Boomm Post Recommendation System – Report

Objective

Build a hybrid post recommendation system for Boomm, a social platform where finance and community meet, to suggest relevant posts to users based on their interests, content similarity, engagement metrics and collaborative behavior.

Data

- **Users (users.csv)**: user_id, interested_in (topics of interest)
- **Posts (posts.csv)**: post_id, content, is_anonymous, created_at, likes, shares, reports, like_user_ids, topics

Cleaning & Preprocessing:

- Filled missing interests and post engagement metrics
- Lowercased and stripped text fields
- Converted created_at to datetime

Methodology

1. Feature Engineering

1. **Popularity Score** – weighted combination of likes, shares, and reports:
   *popularity_raw= 0.7 likes + 0.2 * shares - 0.1 *reports*
   Normalized between 0–1.
2. **Recency Score** – normalized post age:
   recency = created_at - earliest_post / (latest_post - earliest_post)
3. **Semantic Score** – embedding-based similarity between user interests and post content using BERT:
   - Generated embeddings for user interests and post content
   - Computed cosine similarity for each user-post pair
4. **Collaborative Score** – recommendation based on similar users' interactions:
   - Built interaction matrix from like_user_ids
   - Computed user-user similarity and weighted post scores

---

2. Hybrid Scoring & Recommendations

- **Final score for each user-post pair:**
  *final_score = 0.5 * semantic + 0.25 *popularity + 0.15 *recency + 0.1 *collaborative*
- For users with no interests, higher weight given to popularity and recency.
- Selected **top 10 posts per user** based on final_score.

Results

- Produced a **CSV** with columns: user_id, recommended_post_ids (comma-separated top 10 posts)

Insights

- Posts with **high semantic similarity and popularity** dominate top recommendations.
- Users with no declared interests still receive meaningful recommendations via popularity and collaborative signals.
- Combining **content-based and collaborative filtering** provides a balanced, human-like recommendation experience.

Deliverables

- boomm_recommendations.csv – top 10 recommended posts per user
- Code implementation in Jupyter Notebook