AMBRIX – ML Intern – Short Assignment:

Name: K.Poornima | **Mail ID:** poorni1907@gmail.com |

LinkedIn Profile: https://www.linkedin.com/in/poornimakarunanidhi1906 |

Github Profile: https://github.com/PoorniKR

Interest-Based Content Recommendation System

1. Introduction

This project implements a hybrid recommendation system to suggest posts to users based on their interests, content preferences, and past engagement behaviour. The system combines **content-based filtering** and **collaborative filtering** to generate personalized recommendations.

2. Data Overview

Three datasets are used:

Dataset	Description
Users	User demographics, interests, and past engagement score
Posts	Post content types and tags
Engagements	User interactions with posts

- **Users**: Contains age, gender, top_3_interests, past_engagement_score.
- **Posts**: Contains post_id, content_type, tags.
- **Engagements**: Contains user_id, post_id, engagement.

Data is split into **training (80%)** and **test (20%)** sets based on user engagements.

3. Feature Engineering

3.1 User Features

- **Interests:** Top 3 interests are one-hot encoded.
- **Demographics:** Age normalized, gender encoded, engagement score normalized.
- **Final User Feature Vector:** Concatenation of demographics + interest features.

3.2 Post Features

- **Content Type:** Encoded using LabelEncoder.
- **Tags:** Converted to TF-IDF vector of top 20 tags.
- **Final Post Feature Vector:** Concatenation of content type + tag features.

4. Recommendation Approaches

4.1 Content-Based Filtering

- Computes **overlap between user interests and post tags**.
- Boosts posts based on preferred content type (e.g., video/image).
- Returns top-K posts the user is most likely to engage with.

4.2 Collaborative Filtering

- Uses **Truncated SVD** on the user-post engagement matrix.
- Predicts missing interactions for each user by reconstructing engagement scores.
- Excludes posts the user has already engaged with in training.

4.3 Hybrid Recommendation

- Combines content-based and collaborative scores:
    - Weight: **0.6 for content-based**, **0.4 for collaborative**.
- Normalizes scores and selects top-K posts for each user.

5. Evaluation Metrics

- **Precision@K**: Fraction of recommended posts that the user engaged with in the test set.
- **User Coverage@K**: Fraction of users with at least one correct recommendation.

Sample Evaluation (K=3):

Method	Precision@3	Coverage@3
Content-Based	0.420	0.65
Collaborative	0.370	0.60
Hybrid	0.460	0.68

Hybrid method performs best by leveraging both user interest and engagement behavior.

6. Deliverables

- **Recommendations CSV:** recommendations.csv – Top 3 posts per user.
- **Detailed Recommendations CSV:** detailed_recommendations.csv – Includes user interests, post content type, tags, recommendation score, and match reason.
- **Python Notebook:** Implements data preprocessing, feature engineering, content-based, collaborative, and hybrid recommendation pipelines.

7. Possible Extensions

1. **Deep Learning Models:** Use embeddings for users and posts via PyTorch/TensorFlow to capture complex patterns.
2. **Temporal Dynamics:** Factor in recency of posts and user activity for more relevant recommendations.
3. **Context-Aware Recommendations:** Incorporate location, time-of-day, device type, or session context.
4. **Explainability Enhancements:** Generate richer explanations for recommendations (e.g., similar users' behaviour).
5. **Dynamic Weighting:** Learn the optimal hybrid weight for content-based vs collaborative filtering using validation data.

8. Conclusion

This hybrid recommendation system effectively integrates content-based and collaborative filtering, achieving higher precision and coverage than individual approaches. It is scalable, interpretable, and can be extended with deep learning or context-aware features.