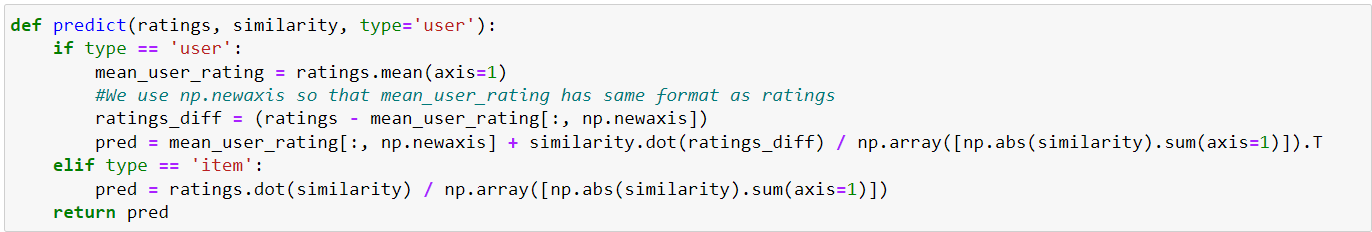
**Collaborative recommendation**



Making predict the collaborative filtering recommendation system. Collaborative filtering is a technique commonly used in recommendation systems and it can be implemented in various ways such as user based or item based collaborative filtering. This code appears to be implementing user based and item based collaborative filtering.

**def predict(ratings, similarity, type=user):**

This function is defined with three parameters: **ratings, similarity** and an optional parameter **type** which defaults to **user**.

It checks the value of the **type** parameter to determine whether to perform user-based or item based collaborative filtering:

If **type** is **user**, it executes the following code block for user based collaborative filtering:

**mean\_user\_rating=rating.mean(axis=1):** calculates the mean rating for each user by taking the average of the ratings along the rows(axis=1).

**ratings\_diff=(ratings-mean\_user\_ratting[:,np.newaxis]):**  This subtracts the mean user rating from each users ratings broadcasting the **mean\_user\_rating** to have the same shape as the **ratings** matrix.

**Pred=mean\_user\_rating[:,np.newaxis]+similarity.dot(ratings\_diff)/np.array([np.abs(similarity).sum(axis=1)]).T:** This calculates the **predicted ratings** using a weighted sum of similarity scores between users and the differences between their ratings.it uses the dot product of the **similarity** matrix with the centered **ratings.diff matrix**. The result is divided by the sum of absolute similarities for each user to normalize the prediction.

If **type** is **item**, it executes the following code black for item based collaborative filtering:

**Pred=ratings.dot(similarity)/np.array([np.abs(similarity).sum(axis=1)]):** calculates thepredicted ratings for items using a weighted sum of similarity scores between items. It uses the dot product of the ratings matrix with the similarity matrix and normalizes the prediction by dividing it by the sum of absolute similarities for each item.

The function returns the **pred** matrix which contains the predicted ratings for users or items depending on the chosen collaborative filtering type.

Its important to note that the code expects **ratings and similarity** as inputs which should be numpy arrays or matrices. Additionally, the code assumes that the input data is properly preprocessed and that similarity scores have already been computed.

This code is a useful function for making recommendations using collaborative filtering techniques either user based or item based depending on the value of the **type** parameter.