Approach

To build a personalized recommendation system, I used the three datasets — **Users.csv**, **Posts.csv**, and **Engagements.csv** — together in the following way:

1. Data Understanding

- **Users.csv** provides each user's top 3 interests.
- **Posts.csv** contains posts with tags and content type (video, image, text).
- **Engagements.csv** shows past interactions between users and posts (what they have already seen or engaged with).

Together, this means I can combine *who the user is (interests)*, *what posts exist (tags, type)*, and *what they liked in the past (engagements)* to decide future recommendations.

---

2. Data Preprocessing

- Cleaned and normalized tags and interests (lowercased, split by commas).
- Converted IDs to consistent string format for joining across datasets.
- Created lists:
    - interest_list for each user
    - tag_list for each post

This ensures user interests and post tags can be compared easily.

---

3. Candidate Post Selection

For each user, I removed posts they had already engaged with (to avoid recommending duplicates). Then, the candidate posts were selected from all remaining posts.

---

4. Scoring Function (Baseline)

Each candidate post was given a **score** based on:

1. **Interest overlap:** Number of matching tags between user's interests and post tags.
2. **Content type bonus:** Extra weight for more engaging formats (e.g., video > image > text).

Formula (conceptually):

score = interest_matches + 0.2 * content_type_weight

---

5. Ranking & Top-3 Recommendation

- For every user, all candidate posts were scored.

- Posts were sorted in descending order of score.
- The **top 3 posts** were selected as recommendations.

The results were stored in a CSV file AMBRIX_recommendations.csv containing:
user_id, rank, post_id, score.

---

6. Evaluation

To check the quality of recommendations, I used a **holdout method**:

- For users with ≥4 engagements, the last engagement was used as the *test post* (what we try to predict).
- Recommendations were generated using earlier engagements.
- **Precision@3** was computed: proportion of users whose true last post appeared in their top-3 recommendations.

---

7. Extensions (Possible Improvements)

- **Learning-based model:** Train a logistic regression or neural network using features (interest overlap, content type, post popularity, user activity) to predict engagement probability.
- **Collaborative Filtering:** Use item–item similarity (cosine similarity of post tags/embeddings) to recommend posts similar to what a user already engaged with.
- **Hybrid approach:** Combine content-based and collaborative filtering for better accuracy.
- **Diversity control:** Ensure recommended posts are not too similar (avoid showing 3 posts from the same tag or creator).