4.1. Item-based Collaborative Filtering for Recommendation (Mahout)

by Vishal Doshi (vdoshi3@uic.edu)

The RecommenderTest.java will do an Item Based Collabarative Filtering using Pearson's Correlation as a similarity measure using Mahout.

Please provide user_id of user, of which you want the the movie recommendations as an argument to the program

Format: java <CLASS-NAME> <USERID>

NOTE: Make sure there is a folder name "input" with MovieLense's dataset's files: "u.item" and "u.data" [ie. input/u.item and input/u.data]

Steps:

1. Extract and Import "IBCFMahout.zip" into Eclipse.
2. Provide "User_id" as an argument, of which you want top recommended movies.

4.2. Topic Modeling using LDA(Spark)

by Vishal Doshi (vdoshi3@uic.edu)

NOTE: make sure a directory "ldaFiles" with "sparkCompatibleLDAInput.txt", "docword.nips.txt" and "vocab.nips.txt"
NOTE 2: To create "sparkCompatibleLDAInput.txt" , run LDAInputGenerator.java.

Steps:

1. Extract and Import "LDATopicModeling.zip" into Eclipse.
2. Paste *"docword.nips.txt" and "vocab.nips.txt"* in "ldaFiles" directory.
3. Run "LDAInputGenerator.java" to generate *"sparkCompatibleLDAInput.txt"*
4. Run "LDATopicModeling.java" and the output would be stored in files "documentTopicDistribution.txt" and "topTopics.txt"

Both packages contain the outputs generated in a directory named: sample outputs