving their learning experience. Unlike collaborative filtering, it does not rely on other users' preferences, making it effective even for niche topics. Additionally, by leveraging Natural Language Processing (NLP) and machine learning, the system can analyze vast amounts of text data, ensuring accurate and meaningful suggestions. This helps users explore a diverse range of TED Talks, fosters knowledge expansion, and makes content discovery more efficient, ultimately improving the accessibility and impact of educational resources.

1.2 Overview of Machine Learning in TedTalks Recommendation using content-based filtering:

1**. Data Processing and Feature Extraction**

ML techniques help extract meaningful insights from TED Talk metadata, such as titles, descriptions, tags, and transcripts:

- TF-IDF (Term Frequency-Inverse Document Frequency): Identifies important words in a document.
- Word Embeddings (Word2Vec, BERT, GloVe): Captures the contextual meaning of words.
- Latent Semantic Analysis (LSA): Finds hidden relationships between words and topics.
- Named Entity Recognition (NER): Identifies important entities like people, places, and concepts in the talk.
- Topic Modeling (LDA - Latent Dirichlet Allocation): Categorizes TED Talks into thematic topics.

FIGURE 1.2-TEDTALKS COMMUNICATION

2. Similarity Computation and Recommendation Generation

ML-based algorithms identify and recommend similar TED Talks using:

- Cosine Similarity: Measures how similar two TED Talks are based on vectorized features.
- Euclidean Distance: Computes the difference between feature vectors to rank similar talks.
- Deep Learning Models: Transformer-based models (e.g., BERT, GPT) enhance content understanding.
- Hybrid Approaches: Combining content-based filtering with collaborative filtering to improve recommendations.

3**. Personalization and Adaptive Learning**

ML continuously improves and personalizes recommendations by learning from user interactions:

- User Behavior Analysis: Tracks watch history, likes, and duration watched.
- Reinforcement Learning: Adapts recommendations based on user feedback.
- Implicit Feedback Learning: Considers browsing patterns and time spent on recommended talks.
- Graph-Based Recommendations: Uses knowledge graphs to connect related TED Talks.

4. **Advantages of ML in TED Talks Recommendation**

- Scalable & Efficient – Processes large amounts of TED Talk data quickly.
- Accurate & Context-Aware – NLP-based models enhance understanding of talk content.
- Dynamic Personalization – Continuously adapts to user preferences.
- Cross-Domain Recommendations – Suggests TED Talks from different but relevant categories.

1.3 Research Objectives and Scope

Research Objectives

The primary objectives of this research are to:

1. Develop a Personalized TED Talks Recommendation System:
   Create a content-based filtering system that analyzes TED Talk metadata (titles, descriptions, transcripts) and user behavior to deliver personalized recommendations.
2. Enhance Recommendation Accuracy:
   Improve recommendation precision by utilizing Machine Learning (ML) techniques such as

3. Natural Language Processing (NLP), TF-IDF, Word2Vec, and BERT to better understand the content of TED Talks and predict user preferences.
4. Analyze User Interaction Patterns:
   Investigate how user interactions (watch history, ratings, search behavior) can be leveraged to refine and adapt recommendations over time, improving the overall user experience.
5. Combine Content-Based Filtering with Hybrid Approaches:
   Explore the integration of content-based filtering with collaborative filtering and knowledge graphs to provide more diverse and accurate TED Talk suggestions.
6. Address Challenges in Recommendation Systems:
   Identify and tackle issues such as the cold start problem, overfitting, and lack of content diversity in TED Talk recommendations through innovative machine learning strategies.
7. Evaluate System Performance:
   Assess the effectiveness of the developed recommendation system using key performance metrics such as precision, recall, F1-score, and user satisfaction.

Scope of Research

This research focuses on the development and evaluation of a content-based recommendation system for TED Talks. The scope includes:

1. **Dataset:**
   The dataset will include TED Talk metadata such as talk titles, descriptions, tags, transcripts, and speakers. Additionally, user interaction data (e.g., view history, ratings, clicks) will be analyzed to refine recommendations.
2. **Recommendation** Techniques:
   The research will primarily focus on content-based filtering with support from hybrid approaches that combine collaborative methods and knowledge graphs. It will also explore advanced ML techniques, such as Deep Learning for better semantic analysis of TED Talks.
3. **User** Feedback:
   The research will examine how implicit feedback (e.g., watch duration, clicks, likes) and explicit feedback (e.g., ratings, comments) can be used to improve recommendations.
4. **Evaluation** Metrics:
   The system's performance will be measured using standard metrics such as accuracy, F1-score, precision, and recall to determine how well the recommendations align with users' interests and preferences.
5. **Limitations:**
   The research will primarily focus on TED Talks, and its findings may not be directly applicable to other domains like movies, books, or e-commerce. The study will also be limited by the available user interaction data and content features.

1.4 Current Challenges :

1. Cold Start Problem

New users or new TED Talks with no prior interaction history face difficulties in receiving accurate recommendations. The system requires enough data to make personalized suggestions, which is challenging for new users or newly added content.

2. Lack of Content Diversity

Content-based filtering tends to recommend similar content based on past behavior, which can lead to a filter bubble. This limits exposure to new topics and reduces content diversity, restricting users' learning opportunities.

3. Feature Extraction and Representation

Extracting meaningful features from TED Talk metadata like titles, descriptions, and transcripts is complex. Common techniques such as TF-IDF and Word2Vec may not capture all nuances of the content, especially in multilingual or specialized talks.

4. Scalability

As the number of TED Talks grows, the system must scale to process large datasets efficiently. The challenge lies in ensuring that the system can handle increased content volume without compromising performance or response time.

5. Personalization Over Time

User preferences evolve over time, and the system needs to adapt accordingly. Continuously improving recommendations based on implicit and explicit feedback is complex but necessary for maintaining relevance in long-term user engagement.

6. Handling Multi-Topic and Cross-Domain Content

TED Talks often span multiple topics, making it difficult to generate cross-domain recommendations. Understanding relationships between topics and offering relevant suggestions across domains is a challenge for content-based systems.

7. Lack of Rich User Data

The recommendation system relies heavily on user interaction data, but many users may not provide enough feedback. Without detailed data, the system struggles to make accurate predictions about a user's preferences.

8. Evaluation and Performance Metrics

Traditional metrics like precision and recall don't always capture user satisfaction or engagement. More comprehensive, user-centric metrics like engagement time and satisfaction are needed to fully evaluate system effectiveness.

1.5 Applications of ML

1. **Personalized Recommendations**

ML algorithms analyze user behavior and preferences to provide personalized recommendations. By tracking user interactions such as watch history, clicks, and likes, the system learns to predict what talks users are most likely to enjoy. For example, if a user consistently watches technology-related talks, the system will prioritize recommending more TED Talks in this domain.

2. **Content-Based Filtering**

In content-based filtering, ML algorithms are used to analyze the textual content of TED Talks (such as titles, descriptions, and transcripts) and match it with user interests. Techniques like TF-IDF (Term Frequency-Inverse Document Frequency) and Word2Vec are used to extract features from the talks, allowing the system to find and recommend similar content based on the talk's thematic elements, keywords, and topics.

3. Natural Language Processing (NLP) for Feature Extraction

ML models that use Natural Language Processing (NLP) help process and understand the text data within TED Talk descriptions, titles, and transcripts. By applying NLP techniques like BERT and Latent Semantic Analysis (LSA), the system can effectively understand the context and meaning behind the words used in TED Talks, improving the quality of content recommendations.

4. **User Behavior Modeling**

ML algorithms can build models of user behavior by analyzing implicit feedback (like time spent watching a video) and explicit feedback (like ratings or likes). Reinforcement Learning can then be applied to optimize recommendations based on how users interact with suggested content. This leads to better predictions and a more personalized user experience.

5. **Dynamic Learning and Adaptation**

One of the core applications of ML is adaptive learning, where the system continuously improves its recommendations based on ongoing user interactions. By tracking real-time feedback, the system can adjust its recommendation strategy, making suggestions more relevant as users' preferences evolve over time. Reinforcement Learning allows the system to adjust the weighting of different features, learning from past mistakes and improving its accuracy.

6. **Hybrid Recommendation Models**

Combining content-based filtering with collaborative filtering techniques creates a hybrid recommendation system. ML algorithms are used to merge data about user preferences with content features, offering recommendations based on both user similarities and content similarities. This hybrid approach helps tackle limitations like the cold start problem and ensures diverse recommendations.

7. **Handling Large-Scale Data**

ML models can handle large-scale datasets efficiently, which is essential as TED Talks continually grow. By using clustering algorithms, the system can group TED Talks into categories and subcategories, making it easier to navigate through large amounts of content. Dimensionality reduction techniques like Principal Component Analysis (PCA) can also be applied to reduce the complexity of the data while retaining meaningful features.

8**. Improving Diversity in Recommendations**

ML techniques can be used to introduce more diversity into the recommendations. By combining algorithms such as content-based filtering with randomization techniques or by applying topic modeling, the system can suggest talks that are slightly outside the user’s usual preferences but still related enough to encourage exploration. This prevents the system from recommending overly repetitive content and helps expose users to a wider range of TED Talks.

9. **Topic and Sentiment Analysis**

ML can be used for topic modeling to identify the central themes of each TED Talk and assign appropriate tags. Sentiment analysis can be applied to evaluate the tone or emotional content of the talk, which can then influence the recommendations (e.g., suggesting motivational or uplifting talks to users based on their emotional preferences).

10. Performance Evaluation and Feedback Loop

ML is applied to evaluate the effectiveness of the recommendation system by measuring key performance metrics such as precision, recall, F1-score, and user engagement. These evaluations allow the system to learn from the data and optimize future recommendations, creating a feedback loop where the system continually improves its accuracy and relevance.

CHAPTER-2

LITERATURE SURVEY

2.1 Literature review

1. Online Course Recommendation System

- **Dataset:** Open Online Courses dataset (22,144 records, 10,000 unique users).
- **Approach:** Collaborative filtering, Apriori Algorithm, SPADE Algorithm, K-Means Clustering, Self -Organizing Maps (SOM).
- **Limitations:** Scalability issues, cold start problem, lack of personalization, fixed cluster size (K-Means).
- **Accuracy:**
  - Apriori Algorithm: 0.268 to 0.374
  - SPADE Algorithm: 0.360 to 0.402

2. Healthcare Recommendation System

- **Dataset:** No specific dataset.
- **Approach:** Matrix Factorization Model (MFM), Singular Value Decomposition (SVD), Multilayer Perceptron (MLP) with Autoencoder, CNN, Content-Based Filtering (CBF), Collaborative-Based Filtering
  Hybrid Filtering.
- **Limitations:** Lack of real-time adaptation.
- **Accuracy:**
  - MFM: 0.089 to 0.064
  - SVD: 0.079 to 0.054

3. Amazon Product Recommendation System

- **Dataset:** Amazon product dataset.
- **Approach:** Collaborative filtering, Alternating Least Squares (ALS), Singular Value Decomposition (SVD), Spark ML Framework.
- **Limitations:** Computational overhead.
- **Accuracy (RMSE):**
  - ALS: 0.866 – 1.247
  - SVD: 1.098

4. Crop Recommendation System

- **Dataset:** Kaggle agricultural dataset (soil type, pH levels, temperature, rainfall, humidity).
- **Approach:** Decision Tree (DT), Random Forest (RF), Logistic Regression (LR), CatBoost Classifier (CC).
- **Limitations:** Weather variability, limited regional data.
- **Accuracy:** RF: 99.09% (best performance).

5. Travel Recommendation System (Image-Based)

- **Dataset:** Tourism image datasets (destination images, location metadata, weather conditions).
- **Approach:** Image recognition with OpenCV, Collaborative & Content-Based Filtering, Weather API Integration.
- **Limitations:** Limited landmark recognition, static data dependency.

6. Nutrition Recommendations for Menstruating Women

- **Dataset:** Medical and nutrition databases (hormonal cycles, food benefits, health conditions).
- **Approach:** Content-Based & Collaborative Filtering, Blockchain for secure data storage, Hybrid Recommendation Algorithm.
- **Limitations:** User privacy concerns, limited generalization.

- **Accuracy:** Hybrid model achieved 91%.

7. Music Playlist Recommendation

- **Dataset:** Kaggle music dataset (user listening history, genre, artist, song attributes).
- **Approach:** Collaborative Filtering (User & Item-Based), Content-Based Filtering (Genre, artist similarity), Hybrid Ensemble Model.
- **Limitations:** Popularity bias.

8. Movie Recommendation System

- **Dataset:** IMDb movie dataset (6 million entries, movie ratings, genres, cast, director, user reviews).
- **Approach:** Content-Based Filtering (Cosine Similarity), K-Nearest Neighbor (KNN) for genre correlation, Sentiment Analysis (Naïve Bayes Classifier).
- **Limitations:** Sentiment misinterpretation.

9. Fashion Recommendation using GANs

- **Dataset:** E-commerce fashion dataset (67,000 shoe images, product images, metadata, customer preferences).
- **Approach:** GAN-Based Image Retrieval, Deep Feature Extraction (VGGNet), Content-Based Filtering for similarity matching.
- **Limitations:** Lack of interpretability, computationally intensive.

10. AUTOGENIE – Vehicle Selection System

- **Dataset:** No specific dataset.
- **Approach:** Hybrid ML-based approach, Matrix Factorization (MF), Deep Matrix Factorization (DMF), Hybridization.
- **Limitations:** Data integration issues, computational complexity, risk of overfitting.
- **Accuracy:** DMF improved accuracy over traditional MF.

2.2 Motivation

1. Enhancing Content Discovery

One of the main motivations is to enhance content discovery. Without a robust recommendation system, users may miss out on valuable TED Talks that align with their interests or needs. A recommendation system would help users discover new talks that they might not have found on their own, ensuring that they are exposed to diverse and relevant content that enhances their learning experience.

2. Personalization of User Experience

Another key motivation is personalization. Different users have varying interests, preferences, and learning goals. By implementing a personalized recommendation system, the project aims to provide each user with content that is tailored to their specific preferences, creating a more engaging and enriching experience. The ability to suggest talks based on users' past interactions can make the TED platform feel more intuitive and user-centric.

3. Overcoming Content Overload

As the number of TED Talks increases, users may experience content overload—the difficulty in filtering and finding talks that are most relevant. A recommendation system powered by machine learning can solve this problem by effectively narrowing down options and offering only the most relevant content, which ultimately helps users save time and find content that best suits their needs.

4. Facilitating Lifelong Learning

The TED platform serves as a hub for lifelong learning, offering educational resources to individuals across the world. A well-designed recommendation system motivates continued learning by constantly suggesting new and relevant content, helping users grow intellectually and expand their knowledge on various topics. The personalized recommendations ensure that the learning experience remains dynamic and adaptable to users' evolving interests.

5. Improving User Engagement

Another motivating factor is user engagement. A personalized and accurate recommendation system can increase user retention and encourage users to interact with the platform more frequently. By offering suggestions that are closely aligned with a user's interests, the system enhances user satisfaction, fostering a deeper connection with the platform.

6. Solving the Cold Start Problem

The cold start problem, where new users or content struggle to receive accurate recommendations, is a common challenge in many recommendation systems. The motivation behind the project is to find solutions to this issue, whether through hybrid recommendation models or the use of metadata, to ensure that both new users and new TED Talks can be recommended effectively.

7. Broader Impact on Education

Finally, the project is motivated by the broader impact on education. By ensuring that users can easily access TED Talks that resonate with their personal or professional growth, the project aims to support knowledge dissemination and contribute to a global learning community. A strong recommendation system not only helps individuals but also contributes to making educational content accessible to diverse audiences worldwide.

CHAPTER-3

PROPOSED SYSTEM

1. Dataset Collection

The dataset utilized for this study is sourced from the TED Talks dataset, comprising 2,550 TED Talks with various attributes such as title, description, tags, speaker information, views, and duration. The dataset is stored in a structured format (**CSV file**) and contains the following key attributes:

- **Textual Attributes:** title, description, tags
- **Categorical Attributes:** speaker_occupation
- **Numerical Attributes:** views, duration

This dataset serves as the foundation for content-based filtering by providing detailed metadata about each TED Talk.

2. Data Preprocessing

To ensure high-quality feature representation, multiple preprocessing steps are applied:

2.1 Handling Missing Data

- Missing values in the speaker_occupation column are filled using the most frequent occupation associated with the speaker.
- Other columns do not have missing values and are retained as is.

2.2 Text Cleaning and Normalization

- **Tokenization:** The textual data (title, description, and tags) are split into individual words.
- **Lowercasing:** Converts all words to lowercase for uniformity.
- **Stopword Removal:** Eliminates common stopwords such as "the", "is", and "and" to reduce noise.
- **Lemmatization:** Converts words to their base form (e.g., "running" → "run") to ensure consistency in textual representation.

2.3 Feature Selection

- **Selected features**: title, description, tags, speaker_occupation, views, and duration.
- Non-informative attributes such as index are dropped.

3. Feature Extraction

Each TED Talk is represented using textual, categorical, and numerical features.

3.1 Text Feature Representation

To convert textual features into numerical representations:

- **TF-IDF (Term Frequency-Inverse Document Frequency)** is applied to title, description, and tags to extract relevant keywords.
- **Alternative Approach:** Pre-trained word embeddings such as Word2Vec, FastText, or BERT can be utilized for deeper semantic understanding.

Mathematically, TF-IDF is computed as:

where:

- is the term frequency of word in document .
- is the document frequency of .
- is the total number of documents.

3.2 Categorical Feature Encoding

- **One-Hot Encoding** is applied to speaker_occupation, converting categorical labels into numerical representations.

3.3 Numerical Feature Scaling

- Min-Max Scaling is used to normalize views and duration within the range [0,1].

The transformed dataset ensures that all features are in a comparable numerical format.

4. Feature Representation

After feature extraction, each TED Talk is represented as a feature vector:

where:

- represents textual features from TF-IDF or embeddings.
- represents categorical encodings.
- represents normalized numerical features.

These representations form the basis for similarity computations.

5. Similarity Computation

To recommend similar TED Talks, a similarity score is computed based on three feature types.

5.1 Text Similarity

- Cosine Similarity is applied to text feature vectors to measure content similarity.

Where TF-IDF vectors of two TED Talks.

5.2 Categorical Similarity

- Jaccard Similarity is used to measure overlap between categorical features such as tags and speaker_occupation.

where and are sets of categorical values.

5.3 Numerical Feature Similarity

- Euclidean Distance is used to compare views and duration.

where and are numerical feature vectors.

- **Combined Similarity Score**

A weighted combination of text, categorical, and numerical similarity scores is computed:

where:

- is the cosine similarity of textual features.
- is the Jaccard similarity of categorical features.
- is the inverse of the Euclidean distance for numerical features.
- are hyperparameters optimized using Grid Search.

6. Generating Recommendations

To generate recommendations:

1. Similarity scores are computed for all TED Talks.
2. TED Talks are ranked in descending order based on the similarity score.
3. The Top-N most similar TED Talks are returned as recommendations.

The system ensures that users receive personalized TED Talk recommendations based on the content and attributes of the talks.

3.1 Input dataset

The dataset utilized for this study is sourced from the TED Talks dataset, comprising 2,550 TED Talks with various attributes such as title, description, tags, speaker information, views, and duration. The dataset is stored in a structured format (CSV file)

3.1.1 Detailed Features of the Dataset

Description:
Provides a brief summary or an overview of the TED Talk, giving insights into the key points or themes covered in the talk.

Duration:
Represents the length of the TED Talk, typically measured in minutes, indicating how long the talk lasts.

Main Speaker:
Identifies the individual giving the TED Talk, representing the primary speaker or presenter.

Name:
Refers to the name of the TED Talk, usually a title that describes the main subject or idea discussed in the presentation.

Speaker Occupation:
Indicates the professional background or occupation of the main speaker, such as scientist, educator, entrepreneur, etc.

Tags:
A set of keywords or labels that categorize the TED Talk into themes or topics (e.g., technology, education, science, personal development).

Title:
The title of the TED Talk, often chosen to capture the essence of the talk and attract viewers.

Views:
Represents the number of views or the level of engagement the TED Talk has received, serving as a metric of its popularity or reach.

Index:
A unique identifier or index that distinguishes each TED Talk in the dataset, allowing for efficient data management and retrieval.

3.2Data Preprocessing

To ensure high-quality feature representation, multiple preprocessing steps are applied:

3.2.1 Handling Missing Data

- Missing values in the speaker_occupation column are filled using the most frequent occupation associated with the speaker.
- Other columns do not have missing values and are retained as is.

3.2.2 Text Cleaning and Normalization

- **Tokenization:** The textual data (title, description, and tags) are split into individual words.
- **Lowercasing:** Converts all words to lowercase for uniformity.

- **Stopword Removal:** Eliminates common stopwords such as "the", "is", and "and" to reduce noise.
- **Lemmatization:** Converts words to their base form (e.g., "running" → "run") to ensure consistency in textual representation.

3.3 Model Building

The model-building process for the TED Talks recommendation system involves several steps, from data preprocessing to implementing machine learning algorithms for personalized recommendations. Below is a structured approach to developing the model:

1. **Data Preprocessing**

- Cleaning the Data: Handle missing values, remove duplicates, and standardize text formats.
- Text Processing: Apply Natural Language Processing (NLP) techniques such as tokenization, stopword removal, stemming, and lemmatization to process textual features like description, title, and tags.
- Feature Encoding: Convert categorical features (e.g., speaker occupation) into numerical representations using One-Hot Encoding or Label Encoding.
- Normalization: Standardize numerical features like duration and views to ensure uniform scaling.

2. Feature Engineering

- TF-IDF Vectorization: Transform textual features (description, title, tags) into numerical vectors for similarity analysis.
- Embedding Techniques: Use Word2Vec, BERT, or Doc2Vec for deeper semantic understanding of the talks.
- Topic Modeling: Apply Latent Dirichlet Allocation (LDA) to group talks based on underlying topics.

3. Model Selection

- Content-Based Filtering: Recommends TED Talks by measuring similarity between talks using Cosine Similarity or Euclidean Distance.
- Hybrid Model: Combines content-based filtering with collaborative filtering to overcome limitations like the cold start problem.

4. Similarity Computation

- Use Cosine Similarity or Jaccard Similarity to find similar TED Talks based on text embeddings from the description, tags, and title.
- Generate a recommendation matrix by ranking similar TED Talks for each input talk.

5. Recommendation Generation

- For a given TED Talk, retrieve top-N most similar talks based on similarity scores.
- Display recommendations in ranked order, ensuring diversity by adjusting similarity thresholds.

6. Model Evaluation

- Precision, Recall, and F1-Score: Measure the accuracy of recommended TED Talks.
- Mean Average Precision (MAP): Evaluates the relevance of recommendations.
- User Engagement Metrics: Analyze watch time, click-through rates, and interaction patterns to assess model performance.

3.4 Methodology of the system

The methodology for developing the TED Talks recommendation system follows a structured approach, integrating data preprocessing, feature extraction, model selection, and evaluation to deliver accurate and personalized recommendations. The key steps in the methodology are as follows:

1. Data Collection and Preprocessing

- Load the dataset containing TED Talk features such as title, description, duration, main speaker, speaker occupation, tags, views, and index.
- Handle missing values, remove duplicates, and clean text data by applying stopword removal, stemming, and lemmatization.
- Convert categorical variables (e.g., speaker occupation) into numerical format using One-Hot Encoding or Label Encoding.
- Normalize numerical features like duration and views to ensure uniform scaling.

2. Feature Extraction

- Apply TF-IDF (Term Frequency-Inverse Document Frequency) to convert text features (title, description, tags) into numerical vectors.
- Use Word2Vec, BERT, or Doc2Vec embeddings to capture semantic relationships between words.

- Perform topic modeling (Latent Dirichlet Allocation - LDA) to group TED Talks into different themes.

3. Model Selection

- Implement Content-Based Filtering by calculating similarity between TED Talks based on textual features using Cosine Similarity.
- If needed, integrate Hybrid Filtering by combining content-based recommendations with collaborative filtering to improve diversity and mitigate the cold start problem.

4. Recommendation Generation

- Compute similarity scores between TED Talks using Cosine Similarity or Jaccard Similarity on extracted text features.
- Generate a recommendation matrix to rank the most similar TED Talks based on user input.
- Apply diversity enhancement techniques to avoid recommending overly similar content repeatedly.

5. Model Evaluation and Optimization

- Use evaluation metrics like Precision, Recall, F1-Score, and Mean Average Precision (MAP) to measure recommendation accuracy.
- Analyze user engagement metrics such as watch time, click-through rate (CTR), and user feedback to refine recommendations.
- Optimize the model by fine-tuning hyperparameters and applying feedback loops for continuous learning.

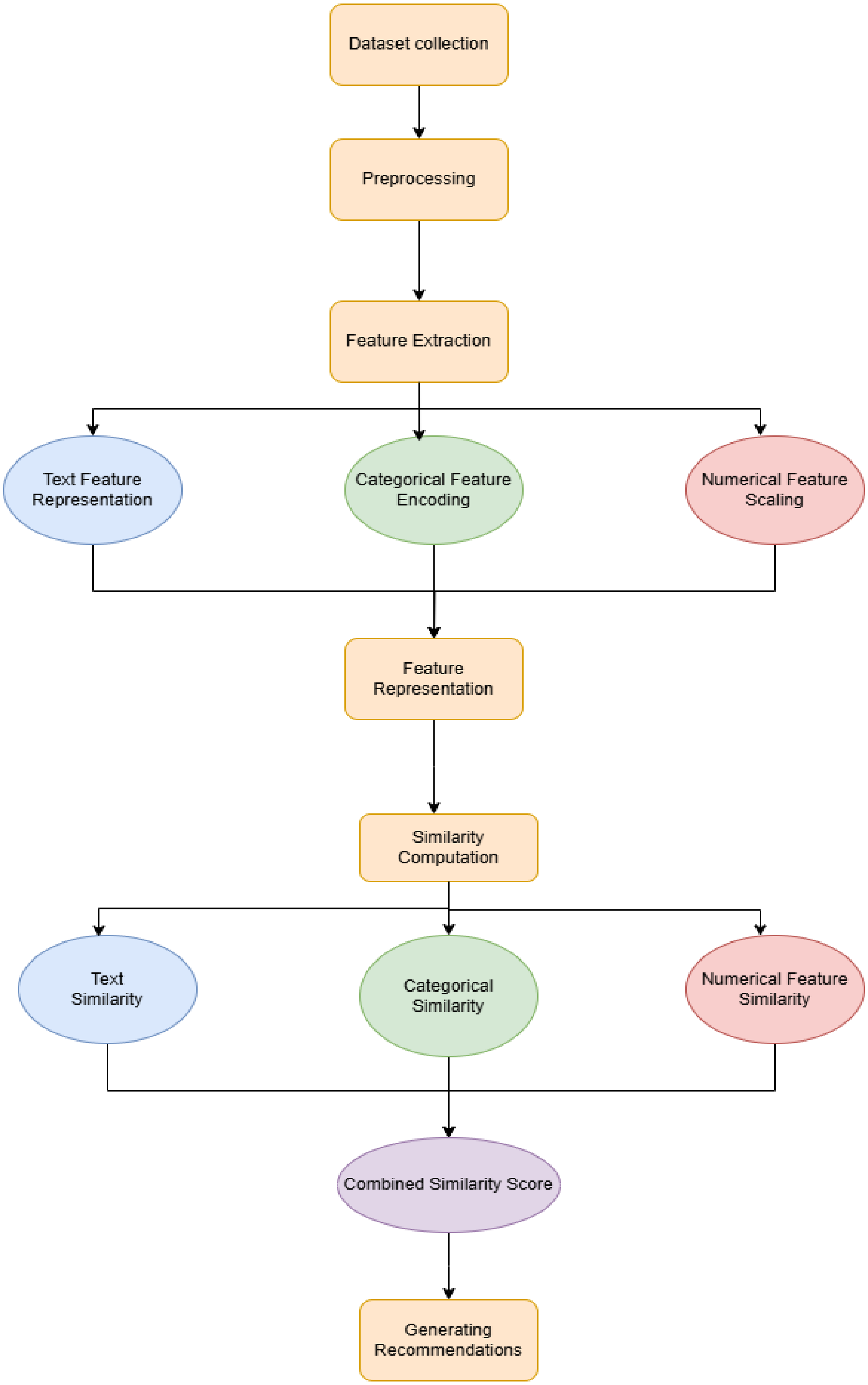

Dataset collection
Preprocessing
Feature Extraction
Text Feature Representation
Categorical Feature Encoding
Numerical Feature Scaling
Feature Representation
Similarity Computation
Text Similarity
Categorical Similarity
Numerical Feature Similarity
Combined Similarity Score
Generating Recommendations

FIGURE 3.4-METHODOLOGY

3.5 Model Evaluation

The system was evaluated using various machine learning models, including:

- **Support Vector Classifier (SVC)**
- **Logistic Regression**
- **Random Forest**
- **XGBoost**
- **Gradient Boosting**

To assess the performance of these models, we analyzed their **confusion matrices**, which provide insights into the classification results by categorizing predictions into:

- **True Positives (TP)**: Relevant TED Talks correctly recommended.
- **False Positives (FP)**: Irrelevant TED Talks incorrectly recommended.
- **True Negatives (TN)**: Irrelevant TED Talks correctly not recommended.
- **False Negatives (FN)**: Relevant TED Talks that were not recommended.

3.6 Constraints

1. Data Availability and Quality

- Limited metadata for some TED Talks may reduce recommendation accuracy.
- Incomplete or missing speaker information, tags, or descriptions can lead to poor feature extraction.
- Noise in textual data (e.g., informal language, abbreviations) may affect NLP-based feature representation.

2. Cold Start Problem

- New users with no interaction history may receive generic recommendations.
- Newly added TED Talks may not have enough engagement data to be effectively recommended.

3. Computational Complexity

- Processing a large dataset of TED Talks requires significant computational resources.
- Algorithms like TF-IDF, Word2Vec, and Cosine Similarity can be computationally expensive for real-time recommendations.

4. Scalability Issues

- As the number of TED Talks increases, the recommendation system must scale efficiently.
- Storage and memory constraints may arise when handling large feature matrices for similarity computation.

5. Over-Specialization of Recommendations

- Content-based filtering may lead to a filter bubble, repeatedly suggesting similar types of talks.
- Lack of diversity in recommendations can limit user exploration of new topics.

6. User Preference Changes Over Time

- The system may struggle to adapt if users' interests shift over time.
- Static recommendations may not reflect evolving user preferences without adaptive learning mechanisms.

7. Multilingual Challenges

- TED Talks in different languages require language-specific NLP models, increasing complexity.
- Translating and normalizing textual data across multiple languages may lead to semantic inconsistencies.

8. Evaluation and Feedback Limitations

- Measuring the true success of recommendations is difficult without explicit user feedback.
- Metrics like watch time and clicks do not always indicate actual user satisfaction.

3.7 Cost and sustainability Impact

1. Cost Impact

- Infrastructure Costs: Hosting and maintaining the recommendation system requires cloud computing resources (e.g., AWS, Google Cloud), leading to storage and processing expenses.
- Computational Costs: ML models (TF-IDF, Word2Vec, BERT, Cosine Similarity, etc.) demand high computational power, increasing energy consumption and server costs.
- Data Processing and Storage: Large datasets require database management solutions, leading to costs associated with storage, backup, and retrieval.
- Development and Maintenance: Ongoing algorithm improvements, debugging, and feature updates require investment in development resources and skilled personnel.
- User Interaction Costs: Implementing real-time recommendations and handling high traffic loads may lead to higher bandwidth consumption and server scaling costs.

2. Sustainability Impact

- Energy Consumption: Running ML models for recommendations continuously can lead to high energy usage, impacting carbon footprint and sustainability.

- Scalability and Efficiency: Optimizing algorithms (e.g., using lightweight models, caching techniques, and batch processing) can reduce power consumption while maintaining efficiency.
- Cloud-Based Solutions: Using green cloud providers (AWS Sustainable Cloud, Google Cloud Carbon-Neutral Computing) helps reduce environmental impact.
- Data Optimization: Reducing unnecessary data storage and using efficient indexing techniques can minimize computational load and save energy.
- Promoting Lifelong Learning: The recommendation system contributes to sustainability by encouraging continuous education, awareness, and knowledge sharing, making high-quality content more accessible.

CHAPTER-4

IMPLEMENTATION

4.1 Environment Setup

Setting up the environment for building and deploying the TED Talks recommendation system involves installing the necessary tools, frameworks, and dependencies. Below are the key steps:

1. System Requirements

- Operating System: Compatible with Windows, macOS, or Linux.
- RAM: At least 8GB (16GB+ recommended for handling large datasets).
- Processor: Multi-core CPU (preferably with GPU support for NLP-based models).
- Storage: Minimum 20GB of free space for dataset storage and processing.

2. Required Software and Tools

- Python (3.8 or higher) – Primary programming language for development.
- Jupyter Notebook or VS Code – For writing and testing code.
- Git – For version control and collaboration.

3. Virtual Environment Setup

Creating a virtual environment is essential for managing dependencies and avoiding conflicts. This ensures that the project runs in an isolated environment with the necessary libraries installed.

4. Installation of Necessary Libraries

The project requires several libraries for data processing, machine learning, and natural language processing. These include:

- Data Handling: Pandas, NumPy
- Machine Learning: Scikit-learn
- Visualization: Matplotlib, Seaborn
- Natural Language Processing (NLP): NLTK, Transformers (for text embeddings)
- Deep Learning (if required): TensorFlow, Keras, PyTorch
- Deployment: Flask, Django, Streamlit (optional for web-based recommendations)

5. Dataset Preparation

- Download the TED Talks dataset and place it in a dedicated project folder.
- Load the dataset for preprocessing, including handling missing values, cleaning text data, and extracting relevant features.

6. Configuring NLP Tools for Text Processing

- Download necessary NLP resources such as stopword lists for text preprocessing.

- Implement vectorization techniques like TF-IDF, Word2Vec, or BERT embeddings to convert text into numerical format.

7. Choosing Execution Environment

- Local Setup: Use Jupyter Notebook or VS Code for development and testing.
- Cloud Setup: Utilize Google Colab, AWS, or Azure ML for scalable processing and model training.

4.2 Sample Code for Preprocessing and MLP Operations

4.2.1 Import Required Libraries

```
import pandas as pd

import numpy as np

import nltk

from nltk.corpus import stopwords

from nltk.tokenize import word_tokenize

from nltk.stem import WordNetLemmatizer

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.model_selection import train_test_split

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, Dropout

from tensorflow.keras.optimizers import Adam
```

4.2.2 Load and Preprocess Dataset

```
df = pd.read_csv("ted_talks.csv")

df = df[['title', 'description', 'tags', 'views']].dropna()

df['text_data'] = df['title'] + " " + df['description'] + " " + df['tags']

nltk.download('stopwords')

nltk.download('punkt')
nltk.download('wordnet')

lemmatizer = WordNetLemmatizer()

stop_words = set(stopwords.words('english'))

def preprocess_text(text):

 words = word_tokenize(text.lower())

words = [lemmatizer.lemmatize(word) for word in words if word.isalnum() and word not in stop_words]

    return " ".join(words)

df['clean_text'] = df['text_data'].apply(preprocess_text)

 vectorizer = TfidfVectorizer(max_features=5000)

X = vectorizer.fit_transform(df['clean_text']).toarray()

y = np.log1p(df['views'])
```

4.2.3 Split Data for Training and Testing

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

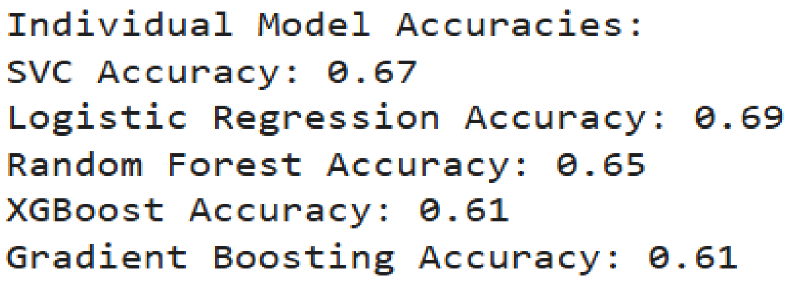
4.2.4 Define the MLP Model

```
model = Sequential([

   Dense(512, activation='relu', input_shape=(X_train.shape[1],)),

   Dropout(0.3),

   Dense(256, activation='relu'),

   Dropout(0.3),

   Dense(128, activation='relu'),

   Dense(1, activation='linear')  # Output is a numerical score (views)

])

model.compile(optimizer=Adam(learning_rate=0.001), loss='mse', metrics=['mae'])

model.fit(X_train, y_train, epochs=10, batch_size=32, validation_data=(X_test, y_test))
```

CHAPTER-5
EXPERIMENTATION & RESULT ANALYSIS

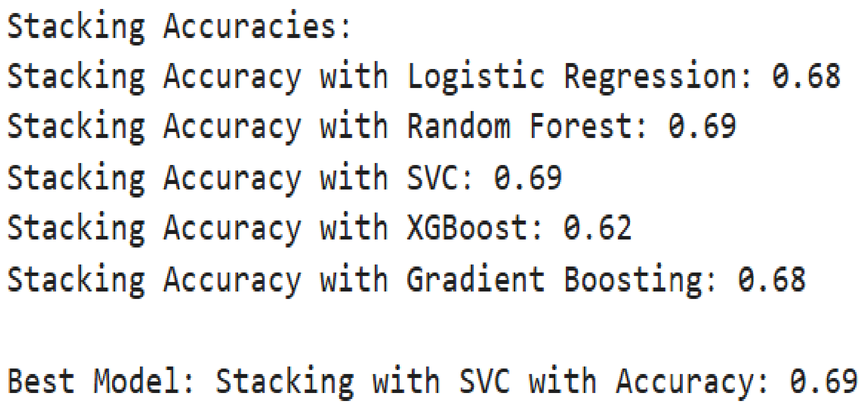
- The experimentation and result analysis phase evaluates the effectiveness of the content-based filtering approach used in the TED Talks recommendation system. This includes testing the model, analyzing its performance, and interpreting the results using appropriate evaluation metrics.

OUTPUTS:



```
Individual Model Accuracies:
SVC Accuracy: 0.67
Logistic Regression Accuracy: 0.69
Random Forest Accuracy: 0.65
XGBoost Accuracy: 0.61
Gradient Boosting Accuracy: 0.61
```

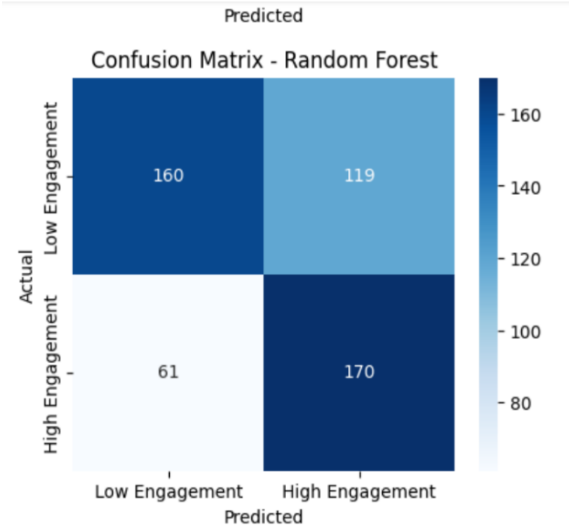
FIGURE 5.1-ACCURACIES



```
Stacking Accuracies:
Stacking Accuracy with Logistic Regression: 0.68
Stacking Accuracy with Random Forest: 0.69
Stacking Accuracy with SVC: 0.69
Stacking Accuracy with XGBoost: 0.62
Stacking Accuracy with Gradient Boosting: 0.68

Best Model: Stacking with SVC with Accuracy: 0.69
```

FIGURE 5.2-STATIC ACCURACIES

RANDOM FOREST:



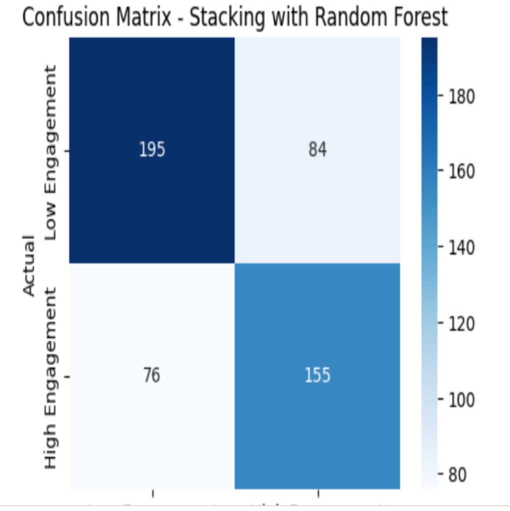
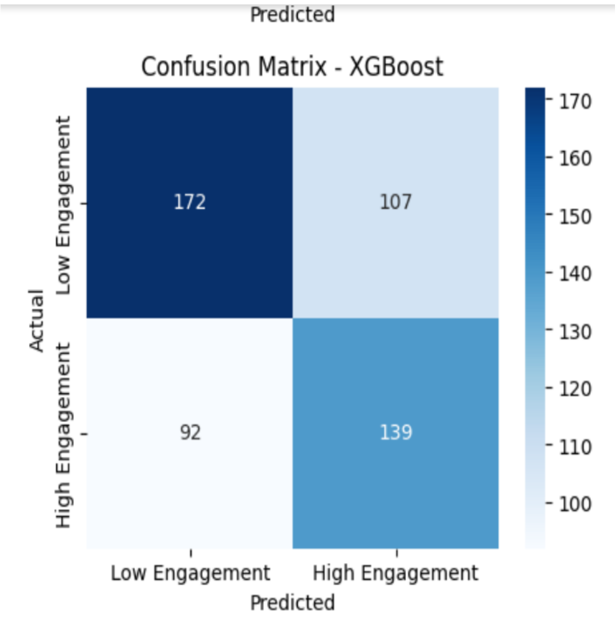


FIGURE 5.3-RANDOM FOREST

FIGURE 5.4-STACKING RANDOMFOREST

XGBOOST:



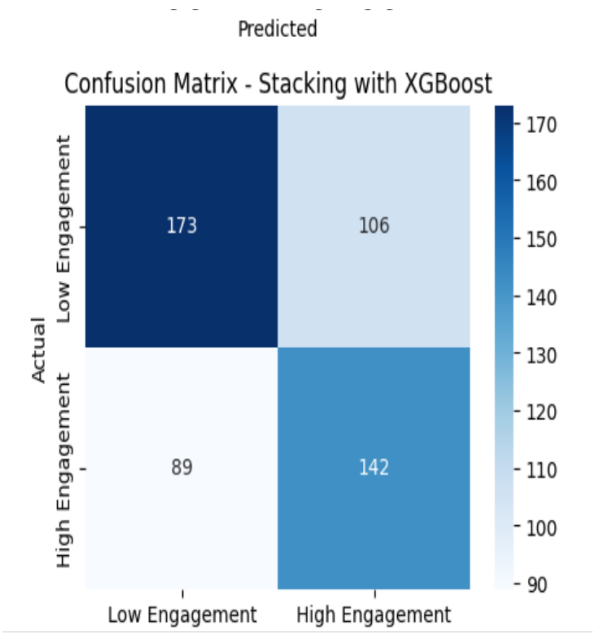
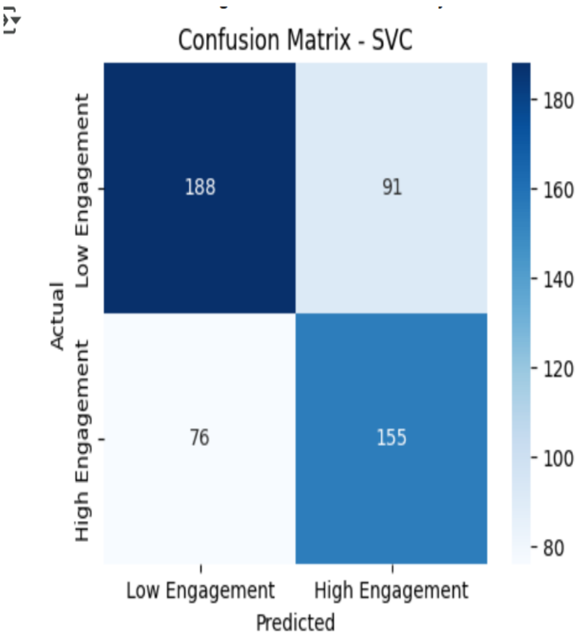


FIGURE 5.5-XGBOOST

FIGURE 5.6-STACKING WITH XGBOOST

SUPPORT VECTOR CLASSIFICATION(SVC):



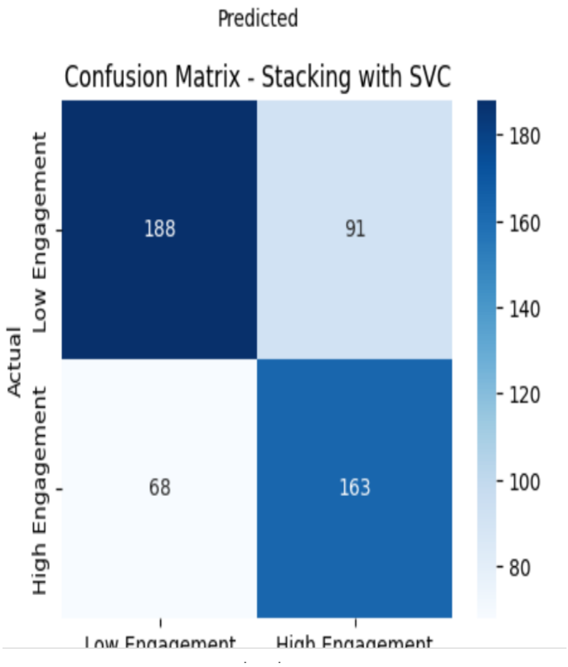
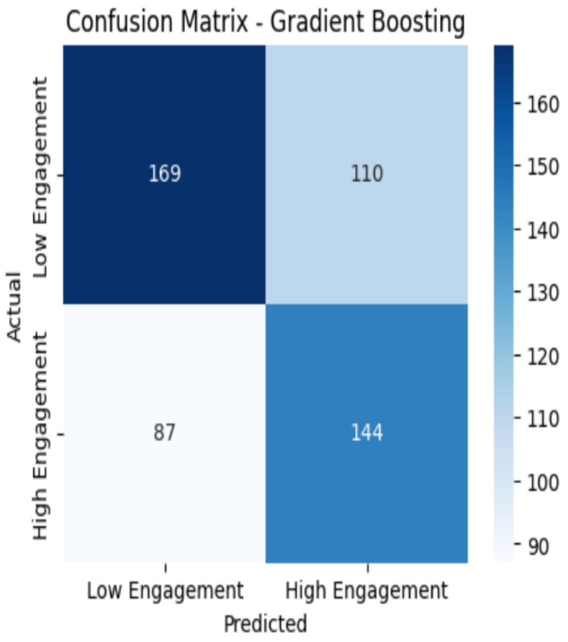


FIGURE 5.7-SVC

FIGURE 5.8-STACKING WITH SVC

GRADIENT BOOSTING:



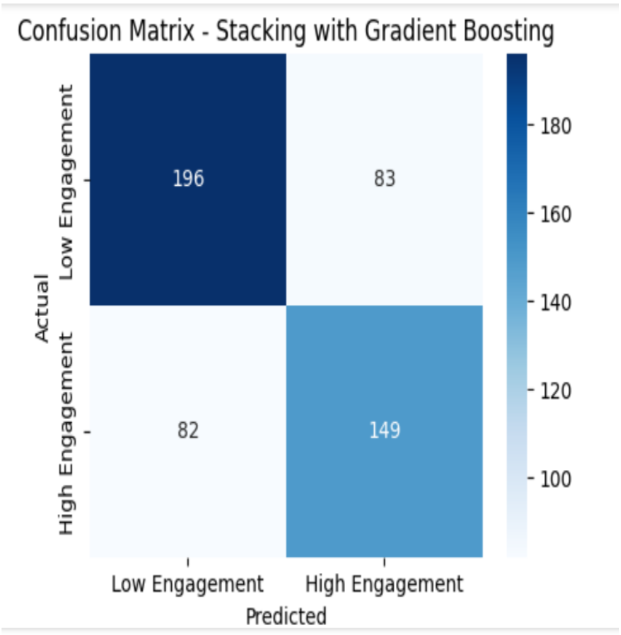
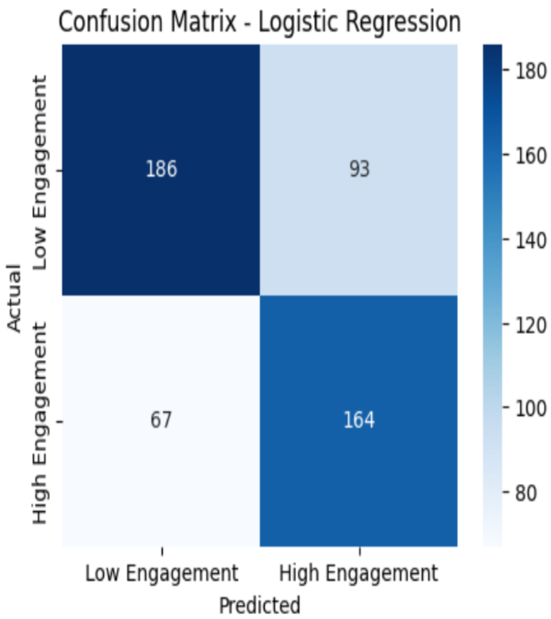


FIGURE 5.9-GRADIENT BOOSTING

FIGURE 5.10-STACKING WITH GRADIENT BOOSTING

LOGISTIC REGRESSION:



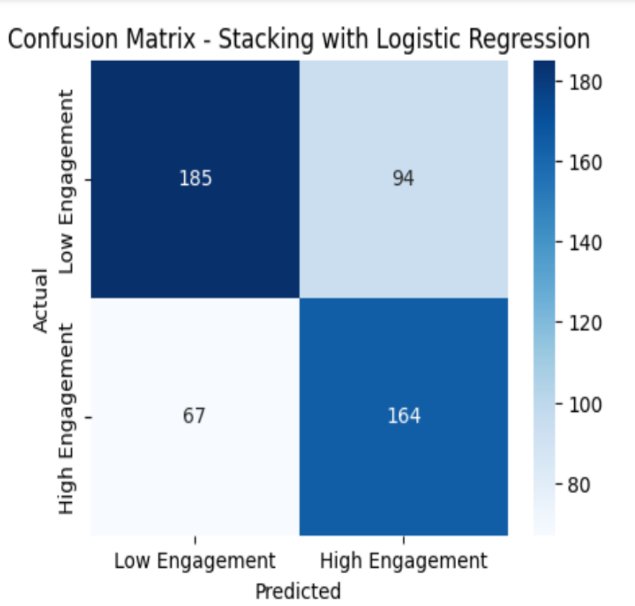


FIGURE 5.11-LOGISTIC REGRESSION

FIGURE 5.12-STACKING WITH LOGISTIC

CHAPTER-6

CONCLUSION

CONCLUSION:

The TED Talks Recommendation System using Content-Based Filtering and MLP effectively personalizes suggestions by analyzing textual features such as descriptions, titles, and tags. By leveraging TF-IDF for feature extraction and an MLP model for predictive ranking, the system provides relevant recommendations based on user interests. The implementation of NLP techniques such as tokenization, stopword removal, and lemmatization enhances data quality, improving recommendation accuracy. Additionally, the MLP model learns patterns from past TED Talk trends to predict their popularity. While content-based filtering ensures personalized recommendations, it may lead to over-specialization, limiting diversity in suggestions. Future enhancements, such as hybrid filtering (collaborative + content-based) and real-time user feedback integration, could further improve the system’s adaptability. Overall, this project demonstrates a practical and scalable machine-learning approach to TED Talk recommendations, enhancing user engagement and content discoverability.

CHAPTER-7

REFERENCES

[1]P. Bailke, A. Asalkar, P. Bansode, N. Baviskar, A. Belote and H. Nadar, &quot;Ted-Talks Recommendation System using ML Algorithms,&quot; 2024 First International Conference on Software, Systems and Information Technology (SSITCON), Tumkur, India, 2024, pp. 1-6, doi: 10.1109/SSITCON62437.2024.10795915.

[2] N. Pappas and A. Popescu-Belis, &quot;Combining content with user preferences for TED lecture recommendation,&quot; 2013 11th International Workshop on Content-Based Multimedia Indexing (CBMI), Veszprem, Hungary, 2013, pp. 47-52, doi: 10.1109/CBMI.2013.6576551.

[3] H. -Y. Chen, Y. -S. Lin and C. -C. Lee, &quot;Through the Eyes of Viewers: A Comment-Enhanced Media Content Representation for TED Talks Impression Recognition,&quot; 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Lanzhou, China, 2019, pp. 414-418, doi: 10.1109/APSIPAASC47483.2019.9023066.

[4] H. Shrimali, R. Saxena and Kavita, &quot;Content based Video Recommendation System,&quot; 2023 3rd International Conference on Intelligent Communication and Computational Techniques (ICCT), Jaipur, India, 2023, pp. 1-3, doi: 10.1109/ICCT56969.2023.10075906.

[5] H. Zarzour, Z. A. Al-Sharif and Y. Jararweh, &quot;RecDNNing: a recommender system using deep neural network with user and item embeddings,&quot; 2019 10th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 2019, pp. 99-103, doi:10.1109/IACS.2019.8809156.

[6] A. Wu and H. Qu, &quot;Multimodal Analysis of Video Collections: Visual Exploration of Presentation Techniques in TED Talks,&quot; in IEEE Transactions on Visualization and Computer Graphics, vol. 26, no. 7, pp. 2429-2442, 1 July 2020, doi: 10.1109/TVCG.2018.2889081.

[7] H. Zarzour et al., &quot;Using K-means Clustering Ensemble to Improve the Performance in Recommender Systems,&quot; 2022 International Conference on Intelligent Data Science

Technologies and Applications (IDSTA), San Antonio, TX, USA, 2022, pp. 176-180, doi: 10.1109/IDSTA55301.2022.9923070.

[8] H. E. Aouifi, M. E. Hajji, Y. Es-Saady and H. Douzi, &quot;Video-Based Learning Recommender Systems: A Systematic Literature Review,&quot; in IEEE Transactions on Learning Technologies, vol. 17, pp. 485-497, 2024, doi: 10.1109/TLT.2023.3313391.

[9] Kavinkumar V., R. R. Reddy, R. Balasubramanian, Sridhar M., Sridharan K. and D. Venkataraman, &quot;A hybrid approach for recommendation system with added feedback component,&quot; 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Kochi, India, 2015, pp. 745-752, doi:10.1109/ICACCI.2015.7275700.

[10] S. M. Al-Ghuribi and S. A. Mohd Noah, &quot;Multi-Criteria Review-Based Recommender System–The State of the Art,&quot; in IEEE Access, vol. 7, pp. 169446-169468, 2019, doi: 10.1109/ACCESS.2019.2954861.

[11] M. -R. Petrusel and S. -G. Limboi, &quot;A Restaurants Recommendation System: Improving Rating Predictions Using Sentiment Analysis,&quot; 2019 21st International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), Timisoara, Romania, 2019, pp. 190-197, doi: 10.1109/SYNASC49474.2019.00034.

[12] A. Oshnoudi, B. S. Neysiani, Z. Aminoroaya and N. Nematbakhsh, &quot;Improving Recommender Systems Performances Using User Dimension Expansion by Movies’ Genres and Voting-Based Ensemble Machine Learning Technique,&quot; 2021 7th International Conference on Web Research (ICWR), Tehran, Iran, 2021, pp. 175-181, doi: 10.1109/ICWR51868.2021.9443146.

[13] P. Nagarnaik and A. Thomas, &quot;Survey on recommendation system methods,&quot; 2015 2nd International Conference on Electronics and Communication Systems (ICECS), Coimbatore,India, 2015, pp. 1603-1608, doi: 10.1109/ECS.2015.7124857.

[14] M. Robillard, R. Walker and T. Zimmermann, &quot;Recommendation Systems for SoftwareEngineering,&quot; in IEEE Software, vol. 27, no. 4, pp. 80-86, July-Aug. 2010, doi: 10.1109/MS.2009.161.

[15] H. Tan, J. Guo and Y. Li, &quot;E-learning Recommendation System,&quot; 2008 InternationalConference on Computer Science and Software Engineering, Wuhan, China, 2008, pp. 430-433, doi: 10.1109/CSSE.2008.305.