REPORT

Q1) Collaborative Filtering

Approach:

1)I converted the dataset into a data frame format where rows represent users, columns represent movies and values represent ratings.

2)Fitted the dataset into Nearest Neighbor algorithm

3)Created a nn function which calculates 15 nearest neighbors of each user and then calculates the weight similarities between two user pairs at a time in a 2D matrix using the formula given in homework research paper.

4)Created a prediction function which takes userid and movieid as input and calculates predicted movie rating based on the formula in the paper.

Time taken to execute: Took around 4 hours to find predictions of all test data points in Google collab, might take more time in other IDE and environment

Results:

RMSE on test (100478) data points: 0.9886979

MAE on test data points: 0.790226

Q2) Please refer the .ipynb files for the precomputed outputs of every code if you like to see more detailed score report of combination of hyperparameters.

1) Use the SVM classifier in scikit learn and try different kernels and values of penalty parameter. Important: Depending on your computer hardware, you may have to carefully select the parameters (see the documentation on scikit learn for details) in order to speed up the computation. Report the error rate for at least 10 parameter settings that you tried (see how it is reported on http://yann.lecun.com/exdb/mnist/). Make sure to precisely describe the parameters used so that your results are reproducible.

Parameter Settings and their Accuracies and Errors

SVM

Kernel	C	Accuracy	Error
rbf	6	0.9839	0.0161
rbf	32	0.9834	0.01659

rbf	16	0.9833	0.0167
rbf	1	0.9792	0.0208
Poly	32	0.9785	0.02149
Poly	6	0.9783	0.0217
Poly	16	0.9779	0.0221
Poly	1	0.9771	0.0229
Linear	1	0.9404	0.0596
Linear	6	0.9325	0.0675
Linear	16	0.9295	0.0705
Linear	32	0.9278	0.0722
sigmoid	1	0.7759	0.224
sigmoid	6	0.768	0.2319
sigmoid	16	0.7672	0.2328
sigmoid	32	0.7633	0.2367

Inference: On kernel='rbf', C=6 best Accuracy is achieved.

2) Use the MLPClassifier in scikit learn and try different architectures, gradient descent schemes, etc. Depending on your computer hardware, you may have to carefully select the parameters of MLPClassifier in order to speed up the computation. Report the error rate for at least 10 parameters that you tried. Make sure to precisely describe the parameters used so that your results are reproducible.

MLP Classifier

H	Alpha	Optimizer	Learning Rate	Activation	Train Accuracy	Test Accuracy	Test Error
200	1.00E-04	Sgd	0.2	relu	1	0.9838	0.0162
50	1.00E-04	Sgd	0.2	Relu	1	0.9744	0.0256
50	1.00E-04	sgd	0.1	logistic	0.9999	0.974	0.026
50	1.00E-04	lbfgs	0.001	relu	0.9989	0.9658	0.0342
50	1.00E-04	Lbfgs	0.1	Logistic	0.9997	0.9628	0.0372
10	1.00E-04	sgd	0.1	logistic	0.95595	0.9378	0.0622
10	1.00E-04	sgd	0.1	relu	0.94175	0.9273	0.0727
10	1.00E-04	sgd	0.1	tanh	0.95123	0.9165	0.0835
10	1.00E-04	sgd	0.001	logistic	0.9142	0.9164	0.0836
10	1.00E-04	adam	0.1	tanh	0.8958	0.8949	0.1051
10	1.00E-04	adam	0.1	relu	0.6584	0.6566	0.3434
10	1.00E-02	adam	0.1	relu	0.3751	0.3714	0.6286
10	1.00E-04	adam	0.2	relu	0.1124	0.1135	0.8865

3) Use the k Nearest Neighbors classifier called KNeighborsClassifier in scikit learn and try different parameters (see the documentation for details). Again, depending on your computer hardware, you may

have to carefully select the parameters in order to speed up the computation. Report the error rate for at least 10 parameters that you tried. Make sure to precisely describe the parameters used so that your results are reproducible.

KNN

K	Algorithm	Test Accuracy	Error
7	Ball Tree	96.94%	3.05%
7	Brute	96.94%	3.05%
1	Kd tree	96.91%	3.09%
5	Ball Tree	96.88%	3.12%
4	Ball Tree	96.82%	3.18%
8	Ball Tree	96.70%	3.30%
25	Ball_tree	96.09%	3.91%
40	Brute	95.60%	4.40%
63	Brute	95.09%	4.91%
Also tried different weights but they were of no use.			

4) What is the best error rate you were able to reach for each of the three classifiers? Note that many parameters do not affect the error rate and we will deduct points if you try them. It is your duty to read the documentation and then employ your machine learning knowledge to determine whether a particular parameter will affect the error rate. Finally, don't change just one parameter 10 times; we want to see diversity

Best Error Rate which is 0.0161**for SVM Classifier comes with**

Test Error for kernel='rbf', C=6, Optimization= sgd, is 0.0161(1.61%)

Best Error Rate which is 0.0162 **for MLP Classifier comes with**
Test Error for Hidden Layers= 200, Alpha= 0.0001, Optimization= sgd, Learning Rate= 0.2, and Activation= relu is 0.0162 (1.62%)

Best Error Rate which is 3.05% for KNN Classifier comes with
K=7, Algorithm=Ball Tree is 0.0305