<u>ASSIGNMENT 2</u>

1) High-Level Pipeline Overview

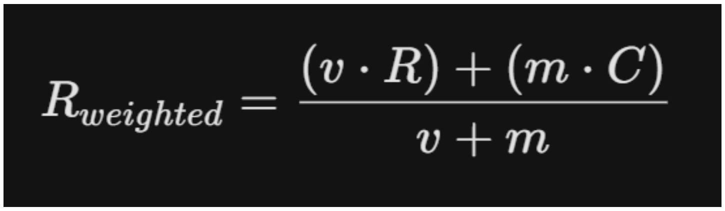
The pipeline follows a strict 5-fold cross-validation strategy to ensure model robustness and prevent data leakage. The process for each fold is as follows:

1. Data Ingestion Loading specific train/test splits (u1.base to u5.test).
2. Feature Engineering:
   - Calculating Bayesian smoothed means for Users and Items.
   - Generating latent factors via Truncated SVD.
   - Computing interaction features (Dot Product).
3. Model Training: Training three distinct classifiers (XGBoost, LightGBM, CatBoost) using optimized hyperparameters.
4. Ensembling: Combining predictions via Soft Voting (averaging class probabilities).
5. Evaluation: Measuring performance using Accuracy.

2) Methodology & Mathematical Formulas

A. Bayesian Smoothing (Shrinkage)

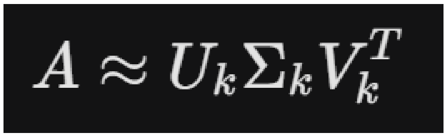
Standard averages for users or items with very few ratings can be misleading (e.g., a movie with one 5-star rating isn't necessarily better than one with 500 4.5-star ratings). Bayesian smoothing pulls these "low-confidence" means toward the global average.



$$R_{weighted} = \frac{(v \cdot R) + (m \cdot C)}{v + m}$$

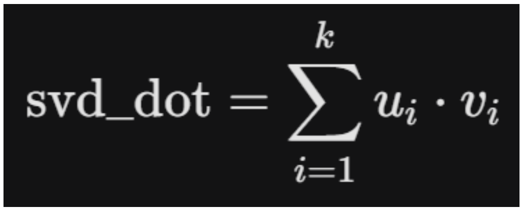
B. Truncated SVD (Latent Factor Analysis)

The code creates a User-Item interaction matrix with rows for users, columns for items, and ratings as values. Since this matrix is sparse, a Truncated Singular Value Decomposition (SVD) is used to compress it to 8 dimensions.



$$A \approx U_k \Sigma_k V_k^T$$

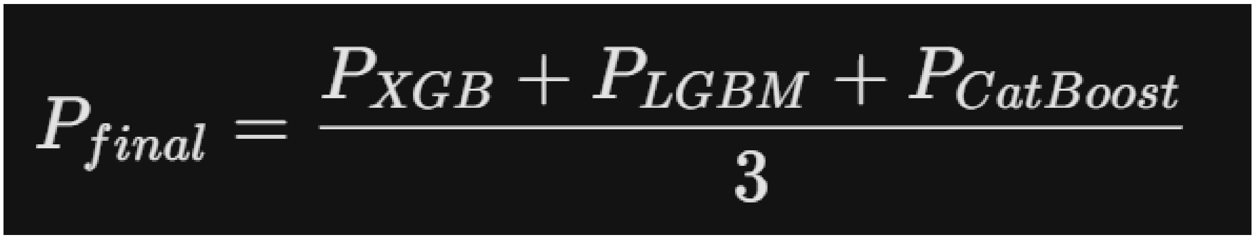
Interaction Feature (Dot Product): The code calculates the similarity between a user and an item by taking the dot product of their latent vectors:



$$\text{svd\_dot} = \sum_{i=1}^{k} u_i \cdot v_i$$

C. Triple Ensemble (Soft Voting)

Instead of relying on a single model, the pipeline uses a Triple Ensemble. Each model provides a probability distribution across the 5 possible rating classes (1–5).



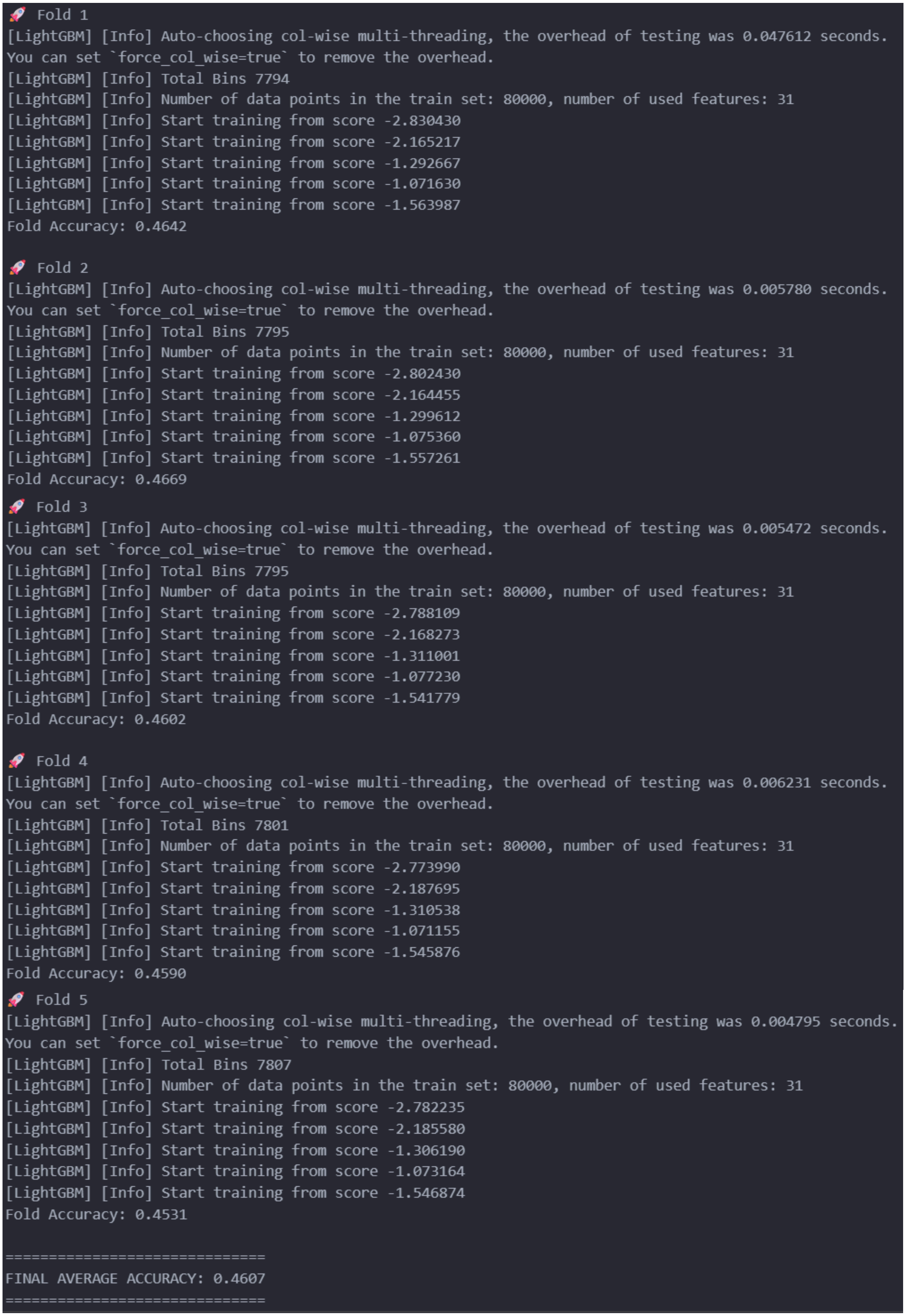
$$P_{final} = \frac{P_{XGB} + P_{LGBM} + P_{CatBoost}}{3}$$

The final prediction is the class with the highest average probability.

3) Performance Results

The following table shows the ensemble model's accuracy across the 5 folds of the MovieLens 100k dataset.

FOLD	ACCURACY
1	**0.4642**
2	**0.4669**
3	**0.4602**
4	**0.4590**
5	**0.4531**
AVERAGE	**0.4607**



```
🚀 Fold 1
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of testing was 0.047612 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 7794
[LightGBM] [Info] Number of data points in the train set: 80000, number of used features: 31
[LightGBM] [Info] Start training from score -2.830430
[LightGBM] [Info] Start training from score -2.165217
[LightGBM] [Info] Start training from score -1.292667
[LightGBM] [Info] Start training from score -1.071630
[LightGBM] [Info] Start training from score -1.563987
Fold Accuracy: 0.4642

🚀 Fold 2
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of testing was 0.005780 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 7795
[LightGBM] [Info] Number of data points in the train set: 80000, number of used features: 31
[LightGBM] [Info] Start training from score -2.802430
[LightGBM] [Info] Start training from score -2.164455
[LightGBM] [Info] Start training from score -1.299612
[LightGBM] [Info] Start training from score -1.075360
[LightGBM] [Info] Start training from score -1.557261
Fold Accuracy: 0.4669
🚀 Fold 3
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of testing was 0.005472 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 7795
[LightGBM] [Info] Number of data points in the train set: 80000, number of used features: 31
[LightGBM] [Info] Start training from score -2.788109
[LightGBM] [Info] Start training from score -2.168273
[LightGBM] [Info] Start training from score -1.311001
[LightGBM] [Info] Start training from score -1.077230
[LightGBM] [Info] Start training from score -1.541779
Fold Accuracy: 0.4602

🚀 Fold 4
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of testing was 0.006231 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 7801
[LightGBM] [Info] Number of data points in the train set: 80000, number of used features: 31
[LightGBM] [Info] Start training from score -2.773990
[LightGBM] [Info] Start training from score -2.187695
[LightGBM] [Info] Start training from score -1.310538
[LightGBM] [Info] Start training from score -1.071155
[LightGBM] [Info] Start training from score -1.545876
Fold Accuracy: 0.4590
🚀 Fold 5
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of testing was 0.004795 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 7807
[LightGBM] [Info] Number of data points in the train set: 80000, number of used features: 31
[LightGBM] [Info] Start training from score -2.782235
[LightGBM] [Info] Start training from score -2.185580
[LightGBM] [Info] Start training from score -1.306190
[LightGBM] [Info] Start training from score -1.073164
[LightGBM] [Info] Start training from score -1.546874
Fold Accuracy: 0.4531

==============================
FINAL AVERAGE ACCURACY: 0.4607
==============================
```