Documentation:

Module: util.py

util.readMatrix(path **)**

Load a matrix from a csv file specified by the path

Parameters:	**path : str**
	The file name (string) from where the matrix will be loaded.
Returns:	**urm : ndarray**
	A matrix containing the loaded data.

util.train_test_split(matrix, trate=0.1 **)**

Split the matrix into two matrices: one containing the training data and one containing the testing data

Parameters:	**matrix : ndarray**
	The ndarray matrix which contains the data.
	trate : float
	Amount split for testing data. Floating value between 0 and 1
Returns:	**train : ndarray**
	A matrix containing the training data.
	test : ndarray
	A matrix containing the testing data.

util.rmse(pred, actual **)**

Calculates the Root mean squared error between the actual and predicted values

Parameters:

pred : ndarray

The ndarray matrix which contains the predicted data.

actual : ndarray

The ndarray matrix which contains the actual data.

Returns:

error : float

Root mean square error

util.precision_topk(pred, actual, k, mark=3.5 **)**

Split the matrix into two matrices: one containing the training data and one containing the testing data

Parameters:

pred : ndarray

The ndarray matrix which contains the predicted data.

actual : ndarray

The ndarray matrix which contains the actual data.

k: int

The closest k similar items.

mark : float

The point above which the movie is relevant.

Returns:

precision : float

The precision

util.spearman_corr(pred, actual **)**

Calculates the Root mean squared error between the actual and predicted values

Parameters:

pred : ndarray

The ndarray matrix which contains the predicted data.

actual : ndarray

The ndarray matrix which contains the actual data.

Returns:

val : float

The spearman correlation

colab.py

findSimilarity(ratings, epsilon=1e-9 **)**

Finds the similarity matrix given an input matrix

Parameters:

ratings : ndarray

The ndarray matrix which contains the data.

epsilon : float

The small value added to prevent zero error

Returns:

similarity : ndarray

Returns the similarity matrix

predict_topk(ratings, similarity, k=40 **)**

Finds the matrix with predicted ratings with bias included

Parameters:

ratings : ndarray

The ndarray matrix which contains the data.

similarity : ndarray

The similarity matrix

k : int

The closest k-items considered to calculate predicted score

Returns:

predicted: ndarray

Returns the predicted matrix

predict_topk_nobias(ratings, similarity, k=40 **)**

Finds the matrix with predicted ratings without bias

Parameters:

ratings : ndarray

The ndarray matrix which contains the data.

similarity : ndarray

The similarity matrix

k : int

The closest k-items considered to calculate predicted score

Returns:

predicted: ndarray

Returns the predicted matrix

cur.py

select_cols(mat, k **)**

Returns the matrix with selected columns given an input matrix

Parameters:

mat : ndarray

The ndarray matrix which contains the data.

k : int

The number of columns to select

Returns:

C : ndarray

Returns the matrix with the selected columns

col_ind : ndarray

The array that contains the indices of the chosen columns

select_rows(mat, k **)**

Returns the matrix with selected rows given an input matrix

Parameters:

mat : ndarray

The ndarray matrix which contains the data.

k : int

The number of rows to select

Returns:

R : ndarray

Returns the matrix with the selected rows

row_ind : ndarray

The array that contains the indices of the chosen rows

pseudoInverse(W, reduce=False **)**

Returns the pseudoinverse of a given input matrix

Parameters:

W : ndarray

The ndarray matrix which contains the data.

reduce : boolean

	Toggles whether the reduction takes place during decomposition or not
Returns:	**WP : ndarray** Returns the pseudo-inverse matrix

Module : SVD.py

readURM( )

Reads dataset and splits into test and train after user bias correction

Parameters:	**None**
Returns:	**User_bias_1d: ndarray** 1d array of bias for each user **urm: ndarray** User Rating Matrix for training **test_matrix: ndarray** Matrix containing actual user ratings to be tested

ComputeSVD( )

Uses getSVD() and modifies to the appropriate format to return SVD matrices

Parameters:	**urm: ndarray** User Rating Matrix for training **K : int** Rank/No. Of required Dimensions
Returns:	**U: ndarray** U in SVD **S: ndarray** S in SVD **Vt: ndarray** V in SVD

getSVD( )

Returns Singular Value Decomposition

Parameters:	**urm: ndarray** User Rating Matrix for training **K : int** Rank/No. Of required Dimensions

Returns:	**U: ndarray** U in SVD **S: ndarray** S in SVD **Vt: ndarray** V in SVD

computeEstimatedRatings( )

Returns Estimated Rating Matrix by reconstructing after dimension reduction

Parameters:	**urm: ndarray** User Rating Matrix for training **U: ndarray** U in SVD **S: ndarray** S in SVD **Vt: ndarray** V in SVD

User_bias_1d: ndarray

1d array of bias for each user

K : int

Rank/No. Of required Dimensions

Returns:

estimatedRatings: ndarray

Estimated Rating matrix for all users to all movies.