Mini Project Report:

Course Recommendation System

## 1. Project Title

Content-Based Course Recommendation System

## 2. Problem Statement

Online learners face difficulty discovering courses aligned with their interests or skill levels due to the overwhelming number of available options. This project aims to build a content-based recommendation system that suggests relevant courses to users based on course metadata and textual information.

## 3. Objective

• Implement a content-based recommendation system using NLP.  
• Compare classical TF-IDF and Transformer-based sentence embeddings.  
• Provide an interactive Streamlit web app to visualize recommendations.  
• Demonstrate understanding of cosine similarity, vector representations, and feature-based retrieval.

## 4. Dataset Description

Dataset Used: coursera\_data.csv (Coursera course metadata)  
• Total Entries: 891  
• Attributes: course\_title, course\_organization, course\_Certificate\_type, course\_rating, course\_difficulty, course\_students\_enrolled, clean\_text.

## 5. Methodology

**Step 1: Data Preprocessing**  
• Lowercasing, removing punctuation, and cleaning text.  
• Stopword removal and tokenization.  
• Combining multiple columns into one textual feature ('clean\_text').  
  
**Step 2: Text Vectorization (Feature Extraction)**  
1. TF-IDF Vectorization: Captures term frequency weighted by inverse document frequency.  
2. Transformer Sentence Embeddings: Uses SentenceTransformer (paraphrase-MiniLM-L3-v2) for contextual representation.  
  
**Step 3: Similarity Computation**  
• Cosine Similarity: Measures similarity between two vectors.  
• Formula: similarity(A, B) = (A · B) / (||A|| ||B||)

**Step 4: Recommendation Generation**  
• Course-to-Course: Finds top-K most similar courses.  
• Query-to-Course: Encodes user query and finds top-K similar courses.  
  
**Step 5: Streamlit App**  
• Built using Streamlit with sidebar controls for mode selection, encoder choice, and K value.

## 6. Concepts Applied

• Content-Based Filtering: Recommends items based on content similarity.  
• Cosine Similarity: Measures angular distance between feature vectors.  
• TF-IDF Vectorization: Traditional word-based representation for text.  
• Transformer Embeddings: Deep contextual representation for semantic understanding.  
• Hybrid Approach: Combines traditional and deep NLP for better results.

## 6.1. Recommendation System Paradigms

Recommendation systems can be broadly classified into four types:  
1. Content-Based Filtering – Recommends items based on similarity in item attributes.  
2. Collaborative Filtering – Recommends items based on user behavior or preferences.  
3. Hybrid Systems – Combine content and collaborative filtering.  
4. Context-Aware Systems – Incorporate external factors such as time, location, or user context.  
This project specifically implements the Content-Based Filtering approach and partially integrates hybrid elements by combining TF-IDF and Transformer embeddings for richer text representation.

## 6.2 Content-Based Filtering (Implemented Technique)

Content-Based Filtering (CBF) is an item-centric approach. It recommends items similar to those the user has already interacted with, based on item features. In this project, each course is represented as a feature vector derived from its textual description (course title, organization, and difficulty).

The similarity between items is computed using \*\*cosine similarity\*\*, which measures how close two vectors are in multi-dimensional space. If the cosine of the angle between two vectors is close to 1, the items are similar.

Formula:  
similarity(A, B) = (A · B) / (||A|| \* ||B||)

## 6.3 NLP Concepts in Feature Extraction

Two feature extraction techniques were implemented:  
  
1. TF-IDF (Term Frequency–Inverse Document Frequency):  
 - Measures the importance of words in a document relative to the corpus.  
 - Gives higher weight to unique words and lower weight to common words.  
 - Efficient, interpretable, but limited to surface-level lexical similarity.  
  
2. Transformer-Based Sentence Embeddings:  
 - Uses a pre-trained SentenceTransformer model ('paraphrase-MiniLM-L3-v2') to generate contextual embeddings.  
 - Captures deeper semantic relationships beyond keyword matching.  
 - Represents each course as a 384-dimensional vector embedding.  
  
The combination of these two representations introduces a hybrid content-based system, improving recommendation quality.

## 6.4 System Workflow

The recommendation process involves the following steps:  
1. Preprocess course text by cleaning and normalizing.  
2. Convert text to numerical representations using TF-IDF and Transformer embeddings.  
3. Compute cosine similarity between all course vectors.  
4. Rank and display the top-K most similar courses.  
5. Provide two user interaction modes:  
 - By Example Course (select from dataset)  
 - By Text Query (enter keywords or phrases)

## 6.5 Theoretical Concepts Applied

**• Vector Space Model:** Each course is represented as a point in high-dimensional space.  
**• Cosine Similarity:** Measures angular similarity between feature vectors.  
• **Feature Engineering:** Combines multiple text fields for richer semantic representation.  
**• Dimensionality Reduction (implicit):** Transformer embeddings compress semantic information into fixed-size vectors.  
**• Ranking:** Items are ranked by decreasing similarity score.  
**• Hybrid NLP Representation:** Combines statistical and contextual text embeddings.

## 7. Results and Observations

Example Query: 'Python'  
  
Transformer-based Recommendations:  
1. Using Python to Access Web Data – University of Michigan (Score: 0.204)  
2. Introducción a la programación en Python I – Universidad Católica de Chile (Score: 0.191)  
  
TF-IDF-based Recommendations:  
1. Python Basics – University of Michigan (Score: 0.499)  
2. Data Analysis with Python – IBM (Score: 0.474)  
  
Observation:  
• TF-IDF performs better for keyword-based matches.  
• Transformer embeddings capture semantic similarity even across languages.

## 8. Evaluation

Metric Comparison:  
• Computation Speed: TF-IDF is faster; Transformer slower (model load once).  
• Interpretability: TF-IDF easier to interpret.  
• Semantic Relevance: Transformer performs better for concept-based matching.  
• Memory Usage: Transformer requires more memory due to embedding size.

## 9. Learning Outcomes

This project covers the core principle of **Content-Based Filtering**, demonstrating how item similarity can be derived from textual attributes using NLP. It integrates key theoretical and practical concepts from recommendation system design, such as feature representation, similarity metrics, and ranking mechanisms.  
  
Through experimentation, the following insights were observed:  
• TF-IDF offers speed and interpretability but limited semantic understanding.  
• Transformer embeddings provide deeper contextual similarity but require more computation.  
• The system achieves a balance between accuracy and efficiency, suitable for real-world deployment.  
  
Thus, this project provides a hands-on understanding of how modern recommendation systems can leverage linguistic and semantic features to deliver personalized, content-driven suggestions.

## 10. Conclusion

This mini project demonstrates the design and implementation of a content-based recommendation system leveraging both classical (TF-IDF) and deep learning (Transformer) NLP methods. It effectively recommends relevant courses based on content similarity and showcases practical understanding of recommendation system concepts. The system balances performance and interpretability and can serve as a foundation for future hybrid recommenders.