SAR Version 1

1. We imported necessary libraries and read the cleaned workspace data file into a Pandas DataFrame called clean_df.

2. We generated a synthetic dataset for the SAR model, consisting of 100 users and their ratings of the workspaces in the clean_df DataFrame.

3. We initialised a data dictionary to store the workspace information from the clean_df DataFrame.

4. For each user, we generated random weights for each workspace location using np.random.uniform(1, 5, num_rows) and added them to the data dictionary under the key of the user's label.

5. We created the weighted_clean_df DataFrame by passing the data dictionary to the pd.DataFrame() constructor, showing the weighted average user rating for each workspace.

6. We calculated average ratings for each workspace category across all users using the groupby() method on the weighted_clean_df DataFrame.

7. We transposed the resulting DataFrame using the .T method.

8. We dropped the "Workspace_Id" row from the category_averages_df DataFrame.

9. We created a relevant_train_df DataFrame containing the workspace Id along with the rating, price range, category, latitude, and longitude, which we believed to be relevant for comparison between workspaces.

10. We created a workspace to workspace affinity matrix based on the synthetic dataset generated, calculating the cosine similarity between all pairs of workspaces.

11. We calculated recommendation scores for each workspace for a hypothetical user "User_1" using the workspace to workspace affinity matrix and the affinity vector for "User_1" generated.

12. We added the resulting recommendations to a new DataFrame called user_1_rec, which we sorted in descending order based on the recommendation scores.

13. We extracted the top 5 recommended workspaces for "User_1" and stored them in a new DataFrame called top_5_workspaces.

14. We print the details for the top 5 recommended workspaces for User_1, including the name, address, category, price range and overall rating.

15. Finally, we print the details for the top 5 recommended workspaces for User_1, using a for loop. For each of the top 5 recommended workspaces, we increment a ‘top’ variable to keep track of the ranking, print the top choice number using an f-string, and call the ‘print_workspace()’ function to print the details of the recommended workspace. This function extracts relevant information from the ‘clean_df’ dataframe and formats it for display.

In summary, the SAR model for the workspace recommendation engine generates a workspace-to-workspace affinity matrix based on the the features Rating, Price_Range (label encoded), Category (label encoded), Latitude and Longitude to get the cosine similarity between each pair of workspaces and store that as a matrix, and then uses this matrix to calculate recommendation scores for each workspace location based on a user’s affinity vector. The top recommended workspaces for the user are then determined by sorting the recommendation scores and selecting the highest scoring workspaces. The details for the top recommended are printed to the console for the user to see.