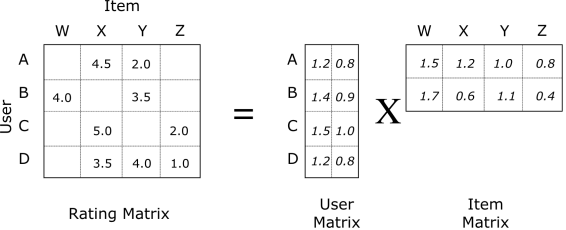
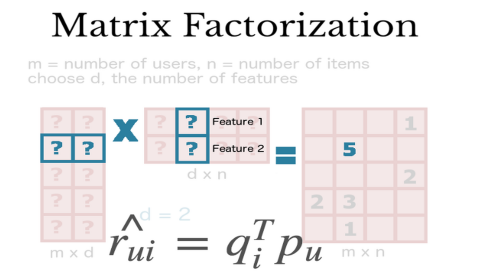
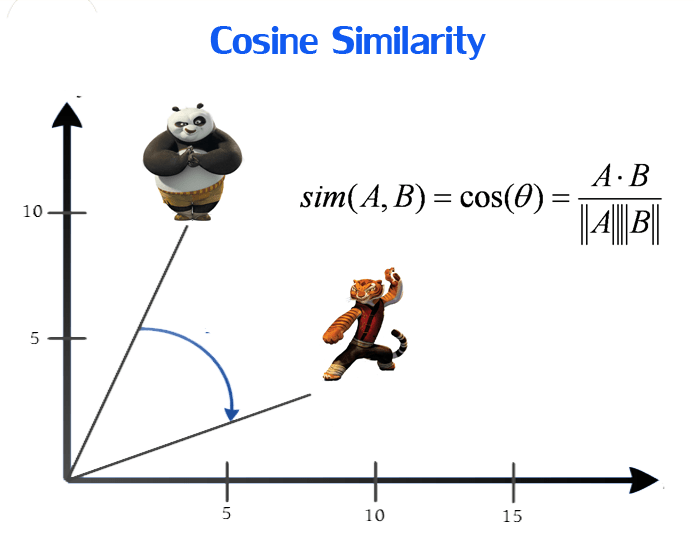
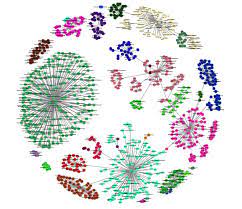
1. **Matrix Factorization(Collaborative filtering)** - Given a list of users and their ratings for different products, we will try to generate their ratings for the other products/ the missing ratings. This will be done using Matrix Factorization techniques. Some of the algorithms for this purpose are **Alternating Least squares, SVD, Stochastic Gradient Descent**. Here, we will first form a ‘Utility’ matrix with each user as a row and each product as the column, we consider this as the matrix M. we will try to find 2 matrices such that . For e.g, if M is of size , we will try to find 2 matrices of size and **r** , such that dot product of any row and column vector gives us the rating of a user for a product, that is , , gives us the most approximate value for , the element at row i and column j. is considered as the no. of features, where higher the **r**, better the approximation to the original matrix . The entries in M which are missing will get filled up during the process of finding In the training phase, the algorithms will try to find the best values for that will minimize the cost function. At the end of the training process , the approximate matrix , will be close to M.
   1. For the above algorithm, we will not require any details about the users or the products themselves.



1. **Clustering Algorithm** - Above algorithm will face a problem(cold-start problem) when there is a new user or new item, since we have no idea about which user prefers the new item or which item the new user prefers. In this case, we will try to use details about the item/product(**metadata**), to find products which are most similar to this new item. **K-means** is one of the clustering algorithms, which will use some kind of similarity metric, for e.g. , ‘cosine’ similarity to find which item vectors are most similar to the new item.



1. \*\* There are 2 different types of feedback that Users can give for products, **implicit,** where User purchasing or interacting with a product is considered as 1 and no interaction is considered as 0 and **explicit,** where User gives a rating to the item / product .We will be using subset(s) of the the Amazon Reviews dataset. We will look into the size of the varioys categories of data and decide which ones to use by the next milestone. For the Clustering part, we will be using the **Metadata** dataset provided in the same link. This dataset will contain details of the items/ products.
2. The most important hyperparameter in our algorithm will be the value , which will drive the strength of our recommendation. We will run the training process with different values of .
3. To measure the performance of our recommendations on **existing users,** we will mask some of the entries for some of the Users, and after finding our approximate Matrix, calculate the error. The error will be one value , either RMSE or MAE, which will be calculated by observing the difference in the entries in the already rated values and the predicted ratings value.
4. **To summarize, our work will be as follows -** 
   1. Identify Datasets to work on.
   2. Create a pipeline to run training and testing methods iteratively.
   3. Generate Utility Matrix of users and items using available Datasets
   4. Find the best Low-Rank approximation matrix using one of the Matrix Factorization Techniques and by checking this for different values of
   5. Address the Cold-Start problem, using most popular products.
   6. **Output - For any User U, the recommended products will be the items (which are not already rated by U) which get the best predicted ratings by U in the approximation Matrix M. Research Question - Recommend product to a user based on purchase history of users with similar likings.**
   7. (Optional) Increase dataset size or product categories and again run above steps.