Loading Training Data:

- The training data, containing user ratings for items (tracks, albums, or artists), is read from `trainIdx2\_matrix.txt`.

- A dictionary named `train\_data` is created, where each key is a `userID` and its value is another dictionary mapping `itemID` to `rating`.

- This structure allows quick access to any user's rating for a specific item.

Processing Test Data and Writing Predictions:

- The test data file `testTrack\_hierarchy.txt` is read line by line.

- Each line contains a `userID`, `trackID`, `albumID`, and `artistID`, which are extracted.

User-Specific Processing:

- For each new user encountered in the test data, a list named `trackID\_vec` is initialized or reset to store that user's track IDs.

- A corresponding `predRatings` array is also reset, intended to store predicted ratings for up to 6 tracks for the user.

Prediction Logic ( Main ):

- If the current line in the test data belongs to the same user as the previous line, the track ID from the current line is appended to `trackID\_vec`.

- For each track associated with the current user, the code checks if there's a rating for the corresponding album or artist in the training data.

- If such a rating exists, it is used to predict the user's rating for the track (binary prediction: `1` if any rating exists, `0` otherwise).

- This prediction logic assumes that a user's rating for an album or artist can indicate their preference for individual tracks.

Output to CSV:

- Once all tracks for a user have been processed, or when transitioning to data for a new user, the predictions (`trackID\_pred` and binary `predRating`) are written to the output CSV file.

- The `TrackID` in the output file is formatted as `UserID\_TrackID`, and `Predictor` is the binary predicted rating.

Finalizing Output:

- After processing all lines in the test data, the predictions for the last user are output to the CSV file.

This approach essentially uses a user's historical ratings for albums and artists as a basis to predict their preferences for specific tracks. We achieved a score of 0.856 .