CSC 577 Project Proposal

Team Name: AI Recommenders

Team Members:

- Sivaramakrishna Yarra
- Thusharika Sree Peethala
- Avinash Betha

Project Title:
Hybrid Recommender System for Personalized Product Recommendations on Amazon

Type of Project:
Algorithm-Based

Problem Statement:
Recommender systems play a crucial role in e-commerce platforms by helping users discover products that align with their interests. Traditional recommendation techniques like Collaborative Filtering (CF) and Content-Based Filtering (CBF) often suffer from challenges such as the cold start problem, sparse user-product interactions, and lack of diversity in recommendations.

This project aims to build a Hybrid Recommender System that combines Collaborative Filtering, Content-Based Filtering, and Deep Learning models to enhance recommendation accuracy and user satisfaction. Using the Amazon Product Reviews dataset, we will develop a system that generates personalized recommendations by leveraging user behavior, product metadata, and deep learning-based embeddings.

Dataset Description:
We will use the Amazon Product Reviews dataset, which contains:

- **User Ratings & Reviews** (Text feedback, star ratings, helpfulness votes)
- **Product Metadata** (Descriptions, brand, category, price, and image features)
- **User-Item Interactions** ("Also viewed" and "Also bought" relationships)
- **Timestamped Transactions** (Temporal purchase patterns)

This dataset provides a rich feature set for personalized recommendations.

Data Source:
Amazon Product Reviews dataset from Kaggle.

Recommendation Tasks:

- **Personalized Product Recommendation:** Predict user preferences based on past behavior.
- **Cold-Start Problem Mitigation:** Recommend items using content-based approaches.
- **Hybrid Model Development:** Improve accuracy and diversity by combining multiple techniques.

Algorithms to be Implemented:
We will integrate multiple techniques to create a robust hybrid recommender system:

1. **Collaborative Filtering (CF):**
   - User-Based CF (Finding similar users to recommend products)
   - Item-Based CF (Finding similar items based on user interactions)
2. **Content-Based Filtering (CBF):**
   - TF-IDF & Word2Vec (Extracting features from product descriptions)
   - Product Embeddings (Learning feature representations from metadata)
3. **Matrix Factorization Techniques:**
   - Singular Value Decomposition (SVD)
   - Alternating Least Squares (ALS)
4. **Deep Learning Models:**
   - Neural Collaborative Filtering (NCF)
   - Autoencoders for User & Item Embeddings
   - Transformer-based Product Recommendation Model

Methodology:
The project will follow these structured steps:

Step 1: Data Preprocessing

- Load and clean dataset (handling missing values, duplicates)
- Tokenize and preprocess textual data (TF-IDF, Word2Vec embeddings)
- Normalize numerical features (ratings, price, popularity)

Step 2: Exploratory Data Analysis (EDA)

- Visualize rating distributions and user-item interactions
- Analyze trends in user behavior and product categories

Step 3: Implement Recommendation Models

- Collaborative Filtering (User-Based & Item-Based)
- Content-Based Filtering (TF-IDF, embeddings, similarity scores)
- Matrix Factorization (SVD, ALS)
- Neural Networks (Autoencoders, Transformers, NCF)

Step 4: Hybrid Model Development

- Combine CF, CBF, and deep learning models using weighted averaging or an ensemble learning approach
- Use meta-learning techniques to optimize model weights

Step 5: Model Evaluation & Tuning

- Train-test split for validation
- Evaluate performance using the following metrics:
    - **Precision, Recall, and F1-Score** (Accuracy of recommendations)
    - **Mean Squared Error (MSE) & RMSE** (Rating prediction performance)
    - **Hit Rate & Mean Average Precision (MAP)** (Ranking effectiveness)
    - **Diversity & Novelty Metrics** (Avoiding repetitive recommendations)
- Tune hyperparameters using GridSearchCV or Bayesian Optimization

Step 6: Deployment

- Develop a Flask/Streamlit web interface for users to interact with recommendations
- Deploy on AWS/GCP with an API for real-time recommendations
- Implement a CI/CD pipeline for model updates and continuous improvements

Evaluation Metrics:
We will use the following evaluation metrics to assess the performance of our recommendation models:

- **Precision, Recall, and F1-Score** (Accuracy of recommendations)
- **Mean Squared Error (MSE) & Root Mean Squared Error (RMSE)** (Rating prediction performance)
- **Hit Rate & Mean Average Precision (MAP)** (Ranking effectiveness)
- **Diversity & Novelty Metrics** (Ensuring non-repetitive, diverse recommendations)