

1. The data should be in the **bytecup2016data** folder.

Sr. No.	Purpose	Name
1.	train file containing qid, uid, label	invited_info_train.txt
2.	Validation data (to predict)	validate_nolabel.txt
3.	Test data (final test dataset)	test_nolabel.txt
4.	Expert information	user_info.txt
6.	Predicted labels for validation (<i>by Non-negative matrix factorization</i>)	temp_NMF.csv
7.	Predicted labels for test (<i>by Non-negative matrix factorization</i>)	final_NMF.csv
8.	Predicted labels for validation (<i>by collaborative filtering</i>)	temp_CF.csv
9.	Predicted labels for test (<i>by collaborative filtering</i>)	final_CF.csv
10.	Predicted labels for validation (<i>by hybrid recommendation</i>)	temp_hybrid.csv
11.	Predicted labels for test (<i>by hybrid recommendation</i>)	final_hybrid.csv
12.	Predicted labels for validation (<i>by Decision Tree</i>)	DecisionTree_ValidationOutput.csv
13	Predicted labels for test (<i>by Decision Tree</i>)	DecisionTree_TestOutput.csv

Executable Programs:

Sr. No.	Prupose	File	Library dependencies
1.	Generates features for training, validation and test data	FeatureGeneration.py	Pandas
2.	Code for hybrid recommendation system	RecSysHybrid.py	GraphLab, pandas, numpy
3.	Code for collaborative filtering	CollaborativeFiltering.py	Numpy, pandas
4.	Code for non-negative matrix factorization	NMF-Code.py	Sklearn, pandas, numpy
5.	Code for Decision Tree	DecisionTree.py	Pandas, Sklearn
6.	Code for Dimensionality Reduction	AutoEncoder.py	Pandas, Numpy, Keras
7.	Code for calculating NDCG scores	eval_ndcg_on_results.py	Checks the performance of the predicted outputs by ndcg scoring

2. Run any python file in the above table by calling it as `python <filename>`
3. For evaluating the model, use `eval_ndcg_on_results.py` by calling it as `python eval_ndcg_on_results.py <[optional argument]>`