Design Document

Data Structures-

The data is loaded into pandas dataframe and the relevant information is extracted to construct an MxN utility matrix in the form of a numpy matrix were M= No. of Users and N= No of movies. The result data have been loaded into the new matrices created and the reconstruction error and other parameters are loaded into other basic data type variables. The results are plotted in the graphs using the pyplot library of python.

The following are the main functions written and the data structure used for the return type are also mentioned for reference:

1) getEigenPairs (M,k)
This function computes the eigenvalues and eigenvectors using a generalised power iteration method. **Power iteration actually just calculates the principal eigenvector, but we iteratively keep on reducing the matrix to find eigenvectors corresponding to largest k eigenvalues.**
Arguments:
M: Square Matrix of which we want to calculate eigenvalues and eigenpairs.
k: Number of eigenpairs we want to calculate. k can can range from 1 to no. Of rows or collumns of M.
Returns:
Val: list of k eigenvalues caluculated
Vec: list of corresponding eigenvectors as list of numpy column arrays.

2) SVD(AAT,ATA,k):
This function calculates Singular Value Decomposition of the matrix.
Arguments:
AAT,ATA:
A*A^T (Dimensions: MxM) and A^T*A (Dimensions NxN) where A is the MxN utility matrix which we want to decompose.
k: Rank of SVD we want to compute/ no. of singular values we want to consider /No of latent dimensions we want to break our space into
Returns:
U: User to concept matrix (dimension: M*k)
Sigma: Matrix of singular values (dimension: k*k)
V: Movie to concept matrix. (dimension: k*N)

3) getCUR(A,r):
This function calculates the CUR decomposition of utility matrix A breaking it into r latent factors.
Arguments:
A: Utility matrix that we want to decompose.
R: Rank.
Returns:
C: Matrix of randomly chosen columns of A using the probability function defined by the length of each collumn (list pfc in the code).
U: Moore Penrose Pseudoinverse of the matrix obtained by taking the intersection of C and R matrices.
R: Matrix of randomly chosen rows of A using the probability function defined by the length of each row (list pfr in the code).

4) **getU(W):**

Calculates the pseudoinverse of matrix W. Used in the function getCUR for calculating U.

Arguments:

W: Matrix for of which we want to calculate the pinv.

Returns:

U: Pseudoinverse of W.

5) signU(A,u,v):

Eigenvectors corresponding to a singular value can be of two opposite directions. Out of these two directions only one gurantees minimum reconstruction error. The correct direction is got by fixing v and multiplying u such that Avi/ei is positive where ei is the i th eigenvector of AA^T