Submission for Mohammed Allama Hossain.

Problem	Filename
1. Collaborative Filtering	finalCollabFiltering.py
2. Neural Networks, K-nearest neighbours and SVMs	knn.py mlpClassifier.py svm.py

Kindly note that none of the files take any argument.

Problem 1.

Kindly change the TRAIN_PATH and TEST_PATH at line number 13 and 14 respectively in the file finalCollabFiltering.py

Kindly provide the complete path up to the filename in both the mentioned paths.

The training data was filtered using the K Nearest Neighbours algorithm; setting n = 20 and using the algorithm 'ball_tree'.

The weight w(a, i) of the user for which rating is to be determined in respect to i other users was found using the function cosine from the package scipy.

The test data was randomized and a sample of 300 was extracted for obtaining the results. The results are as follows:

Mean Absolute Error: 0.9833333333333333

Root Mean Squared Error: 2.17

Problem 2.

SVM Results

1.	kernel='linear', C=2.0	Error Rate: 0.06189999999999996 Accuracy: 0.9381
2.	kernel='poly', gamma='auto', max_iter=10, degree=3	Error Rate: 0.8298 Accuracy: 0.1702
3.	kernel='rbf', gamma='scale', C=5.0, tol=1e-4	Error Rate: 0.15900000000000025 Accuracy: 0.9841
4.	kernel='sigmoid', tol=0.1, C=1.0	Error Rate: 0.22160000000000004 Accuracy: 0.7784
5.	kernel='linear', gamma='scale', max_iter=50, random_state=1	Error Rate: 0.36970000000000006 Accuracy: 0.63029999999999994
6.	kernel='poly',	Error Rate: 0.8991

	coef0=5.5, tol=5.5	Accuracy: 0.1009
7.	kernel='rbf', class_weight='balanced', max_iter=-1, C=1.0	Error Rate: 0.20399999999999974 Accuracy: 0.9796000000000001
8.	kernel='sigmoid', coef0=0.68, degree=15, random_state=1	Error Rate: 0.28449999999999996 Accuracy: 0.7155
9.	kernel='poly', degree=20, coef0=50.08, class_weight='balanced', max_iter=-1	Error Rate: 0.22399999999999975 Accuracy: 0.9776
10.	kernel='rbf', tol=90.76, gamma='auto', degree=10, class_weight='balanced'	Error Rate: 0.8991 Accuracy: 0.1009

MLPClassifier Results

1.	solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(6,), random_state=1	Error Rate: 0.09999999999999998 Accuracy: 0.9
2.	solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(5, 2), random_state=1	Error Rate: 0.5864 Accuracy: 0.4136
3.	solver='sgd', activation='logistic',alpha=1, learning_rate_init=.1, early_stopping=True	Error Rate: 0.09099999999999997 Accuracy: 0.909
4.	hidden_layer_sizes=(10, 5), max_iter=1000	Error Rate: 0.06799999999999995 Accuracy: 0.932
5.	hidden_layer_sizes=(100,), max_iter=480, alpha=1e-4, solver='sgd', verbose=10, tol=1e-4, random_state=1, learning_rate_init=.1	Error Rate: 0.020000000000000018 Accuracy: 0.98
6.	hidden_layer_sizes=(100,), max_iter=480, alpha=1e-4, solver='sgd', verbose=10, tol=1e-4, random_state=1, learning_rate_init=.1	Error Rate: 0.020000000000000018 Accuracy: 0.98
7.	solver='lbfgs', alpha=0.1, random_state=1, max_iter=2000,	Error Rate: 0.023299999999999987 Accuracy: 0.9767

	early_stopping=True, hidden_layer_sizes=[100, 100]	
8.	solver='lbfgs', alpha=10.0, random_state=1, max_iter=2000, early_stopping=True, hidden_layer_sizes=[100, 100]	Error Rate: 0.01859999999999995 Accuracy: 0.9814
9.	solver='lbfgs', activation='tanh', alpha=1, hidden_layer_sizes=(150,100,50,50), random_state=1, max_iter=1000	Error Rate: 0.02080000000000004 Accuracy: 0.9792
10.	solver='lbfgs', alpha=1.0, random_state=1, max_iter=2000, early_stopping=True, hidden_layer_sizes=[100, 100]	Error Rate: 0.02290000000000003 Accuracy: 0.9771

KNeighborsClassifier Result

1.	n_neighbors=9	Error Rate: 0.03410000000000002 Accuracy: 0.9659
2.	leaf_size=50, n_neighbors=9, p=2	Error Rate: 0.03410000000000002 Accuracy: 0.9659
3.	leaf_size=10, n_neighbors=9, p=1, algorithm='ball_tree'	Error Rate: 0.0403 Accuracy: 0.9597
4.	n_neighbors=9, algorithm='kd_tree'	Error Rate: 0.03410000000000002 Accuracy: 0.9659
5.	n_neighbors=9, algorithm='brute'	Error Rate: 0.03410000000000002 Accuracy: 0.9659
6.	algorithm='ball_tree', n_neighbors=9, leaf_size=10, n_jobs=3	Error Rate: 0.03410000000000002 Accuracy: 0.9659
7.	algorithm='brute', n_neighbors=9 , n_jobs=5, p=2, metric='minkowski'	Error Rate: 0.03410000000000002 Accuracy: 0.9659
8.	algorithm='kd_tree', p=2, leaf_size=30, weights='uniform', n_neighbors=11	Error Rate: 0.03320000000000001 Accuracy: 0.9668
9.	algorithm='auto',	Error Rate: 0.033499999999999974

	leaf_size=50, weights='distance', n_jobs=5, n_neighbors=13	Accuracy: 0.9665
10.	algorithm='brute', n_jobs=10, weights='uniform', n_neighbors=15	Error Rate: 0.036699999999999955 Accuracy: 0.9633

Best Error Rate

Classifier	Error Rate (Rounded to 4 decimal places)
SVM	0.0619
MLPClassifier	0.0186
KNN	0.0332