用这些算法搞训练集和测试集

- 训练集不分割
- 测试集：不分割，按年龄分割成5组

In [1]:

```
import pandas as pd
import numpy as np
from glob import glob
from time import time

from surprise import Reader
from surprise import Dataset
from surprise.model_selection import cross_validate
from surprise import NormalPredictor
from surprise import KNNBasic
from surprise import KNNWithMeans
from surprise import KNNWithZScore
from surprise import KNNBaseline
from surprise import SVD
from surprise import BaselineOnly
from surprise import SVDpp
from surprise import NMF
from surprise import SlopeOne
from surprise import CoClustering
from surprise.accuracy import rmse, mae
from surprise import accuracy
from surprise.model_selection import train_test_split
from surprise.model_selection import GridSearchCV

from plotly.offline import init_notebook_mode, plot, iplot
import plotly.graph_objs as go
init_notebook_mode(connected=True)
```

In [16]:

```
def build_train_test_age(df_train, df_test):
    reader = Reader(rating_scale=(1, 5))
    data_train = Dataset.load_from_df(df_train[['user_id', 'movie_id', 'rating']], reader)
    data_train = data_train.build_full_trainset()
    data_test = Dataset.load_from_df(df_test[['user_id', 'movie_id', 'rating']], reader)
    data_test = data_test.build_full_trainset().build_testset()

    df_test_age1 = df_test[df_test['age'] < 20]
    data_test_age1 = Dataset.load_from_df(df_test_age1[['user_id', 'movie_id', 'rating']], reader)
    data_test_age1 = data_test_age1.build_full_trainset().build_testset()
    df_test_age2 = df_test[(df_test['age'] >= 20) & (df_test['age'] < 30)]
    data_test_age2 = Dataset.load_from_df(df_test_age2[['user_id', 'movie_id', 'rating']], reader)
    data_test_age2 = data_test_age2.build_full_trainset().build_testset()
    df_test_age3 = df_test[(df_test['age'] >= 30) & (df_test['age'] < 40)]
    data_test_age3 = Dataset.load_from_df(df_test_age3[['user_id', 'movie_id', 'rating']], reader)
    data_test_age3 = data_test_age3.build_full_trainset().build_testset()
    df_test_age4 = df_test[(df_test['age'] >= 40) & (df_test['age'] < 50)]
    data_test_age4 = Dataset.load_from_df(df_test_age4[['user_id', 'movie_id', 'rating']], reader)
    data_test_age4 = data_test_age4.build_full_trainset().build_testset()
    df_test_age5 = df_test[df_test['age'] >= 50]
    data_test_age5 = Dataset.load_from_df(df_test_age5[['user_id', 'movie_id', 'rating']], reader)
    data_test_age5 = data_test_age5.build_full_trainset().build_testset()

    return data_train, data_test, [data_test_age1, data_test_age2, data_test_age3, data_test_age4, data_test_
age5]
```

In [19]:

```
algorithms = {'SVD':SVD(), 'SVDpp':SVDpp(), 'SlopeOne':SlopeOne(), 'NMF':NMF(), 'NormalPredictor':NormalPredi
ctor(),
              'KNNBaseline':KNNBaseline(), 'KNNBasic':KNNBasic(), 'KNNWithMeans':KNNWithMeans(),
              'KNNWithZScore':KNNWithZScore(), 'BaselineOnly':BaselineOnly(), 'CoClustering':CoClustering()}
```

In [27]:

```
def train_single_algorithm_age(algorithm_name, data_train, data_test, data_test_age_list, save_model=False):
    algorithms = {'SVD':SVD(), 'SVDpp':SVDpp(), 'SlopeOne':SlopeOne(), 'NMF':NMF(), 'NormalPredictor':NormalP
redictor(),
              'KNNBaseline':KNNBaseline(), 'KNNBasic':KNNBasic(), 'KNNWithMeans':KNNWithMeans(),
              'KNNWithZScore':KNNWithZScore(), 'BaselineOnly':BaselineOnly(), 'CoClustering':CoClustering()}
    assert(algorithm_name in algorithms), "{} does not exist!".format(algorithm_name)
    algo = algorithms[algorithm_name]
    start_time = time()
    algo.fit(data_train)
    # test
    predictions = algo.test(data_test)
    result = {}
    result['rmse'] = accuracy.rmse(predictions, verbose=True)
    result['mae'] = accuracy.mae(predictions, verbose=True)

    # test age_list
    assert (len(data_test_age_list) == 5), "5 age groups!"
    for i in range(5):
        rmse_name = "rmse_age{}".format(i+1)
        mae_name = "mae_age{}".format(i+1)
        predictions_age = algo.test(data_test_age_list[i])
        result[rmse_name] = accuracy.rmse(predictions_age, verbose=True)
        result[mae_name] = accuracy.mae(predictions_age, verbose=True)

    if save_model:
        result['model'] = algo

    print("===== ===== ===== ===== =====")
    print_result = "{:<15}|{:.2f} mins|rmse: {:.4f}|mae: {:.4f}"
    print_result = print_result.format(algorithm_name, (time() - start_time) / 60.,result['rmse'],result['ma
e'])
    print(print_result)
    for i in range(5):
        rmse_name = "rmse_age{}".format(i+1)
        mae_name = "mae_age{}".format(i+1)
        print("AgeGroup      {}|rmse: {:.4f}|mae: {:.4f}".format(i+1, result[rmse_name], result[mae_name]))
    print("===== ===== ===== ===== =====")
    return result
```

In [28]:

```
def get_mean_results(algorithms, all_results_list):
    for curr_algo_name in algorithms.keys():
        curr_algo_rmse = []
        curr_algo_mae = []
        for curr_all_results in all_results_list:
            curr_algo_rmse.append(curr_all_results[curr_algo_name]['rmse'])
            curr_algo_mae.append(curr_all_results[curr_algo_name]['mae'])
        print("{:<15}|rmse: {:.4f}+-{:.4f}|mae: {:.4f}+-{:.4f}".format(curr_algo_name,
                                                                   np.mean(curr_algo_rmse), np.std(curr_algo_
rmse),
                                                                   np.mean(curr_algo_mae), np.std(curr_algo_m
ae),
                                                                  ))
```

In [29]:

```
def get_mean_results_age(algorithms, all_results_list, age_group = 1):
    """
    The age group can be [1,2,3,4,5]
    """
    for curr_algo_name in algorithms.keys():
        curr_algo_rmse = []
        curr_algo_mae = []
        for curr_all_results in all_results_list:
            curr_algo_rmse.append(curr_all_results[curr_algo_name]['rmse_age{}'.format(age_group)])
            curr_algo_mae.append(curr_all_results[curr_algo_name]['mae_age{}'.format(age_group)])
        print("{:<15}|rmse: {:.4f}+-{:.4f}|mae: {:.4f}+-{:.4f}".format(curr_algo_name,
                                                                   np.mean(curr_algo_rmse), np.std(curr_algo_
rmse),
                                                                   np.mean(curr_algo_mae), np.std(curr_algo_m
ae),
                                                                  ))
```

u1

In [30]:

```
# load
df_train = pd.read_csv("data/ml-100k_merged/u1.base")
df_test = pd.read_csv("data/ml-100k_merged/u1.test")
df_test.head(3)
```

Out[30]:

	movie_id	movie_title	user_id	age	sex	occupation	rating
0	1	Toy Story (1995)	5	33	F	other	4
1	2	GoldenEye (1995)	5	33	F	other	3
2	17	From Dusk Till Dawn (1996)	5	33	F	other	4

In [31]:

```
data_train, data_test, data_test_age_list = build_train_test_age(df_train, df_test)
```

In [32]:

```
# start
all_results = {}
save_model = False
for algorithm_name in algorithms.keys():
    result = train_single_algorithm_age(algorithm_name, data_train, data_test, data_test_age_list, save_model
)
    all_results[algorithm_name] = result
    print("===== ===== ===== =====")
```

```
RMSE: 0.9495
MAE:  0.7494
RMSE: 0.9882
MAE:  0.7867
RMSE: 0.9535
MAE:  0.7507
RMSE: 0.9452
MAE:  0.7466
RMSE: 0.9972
MAE:  0.7910
RMSE: 0.8515
MAE:  0.6733
===== ===== ===== ===== =====
SVD            |0.06 mins|rmse: 0.9495|mae: 0.7494
AgeGroup      1|rmse: 0.9882|mae: 0.7867
AgeGroup      2|rmse: 0.9535|mae: 0.7507
AgeGroup      3|rmse: 0.9452|mae: 0.7466
AgeGroup      4|rmse: 0.9972|mae: 0.7910
AgeGroup      5|rmse: 0.8515|mae: 0.6733
===== ===== ===== ===== =====
===== ===== ===== =====
RMSE: 0.9315
MAE:  0.7292
RMSE: 0.9682
MAE:  0.7562
RMSE: 0.9334
MAE:  0.7299
RMSE: 0.9217
MAE:  0.7216
RMSE: 0.9854
MAE:  0.7769
RMSE: 0.8469
MAE:  0.6637
===== ===== ===== ===== =====
SVDpp          |2.44 mins|rmse: 0.9315|mae: 0.7292
AgeGroup      1|rmse: 0.9682|mae: 0.7562
AgeGroup      2|rmse: 0.9334|mae: 0.7299
AgeGroup      3|rmse: 0.9217|mae: 0.7216
AgeGroup      4|rmse: 0.9854|mae: 0.7769
AgeGroup      5|rmse: 0.8469|mae: 0.6637
===== ===== ===== ===== =====
===== ===== ===== =====
RMSE: 0.9567
MAE:  0.7506
RMSE: 1.0013
MAE:  0.7890
RMSE: 0.9587
MAE:  0.7501
RMSE: 0.9551
MAE:  0.7506
RMSE: 1.0039
MAE:  0.7923
RMSE: 0.8562
MAE:  0.6724
===== ===== ===== ===== =====
SlopeOne       |0.06 mins|rmse: 0.9567|mae: 0.7506
AgeGroup      1|rmse: 1.0013|mae: 0.7890
AgeGroup      2|rmse: 0.9587|mae: 0.7501
AgeGroup      3|rmse: 0.9551|mae: 0.7506
AgeGroup      4|rmse: 1.0039|mae: 0.7923
AgeGroup      5|rmse: 0.8562|mae: 0.6724
===== ===== ===== ===== =====
===== ===== ===== =====
RMSE: 0.9765
MAE:  0.7659
RMSE: 1.0222
MAE:  0.8015
RMSE: 0.9808
MAE:  0.7714
RMSE: 0.9741
MAE:  0.7632
RMSE: 1.0175
MAE:  0.7996
RMSE: 0.8785
MAE:  0.6873
===== ===== ===== ===== =====
NMF            |0.07 mins|rmse: 0.9765|mae: 0.7659
AgeGroup      1|rmse: 1.0222|mae: 0.8015
AgeGroup      2|rmse: 0.9808|mae: 0.7714
AgeGroup      3|rmse: 0.9741|mae: 0.7632
AgeGroup      4|rmse: 1.0175|mae: 0.7996
AgeGroup      5|rmse: 0.8785|mae: 0.6873
===== ===== ===== ===== =====
===== ===== ===== =====
RMSE: 1.5502
MAE:  1.2457
RMSE: 1.5278
MAE:  1.2093
```

```
RMSE: 1.5768
MAE:  1.2696
RMSE: 1.5249
MAE:  1.2247
RMSE: 1.5534
MAE:  1.2506
RMSE: 1.4563
MAE:  1.1743
===== ===== ===== ===== =====
NormalPredictor|0.01 mins|rmse: 1.5502|mae: 1.2457
AgeGroup      1|rmse: 1.5278|mae: 1.2093
AgeGroup      2|rmse: 1.5768|mae: 1.2696
AgeGroup      3|rmse: 1.5249|mae: 1.2247
AgeGroup      4|rmse: 1.5534|mae: 1.2506
AgeGroup      5|rmse: 1.4563|mae: 1.1743
===== ===== ===== ===== =====
===== ===== ===== =====
Estimating biases using als...
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9418
MAE:  0.7413
RMSE: 0.9837
MAE:  0.7818
RMSE: 0.9443
MAE:  0.7424
RMSE: 0.9375
MAE:  0.7379
RMSE: 0.9868
MAE:  0.7791
RMSE: 0.8512
MAE:  0.6705
===== ===== ===== ===== =====
KNNBaseline    |0.11 mins|rmse: 0.9418|mae: 0.7413
AgeGroup      1|rmse: 0.9837|mae: 0.7818
AgeGroup      2|rmse: 0.9443|mae: 0.7424
AgeGroup      3|rmse: 0.9375|mae: 0.7379
AgeGroup      4|rmse: 0.9868|mae: 0.7791
AgeGroup      5|rmse: 0.8512|mae: 0.6705
===== ===== ===== ===== =====
===== ===== ===== =====
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9888
MAE:  0.7833
RMSE: 1.0270
MAE:  0.8249
RMSE: 1.0225
MAE:  0.8081
RMSE: 0.9637
MAE:  0.7645
RMSE: 1.0035
MAE:  0.7979
RMSE: 0.8908
MAE:  0.7045
===== ===== ===== ===== =====
KNNBasic       |0.10 mins|rmse: 0.9888|mae: 0.7833
AgeGroup      1|rmse: 1.0270|mae: 0.8249
AgeGroup      2|rmse: 1.0225|mae: 0.8081
AgeGroup      3|rmse: 0.9637|mae: 0.7645
AgeGroup      4|rmse: 1.0035|mae: 0.7979
AgeGroup      5|rmse: 0.8908|mae: 0.7045
===== ===== ===== ===== =====
===== ===== ===== =====
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9648
MAE:  0.7593
RMSE: 1.0124
MAE:  0.8148
RMSE: 0.9686
MAE:  0.7612
RMSE: 0.9585
MAE:  0.7547
RMSE: 1.0126
MAE:  0.7958
RMSE: 0.8674
MAE:  0.6812
===== ===== ===== ===== =====
KNNWithMeans   |0.10 mins|rmse: 0.9648|mae: 0.7593
AgeGroup      1|rmse: 1.0124|mae: 0.8148
AgeGroup      2|rmse: 0.9686|mae: 0.7612
AgeGroup      3|rmse: 0.9585|mae: 0.7547
AgeGroup      4|rmse: 1.0126|mae: 0.7958
AgeGroup      5|rmse: 0.8674|mae: 0.6812
===== ===== ===== ===== =====
===== ===== ===== =====
Computing the msd similarity matrix...
Done computing similarity matrix.
```

```
RMSE: 0.9635
MAE:  0.7553
RMSE: 1.0062
MAE:  0.8050
RMSE: 0.9696
MAE:  0.7597
RMSE: 0.9555
MAE:  0.7494
RMSE: 1.0117
MAE:  0.7913
RMSE: 0.8645
MAE:  0.6763
===== ===== ===== ===== =====
KNNWithZScore  |0.11 mins|rmse: 0.9635|mae: 0.7553
AgeGroup      1|rmse: 1.0062|mae: 0.8050
AgeGroup      2|rmse: 0.9696|mae: 0.7597
AgeGroup      3|rmse: 0.9555|mae: 0.7494
AgeGroup      4|rmse: 1.0117|mae: 0.7913
AgeGroup      5|rmse: 0.8645|mae: 0.6763
===== ===== ===== ===== =====
===== ===== ===== =====
Estimating biases using als...
RMSE: 0.9599
MAE:  0.7616
RMSE: 1.0036
MAE:  0.7978
RMSE: 0.9661
MAE:  0.7675
RMSE: 0.9585
MAE:  0.7594
RMSE: 1.0017
MAE:  0.7982
RMSE: 0.8535
MAE:  0.6767
===== ===== ===== ===== =====
BaselineOnly   |0.00 mins|rmse: 0.9599|mae: 0.7616
AgeGroup      1|rmse: 1.0036|mae: 0.7978
AgeGroup      2|rmse: 0.9661|mae: 0.7675
AgeGroup      3|rmse: 0.9585|mae: 0.7594
AgeGroup      4|rmse: 1.0017|mae: 0.7982
AgeGroup      5|rmse: 0.8535|mae: 0.6767
===== ===== ===== ===== =====
===== ===== ===== =====
RMSE: 0.9817
MAE:  0.7679
RMSE: 1.0336
MAE:  0.8252
RMSE: 0.9786
MAE:  0.7640
RMSE: 0.9816
MAE:  0.7675
RMSE: 1.0305
MAE:  0.8067
RMSE: 0.8872
MAE:  0.6939
===== ===== ===== ===== =====
CoClustering   |0.02 mins|rmse: 0.9817|mae: 0.7679
AgeGroup      1|rmse: 1.0336|mae: 0.8252
AgeGroup      2|rmse: 0.9786|mae: 0.7640
AgeGroup      3|rmse: 0.9816|mae: 0.7675
AgeGroup      4|rmse: 1.0305|mae: 0.8067
AgeGroup      5|rmse: 0.8872|mae: 0.6939
===== ===== ===== ===== =====
===== ===== ===== =====
```

In [33]:

```
all_results
```

```
Out[33]: {'SVD': {'rmse': 0.9494700424337811,
           'mae': 0.7494252882154898,
           'rmse_age1': 0.9881985061027665,
           'mae_age1': 0.7866825738504812,
           'rmse_age2': 0.9535066177099851,
           'mae_age2': 0.7507318552029447,
           'rmse_age3': 0.945236044735059,
           'mae_age3': 0.7465531476619003,
           'rmse_age4': 0.9972076709793622,
           'mae_age4': 0.7910150522499189,
           'rmse_age5': 0.8515099561895363,
           'mae_age5': 0.6732726517618067},
          'SVDpp': {'rmse': 0.9314786233737918,
           'mae': 0.7291904382895492,
           'rmse_age1': 0.9681512485212258,
           'mae_age1': 0.7561624385990997,
           'rmse_age2': 0.9333643482893472,
           'mae_age2': 0.7299397601120939,
           'rmse_age3': 0.9216823478112921,
           'mae_age3': 0.7216358727999111,
           'rmse_age4': 0.985448719911958,
           'mae_age4': 0.7769219614511862,
           'rmse_age5': 0.846885638661987,
           'mae_age5': 0.6637211735404961},
          'SlopeOne': {'rmse': 0.9567192117629564,
           'mae': 0.7505898912181515,
           'rmse_age1': 1.0012541527430894,
           'mae_age1': 0.7890176562577564,
           'rmse_age2': 0.9586730103914695,
           'mae_age2': 0.7501026503142274,
           'rmse_age3': 0.9550554530539456,
           'mae_age3': 0.7505950575990022,
           'rmse_age4': 1.0039080930337734,
           'mae_age4': 0.7923236474921365,
           'rmse_age5': 0.8562157808637371,
           'mae_age5': 0.6724077299886216},
          'NMF': {'rmse': 0.9764954093002502,
           'mae': 0.7658537214312179,
           'rmse_age1': 1.022197659103353,
           'mae_age1': 0.8015064622582946,
           'rmse_age2': 0.9808428647559927,
           'mae_age2': 0.7714182473945047,
           'rmse_age3': 0.9741373988980901,
           'mae_age3': 0.7632076099167372,
           'rmse_age4': 1.01747402771253,
           'mae_age4': 0.7996163263794178,
           'rmse_age5': 0.8785170313196072,
           'mae_age5': 0.6873203050710305},
          'NormalPredictor': {'rmse': 1.5502233274431836,
           'mae': 1.2457127543181556,
           'rmse_age1': 1.5277744389449512,
           'mae_age1': 1.2093240614498035,
           'rmse_age2': 1.5768466506871708,
           'mae_age2': 1.2696273194130636,
           'rmse_age3': 1.5248778071249147,
           'mae_age3': 1.2247329281552304,
           'rmse_age4': 1.553370973561192,
           'mae_age4': 1.2506002473728912,
           'rmse_age5': 1.4562561697044087,
           'mae_age5': 1.1742913298842557},
          'KNNBaseline': {'rmse': 0.9417830614393241,
           'mae': 0.741335988489349,
           'rmse_age1': 0.9837143418766702,
           'mae_age1': 0.781808089984719,
           'rmse_age2': 0.9443294318947308,
           'mae_age2': 0.7423927883167553,
           'rmse_age3': 0.9375139490529291,
           'mae_age3': 0.7379216692366521,
           'rmse_age4': 0.9867832138705797,
           'mae_age4': 0.7790818506890305,
           'rmse_age5': 0.8511638153050718,
           'mae_age5': 0.6705099179662324},
          'KNNBasic': {'rmse': 0.9887958704696975,
           'mae': 0.7832791234223664,
           'rmse_age1': 1.0269799596779086,
           'mae_age1': 0.8248510111181293,
           'rmse_age2': 1.0224697814355022,
           'mae_age2': 0.8081263825449961,
           'rmse_age3': 0.9637455481753019,
           'mae_age3': 0.7644780148977353,
           'rmse_age4': 1.0034862068860406,
           'mae_age4': 0.7978545450695682,
           'rmse_age5': 0.890779794651895,
           'mae_age5': 0.7044610602215687},
          'KNNWithMeans': {'rmse': 0.9648479897763116,
           'mae': 0.7592897649678887,
           'rmse_age1': 1.0124007465806315,
           'mae_age1': 0.8148036557383985,
  'rmse_age2': 0.9685561618866915,
  'mae_age2': 0.7612046093109275,
  'rmse_age3': 0.9585323226276168,
  'mae_age3': 0.7546962299957217,
  'rmse_age4': 1.012583982258421,
  'mae_age4': 0.7957781458859974,
  'rmse_age5': 0.8674064602202763,
  'mae_age5': 0.6811574988713697},
 'KNNWithZScore': {'rmse': 0.9634959916083691,
  'mae': 0.7552704605650169,
  'rmse_age1': 1.0062193311901648,
  'mae_age1': 0.8049730722268396,
  'rmse_age2': 0.9696221001350047,
  'mae_age2': 0.7597476812607408,
  'rmse_age3': 0.9555455894609536,
  'mae_age3': 0.749359881770033,
  'rmse_age4': 1.0116698083205,
  'mae_age4': 0.7912675586778475,
  'rmse_age5': 0.8644815564392303,
  'mae_age5': 0.6763434261123482},
 'BaselineOnly': {'rmse': 0.9599438333077737,
  'mae': 0.7615833440531363,
  'rmse_age1': 1.0036464303435535,
  'mae_age1': 0.7977893054819026,
  'rmse_age2': 0.9660689855323669,
  'mae_age2': 0.7674763390794163,
  'rmse_age3': 0.9584590198847595,
  'mae_age3': 0.7594066527337923,
  'rmse_age4': 1.0016708237661498,
  'mae_age4': 0.7981941808106231,
  'rmse_age5': 0.8534671120208315,
  'mae_age5': 0.6767192779732},
 'CoClustering': {'rmse': 0.9816819846871367,
  'mae': 0.7678899339311189,
  'rmse_age1': 1.0336006601330072,
  'mae_age1': 0.8252447649188829,
  'rmse_age2': 0.9785609692371686,
  'mae_age2': 0.7640004937185445,
  'rmse_age3': 0.9816359886289482,
  'mae_age3': 0.7674713029859267,
  'rmse_age4': 1.0304808914582761,
  'mae_age4': 0.8067311790226308,
  'rmse_age5': 0.8871691812710419,
  'mae_age5': 0.6939098342895637}}
```

u2

In [34]:

```
# load
df_train = pd.read_csv("data/ml-100k_merged/u2.base")
df_test = pd.read_csv("data/ml-100k_merged/u2.test")
data_train, data_test, data_test_age_list = build_train_test_age(df_train, df_test)
all_results2 = {}
save_model = False
for algorithm_name in algorithms.keys():
    result = train_single_algorithm_age(algorithm_name, data_train, data_test, data_test_age_list, save_model
)
    all_results2[algorithm_name] = result
```

```
RMSE: 0.9397
MAE:  0.7382
RMSE: 0.9840
MAE:  0.7808
RMSE: 0.9525
MAE:  0.7475
RMSE: 0.9305
MAE:  0.7310
RMSE: 0.9381
MAE:  0.7387
RMSE: 0.8893
MAE:  0.6954
===== ===== ===== ===== =====
SVD            |0.07 mins|rmse: 0.9397|mae: 0.7382
AgeGroup      1|rmse: 0.9840|mae: 0.7808
AgeGroup      2|rmse: 0.9525|mae: 0.7475
AgeGroup      3|rmse: 0.9305|mae: 0.7310
AgeGroup      4|rmse: 0.9381|mae: 0.7387
AgeGroup      5|rmse: 0.8893|mae: 0.6954
===== ===== ===== ===== =====
RMSE: 0.9270
MAE:  0.7248
RMSE: 0.9383
MAE:  0.7355
RMSE: 0.9420
MAE:  0.7362
RMSE: 0.9160
MAE:  0.7135
RMSE: 0.9305
MAE:  0.7314
RMSE: 0.8894
MAE:  0.6972
===== ===== ===== ===== =====
SVDpp          |2.30 mins|rmse: 0.9270|mae: 0.7248
AgeGroup      1|rmse: 0.9383|mae: 0.7355
AgeGroup      2|rmse: 0.9420|mae: 0.7362
AgeGroup      3|rmse: 0.9160|mae: 0.7135
AgeGroup      4|rmse: 0.9305|mae: 0.7314
AgeGroup      5|rmse: 0.8894|mae: 0.6972
===== ===== ===== ===== =====
RMSE: 0.9485
MAE:  0.7423
RMSE: 0.9821
MAE:  0.7787
RMSE: 0.9685
MAE:  0.7542
RMSE: 0.9380
MAE:  0.7335
RMSE: 0.9416
MAE:  0.7419
RMSE: 0.8907
MAE:  0.6997
===== ===== ===== ===== =====
SlopeOne       |0.07 mins|rmse: 0.9485|mae: 0.7423
AgeGroup      1|rmse: 0.9821|mae: 0.7787
AgeGroup      2|rmse: 0.9685|mae: 0.7542
AgeGroup      3|rmse: 0.9380|mae: 0.7335
AgeGroup      4|rmse: 0.9416|mae: 0.7419
AgeGroup      5|rmse: 0.8907|mae: 0.6997
===== ===== ===== ===== =====
RMSE: 0.9670
MAE:  0.7579
RMSE: 0.9932
MAE:  0.7863
RMSE: 0.9866
MAE:  0.7717
RMSE: 0.9562
MAE:  0.7449
RMSE: 0.9626
MAE:  0.7619
RMSE: 0.9127
MAE:  0.7176
===== ===== ===== ===== =====
NMF            |0.07 mins|rmse: 0.9670|mae: 0.7579
AgeGroup      1|rmse: 0.9932|mae: 0.7863
AgeGroup      2|rmse: 0.9866|mae: 0.7717
AgeGroup      3|rmse: 0.9562|mae: 0.7449
AgeGroup      4|rmse: 0.9626|mae: 0.7619
AgeGroup      5|rmse: 0.9127|mae: 0.7176
===== ===== ===== ===== =====
RMSE: 1.5151
MAE:  1.2153
RMSE: 1.5695
MAE:  1.2556
RMSE: 1.5429
MAE:  1.2323
RMSE: 1.5133
MAE:  1.2160
```

```
RMSE: 1.4747
MAE:  1.1833
RMSE: 1.4803
MAE:  1.1766
===== ===== ===== ===== =====
NormalPredictor|0.01 mins|rmse: 1.5151|mae: 1.2153
AgeGroup      1|rmse: 1.5695|mae: 1.2556
AgeGroup      2|rmse: 1.5429|mae: 1.2323
AgeGroup      3|rmse: 1.5133|mae: 1.2160
AgeGroup      4|rmse: 1.4747|mae: 1.1833
AgeGroup      5|rmse: 1.4803|mae: 1.1766
===== ===== ===== ===== =====
Estimating biases using als...
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9346
MAE:  0.7326
RMSE: 0.9691
MAE:  0.7698
RMSE: 0.9508
MAE:  0.7434
RMSE: 0.9225
MAE:  0.7217
RMSE: 0.9275
MAE:  0.7321
RMSE: 0.8920
MAE:  0.6977
===== ===== ===== ===== =====
KNNBaseline    |0.12 mins|rmse: 0.9346|mae: 0.7326
AgeGroup      1|rmse: 0.9691|mae: 0.7698
AgeGroup      2|rmse: 0.9508|mae: 0.7434
AgeGroup      3|rmse: 0.9225|mae: 0.7217
AgeGroup      4|rmse: 0.9275|mae: 0.7321
AgeGroup      5|rmse: 0.8920|mae: 0.6977
===== ===== ===== ===== =====
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9848
MAE:  0.7750
RMSE: 1.0084
MAE:  0.7997
RMSE: 1.0276
MAE:  0.8113
RMSE: 0.9532
MAE:  0.7473
RMSE: 0.9517
MAE:  0.7498
RMSE: 0.9313
MAE:  0.7306
===== ===== ===== ===== =====
KNNBasic       |0.09 mins|rmse: 0.9848|mae: 0.7750
AgeGroup      1|rmse: 1.0084|mae: 0.7997
AgeGroup      2|rmse: 1.0276|mae: 0.8113
AgeGroup      3|rmse: 0.9532|mae: 0.7473
AgeGroup      4|rmse: 0.9517|mae: 0.7498
AgeGroup      5|rmse: 0.9313|mae: 0.7306
===== ===== ===== ===== =====
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9571
MAE:  0.7510
RMSE: 0.9893
MAE:  0.7959
RMSE: 0.9771
MAE:  0.7648
RMSE: 0.9461
MAE:  0.7417
RMSE: 0.9454
MAE:  0.7450
RMSE: 0.9068
MAE:  0.7052
===== ===== ===== ===== =====
KNNWithMeans   |0.09 mins|rmse: 0.9571|mae: 0.7510
AgeGroup      1|rmse: 0.9893|mae: 0.7959
AgeGroup      2|rmse: 0.9771|mae: 0.7648
AgeGroup      3|rmse: 0.9461|mae: 0.7417
AgeGroup      4|rmse: 0.9454|mae: 0.7450
AgeGroup      5|rmse: 0.9068|mae: 0.7052
===== ===== ===== ===== =====
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9576
MAE:  0.7480
RMSE: 0.9844
MAE:  0.7834
RMSE: 0.9805
MAE:  0.7634
RMSE: 0.9491
MAE:  0.7409
```

```
RMSE: 0.9404
MAE:  0.7393
RMSE: 0.9029
MAE:  0.7020
===== ===== ===== ===== =====
KNNWithZScore  |0.10 mins|rmse: 0.9576|mae: 0.7480
AgeGroup      1|rmse: 0.9844|mae: 0.7834
AgeGroup      2|rmse: 0.9805|mae: 0.7634
AgeGroup      3|rmse: 0.9491|mae: 0.7409
AgeGroup      4|rmse: 0.9404|mae: 0.7393
AgeGroup      5|rmse: 0.9029|mae: 0.7020
===== ===== ===== ===== =====
Estimating biases using als...
RMSE: 0.9477
MAE:  0.7494
RMSE: 0.9929
MAE:  0.7936
RMSE: 0.9624
MAE:  0.7616
RMSE: 0.9393
MAE:  0.7404
RMSE: 0.9442
MAE:  0.7481
RMSE: 0.8904
MAE:  0.7024
===== ===== ===== ===== =====
BaselineOnly   |0.01 mins|rmse: 0.9477|mae: 0.7494
AgeGroup      1|rmse: 0.9929|mae: 0.7936
AgeGroup      2|rmse: 0.9624|mae: 0.7616
AgeGroup      3|rmse: 0.9393|mae: 0.7404
AgeGroup      4|rmse: 0.9442|mae: 0.7481
AgeGroup      5|rmse: 0.8904|mae: 0.7024
===== ===== ===== ===== =====
RMSE: 0.9672
MAE:  0.7525
RMSE: 0.9959
MAE:  0.7916
RMSE: 0.9856
MAE:  0.7652
RMSE: 0.9584
MAE:  0.7461
RMSE: 0.9553
MAE:  0.7449
RMSE: 0.9208
MAE:  0.7092
===== ===== ===== ===== =====
CoClustering   |0.02 mins|rmse: 0.9672|mae: 0.7525
AgeGroup      1|rmse: 0.9959|mae: 0.7916
AgeGroup      2|rmse: 0.9856|mae: 0.7652
AgeGroup      3|rmse: 0.9584|mae: 0.7461
AgeGroup      4|rmse: 0.9553|mae: 0.7449
AgeGroup      5|rmse: 0.9208|mae: 0.7092
===== ===== ===== ===== =====
In [35]: all_results2
```

```
Out[35]: {'SVD': {'rmse': 0.9397139545744244,
           'mae': 0.738170076126467,
           'rmse_age1': 0.9840410051009472,
           'mae_age1': 0.7808033287178034,
           'rmse_age2': 0.9524723308487864,
           'mae_age2': 0.7474921852887403,
           'rmse_age3': 0.9305248040329619,
           'mae_age3': 0.7310083603376073,
           'rmse_age4': 0.9380865464912357,
           'mae_age4': 0.7387198997047327,
           'rmse_age5': 0.8893347634209733,
           'mae_age5': 0.6954332058272052},
          'SVDpp': {'rmse': 0.9270429383318842,
           'mae': 0.7248271018419875,
           'rmse_age1': 0.9383472701433958,
           'mae_age1': 0.7354505135710923,
           'rmse_age2': 0.9419600647392274,
           'mae_age2': 0.7361668150215925,
           'rmse_age3': 0.9159984808772474,
           'mae_age3': 0.7134891091751248,
           'rmse_age4': 0.9305352470031507,
           'mae_age4': 0.7313599093299774,
           'rmse_age5': 0.88938195890162,
           'mae_age5': 0.6972193705555464},
          'SlopeOne': {'rmse': 0.948530796138768,
           'mae': 0.7422881947846279,
           'rmse_age1': 0.9820871668388654,
           'mae_age1': 0.7786663784818674,
           'rmse_age2': 0.9684690024308309,
           'mae_age2': 0.7541881791313859,
           'rmse_age3': 0.9379802418635901,
           'mae_age3': 0.7335463987429047,
           'rmse_age4': 0.9416082581850158,
           'mae_age4': 0.7418554490254425,
           'rmse_age5': 0.8907284933939678,
           'mae_age5': 0.699705996252417},
          'NMF': {'rmse': 0.9669760788346081,
           'mae': 0.7579112571304267,
           'rmse_age1': 0.9931844252934138,
           'mae_age1': 0.7863076893830446,
           'rmse_age2': 0.986593020718887,
           'mae_age2': 0.7717198610669849,
           'rmse_age3': 0.9561896101318924,
           'mae_age3': 0.7449349418700417,
           'rmse_age4': 0.9626151471669678,
           'mae_age4': 0.7618973793305942,
           'rmse_age5': 0.912687019136504,
           'mae_age5': 0.7175869917088765},
          'NormalPredictor': {'rmse': 1.5150782162971181,
           'mae': 1.2152535992529037,
           'rmse_age1': 1.569500702086027,
           'mae_age1': 1.2555761525025224,
           'rmse_age2': 1.5429175503371793,
           'mae_age2': 1.2323312869741203,
           'rmse_age3': 1.5133436380681657,
           'mae_age3': 1.2160405292617755,
           'rmse_age4': 1.4746683850115507,
           'mae_age4': 1.1833312104514728,
           'rmse_age5': 1.4802993563204714,
           'mae_age5': 1.1765965716859306},
          'KNNBaseline': {'rmse': 0.9345585443837379,
           'mae': 0.7326249556089116,
           'rmse_age1': 0.9690639778757134,
           'mae_age1': 0.7698317641612282,
           'rmse_age2': 0.9507951991401281,
           'mae_age2': 0.7433715549626865,
           'rmse_age3': 0.9225425581871517,
           'mae_age3': 0.7217084076203123,
           'rmse_age4': 0.9274739394155848,
           'mae_age4': 0.7321116196486097,
           'rmse_age5': 0.8919891745725138,
           'mae_age5': 0.6976652170340305},
          'KNNBasic': {'rmse': 0.9847974058490248,
           'mae': 0.7750209854439283,
           'rmse_age1': 1.0084494515532185,
           'mae_age1': 0.799695954022524,
           'rmse_age2': 1.0276093046031178,
           'mae_age2': 0.8112788516872514,
           'rmse_age3': 0.9531975370891087,
           'mae_age3': 0.7473333938627962,
           'rmse_age4': 0.9516746048894967,
           'mae_age4': 0.7497716211342738,
           'rmse_age5': 0.9313345356325581,
           'mae_age5': 0.7306263920581824},
          'KNNWithMeans': {'rmse': 0.9570797740894271,
           'mae': 0.7510259453640877,
           'rmse_age1': 0.9892688057173274,
           'mae_age1': 0.7958561843202641,
                    'rmse_age2': 0.9771096127818242,
                    'mae_age2': 0.7647555725073129,
                    'rmse_age3': 0.946139992737961,
                    'mae_age3': 0.7416708938376047,
                    'rmse_age4': 0.9453523720674696,
                    'mae_age4': 0.7450007754536555,
                    'rmse_age5': 0.9068078035807255,
                    'mae_age5': 0.7052338986491661},
                   'KNNWithZScore': {'rmse': 0.9575974988585253,
                    'mae': 0.7480474779974006,
                    'rmse_age1': 0.9844335060105504,
                    'mae_age1': 0.7834301557230423,
                    'rmse_age2': 0.9804604683798683,
                    'mae_age2': 0.7633667454398667,
                    'rmse_age3': 0.9491384189767961,
                    'mae_age3': 0.7408707225875766,
                    'rmse_age4': 0.9404258296286523,
                    'mae_age4': 0.739336183383438,
                    'rmse_age5': 0.9028562768531777,
                    'mae_age5': 0.702014110154094},
                   'BaselineOnly': {'rmse': 0.9476515797376743,
                    'mae': 0.7493986092441747,
                    'rmse_age1': 0.9929042741918218,
                    'mae_age1': 0.7935742418782911,
                    'rmse_age2': 0.9623980904712194,
                    'mae_age2': 0.7616398214379686,
                    'rmse_age3': 0.9392968118089815,
                    'mae_age3': 0.7403927922096207,
                    'rmse_age4': 0.9442189762662868,
                    'mae_age4': 0.7480783872385266,
                    'rmse_age5': 0.890379546425079,
                    'mae_age5': 0.7023958764925953},
                   'CoClustering': {'rmse': 0.9672222174230348,
                    'mae': 0.7524542122319642,
                    'rmse_age1': 0.9959324188219706,
                    'mae_age1': 0.7916319547833477,
                    'rmse_age2': 0.9855851437861507,
                    'mae_age2': 0.7652175908018869,
                    'rmse_age3': 0.9583879332814349,
                    'mae_age3': 0.7460678414400376,
                    'rmse_age4': 0.9553165203021982,
                    'mae_age4': 0.7448907253438006,
                    'rmse_age5': 0.9208340933095958,
                    'mae_age5': 0.7091507532623104}}
```

u3

In [36]:

```
# load
df_train = pd.read_csv("data/ml-100k_merged/u3.base")
df_test = pd.read_csv("data/ml-100k_merged/u3.test")
data_train, data_test, data_test_age_list = build_train_test_age(df_train, df_test)
all_results3 = {}
save_model = False
for algorithm_name in algorithms.keys():
    result = train_single_algorithm_age(algorithm_name, data_train, data_test, data_test_age_list, save_model
)
    all_results3[algorithm_name] = result
```

```
RMSE: 0.9349
MAE:  0.7362
RMSE: 1.0152
MAE:  0.8070
RMSE: 0.9354
MAE:  0.7358
RMSE: 0.9108
MAE:  0.7240
RMSE: 0.9528
MAE:  0.7457
RMSE: 0.8988
MAE:  0.6993
===== ===== ===== ===== =====
SVD             |0.07 mins|rmse: 0.9349|mae: 0.7362
AgeGroup      1|rmse: 1.0152|mae: 0.8070
AgeGroup      2|rmse: 0.9354|mae: 0.7358
AgeGroup      3|rmse: 0.9108|mae: 0.7240
AgeGroup      4|rmse: 0.9528|mae: 0.7457
AgeGroup      5|rmse: 0.8988|mae: 0.6993
===== ===== ===== ===== =====
RMSE: 0.9186
MAE:  0.7206
RMSE: 0.9811
MAE:  0.7688
RMSE: 0.9191
MAE:  0.7198
RMSE: 0.8961
MAE:  0.7081
RMSE: 0.9421
MAE:  0.7362
RMSE: 0.8859
MAE:  0.6944
===== ===== ===== ===== =====
SVDpp           |2.43 mins|rmse: 0.9186|mae: 0.7206
AgeGroup      1|rmse: 0.9811|mae: 0.7688
AgeGroup      2|rmse: 0.9191|mae: 0.7198
AgeGroup      3|rmse: 0.8961|mae: 0.7081
AgeGroup      4|rmse: 0.9421|mae: 0.7362
AgeGroup      5|rmse: 0.8859|mae: 0.6944
===== ===== ===== ===== =====
RMSE: 0.9457
MAE:  0.7427
RMSE: 1.0171
MAE:  0.8032
RMSE: 0.9448
MAE:  0.7410
RMSE: 0.9227
MAE:  0.7289
RMSE: 0.9718
MAE:  0.7611
RMSE: 0.9086
MAE:  0.7098
===== ===== ===== ===== =====
SlopeOne        |0.06 mins|rmse: 0.9457|mae: 0.7427
AgeGroup      1|rmse: 1.0171|mae: 0.8032
AgeGroup      2|rmse: 0.9448|mae: 0.7410
AgeGroup      3|rmse: 0.9227|mae: 0.7289
AgeGroup      4|rmse: 0.9718|mae: 0.7611
AgeGroup      5|rmse: 0.9086|mae: 0.7098
===== ===== ===== ===== =====
RMSE: 0.9577
MAE:  0.7529
RMSE: 1.0264
MAE:  0.8092
RMSE: 0.9576
MAE:  0.7528
RMSE: 0.9366
MAE:  0.7405
RMSE: 0.9795
MAE:  0.7659
RMSE: 0.9216
MAE:  0.7218
===== ===== ===== ===== =====
NMF             |0.06 mins|rmse: 0.9577|mae: 0.7529
AgeGroup      1|rmse: 1.0264|mae: 0.8092
AgeGroup      2|rmse: 0.9576|mae: 0.7528
AgeGroup      3|rmse: 0.9366|mae: 0.7405
AgeGroup      4|rmse: 0.9795|mae: 0.7659
AgeGroup      5|rmse: 0.9216|mae: 0.7218
===== ===== ===== ===== =====
RMSE: 1.5019
MAE:  1.2049
RMSE: 1.5582
MAE:  1.2513
RMSE: 1.5262
MAE:  1.2309
RMSE: 1.5044
MAE:  1.2025
```

```
RMSE: 1.4641
MAE:  1.1711
RMSE: 1.4859
MAE:  1.1948
===== ===== ===== ===== =====
NormalPredictor|0.01 mins|rmse: 1.5019|mae: 1.2049
AgeGroup      1|rmse: 1.5582|mae: 1.2513
AgeGroup      2|rmse: 1.5262|mae: 1.2309
AgeGroup      3|rmse: 1.5044|mae: 1.2025
AgeGroup      4|rmse: 1.4641|mae: 1.1711
AgeGroup      5|rmse: 1.4859|mae: 1.1948
===== ===== ===== ===== =====
Estimating biases using als...
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9292
MAE:  0.7317
RMSE: 1.0070
MAE:  0.7957
RMSE: 0.9286
MAE:  0.7323
RMSE: 0.9076
MAE:  0.7193
RMSE: 0.9483
MAE:  0.7433
RMSE: 0.8915
MAE:  0.6944
===== ===== ===== ===== =====
KNNBaseline    |0.10 mins|rmse: 0.9292|mae: 0.7317
AgeGroup      1|rmse: 1.0070|mae: 0.7957
AgeGroup      2|rmse: 0.9286|mae: 0.7323
AgeGroup      3|rmse: 0.9076|mae: 0.7193
AgeGroup      4|rmse: 0.9483|mae: 0.7433
AgeGroup      5|rmse: 0.8915|mae: 0.6944
===== ===== ===== ===== =====
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9771
MAE:  0.7709
RMSE: 1.0323
MAE:  0.8249
RMSE: 0.9877
MAE:  0.7795
RMSE: 0.9573
MAE:  0.7579
RMSE: 0.9789
MAE:  0.7645
RMSE: 0.9395
MAE:  0.7392
===== ===== ===== ===== =====
KNNBasic       |0.09 mins|rmse: 0.9771|mae: 0.7709
AgeGroup      1|rmse: 1.0323|mae: 0.8249
AgeGroup      2|rmse: 0.9877|mae: 0.7795
AgeGroup      3|rmse: 0.9573|mae: 0.7579
AgeGroup      4|rmse: 0.9789|mae: 0.7645
AgeGroup      5|rmse: 0.9395|mae: 0.7392
===== ===== ===== ===== =====
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9499
MAE:  0.7473
RMSE: 1.0347
MAE:  0.8221
RMSE: 0.9506
MAE:  0.7518
RMSE: 0.9255
MAE:  0.7323
RMSE: 0.9690
MAE:  0.7557
RMSE: 0.9079
MAE:  0.6999
===== ===== ===== ===== =====
KNNWithMeans   |0.09 mins|rmse: 0.9499|mae: 0.7473
AgeGroup      1|rmse: 1.0347|mae: 0.8221
AgeGroup      2|rmse: 0.9506|mae: 0.7518
AgeGroup      3|rmse: 0.9255|mae: 0.7323
AgeGroup      4|rmse: 0.9690|mae: 0.7557
AgeGroup      5|rmse: 0.9079|mae: 0.6999
===== ===== ===== ===== =====
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9501
MAE:  0.7444
RMSE: 1.0398
MAE:  0.8210
RMSE: 0.9504
MAE:  0.7496
RMSE: 0.9252
MAE:  0.7285
```

```
RMSE: 0.9721
MAE:  0.7543
RMSE: 0.9030
MAE:  0.6936
===== ===== ===== ===== =====
KNNWithZScore  |0.10 mins|rmse: 0.9501|mae: 0.7444
AgeGroup      1|rmse: 1.0398|mae: 0.8210
AgeGroup      2|rmse: 0.9504|mae: 0.7496
AgeGroup      3|rmse: 0.9252|mae: 0.7285
AgeGroup      4|rmse: 0.9721|mae: 0.7543
AgeGroup      5|rmse: 0.9030|mae: 0.6936
===== ===== ===== ===== =====
Estimating biases using als...
RMSE: 0.9405
MAE:  0.7445
RMSE: 1.0259
MAE:  0.8193
RMSE: 0.9400
MAE:  0.7447
RMSE: 0.9173
MAE:  0.7321
RMSE: 0.9642
MAE:  0.7587
RMSE: 0.8937
MAE:  0.6980
===== ===== ===== ===== =====
BaselineOnly   |0.01 mins|rmse: 0.9405|mae: 0.7445
AgeGroup      1|rmse: 1.0259|mae: 0.8193
AgeGroup      2|rmse: 0.9400|mae: 0.7447
AgeGroup      3|rmse: 0.9173|mae: 0.7321
AgeGroup      4|rmse: 0.9642|mae: 0.7587
AgeGroup      5|rmse: 0.8937|mae: 0.6980
===== ===== ===== ===== =====
RMSE: 0.9763
MAE:  0.7641
RMSE: 1.0548
MAE:  0.8376
RMSE: 0.9740
MAE:  0.7653
RMSE: 0.9506
MAE:  0.7450
RMSE: 1.0034
MAE:  0.7791
RMSE: 0.9418
MAE:  0.7277
===== ===== ===== ===== =====
CoClustering   |0.02 mins|rmse: 0.9763|mae: 0.7641
AgeGroup      1|rmse: 1.0548|mae: 0.8376
AgeGroup      2|rmse: 0.9740|mae: 0.7653
AgeGroup      3|rmse: 0.9506|mae: 0.7450
AgeGroup      4|rmse: 1.0034|mae: 0.7791
AgeGroup      5|rmse: 0.9418|mae: 0.7277
===== ===== ===== ===== =====
```

In [37]:

```
all_results3
```

```
Out[37]: {'SVD': {'rmse': 0.9349360824691094,
           'mae': 0.7361570109990281,
           'rmse_age1': 1.0152262470933948,
           'mae_age1': 0.8069678981278149,
           'rmse_age2': 0.9353614655891929,
           'mae_age2': 0.7357968305117557,
           'rmse_age3': 0.910792863733666,
           'mae_age3': 0.7240417872710035,
           'rmse_age4': 0.9528008695402274,
           'mae_age4': 0.7456529455795696,
           'rmse_age5': 0.8987772110931995,
           'mae_age5': 0.6992744537364306},
          'SVDpp': {'rmse': 0.9185792921809792,
           'mae': 0.7205642847007804,
           'rmse_age1': 0.9810853799509386,
           'mae_age1': 0.7687974334210007,
           'rmse_age2': 0.9190557456871318,
           'mae_age2': 0.7197926477395016,
           'rmse_age3': 0.8960850283183477,
           'mae_age3': 0.7081269409376556,
           'rmse_age4': 0.9421131544093174,
           'mae_age4': 0.7361720913771379,
           'rmse_age5': 0.8858561141321994,
           'mae_age5': 0.6944149533044772},
          'SlopeOne': {'rmse': 0.9457210548243965,
           'mae': 0.7426791889651031,
           'rmse_age1': 1.017121227981564,
           'mae_age1': 0.8031827234412947,
           'rmse_age2': 0.9447518937079595,
           'mae_age2': 0.740999864059696,
           'rmse_age3': 0.9226659645745868,
           'mae_age3': 0.7289151179330406,
           'rmse_age4': 0.9717517993994954,
           'mae_age4': 0.7610809911723609,
           'rmse_age5': 0.9085731460817501,
           'mae_age5': 0.7098266721456832},
          'NMF': {'rmse': 0.9577301506923016,
           'mae': 0.7528819883867695,
           'rmse_age1': 1.026394214208863,
           'mae_age1': 0.8092340993038045,
           'rmse_age2': 0.9576286061954477,
           'mae_age2': 0.75280768779979,
           'rmse_age3': 0.9366302844988406,
           'mae_age3': 0.740467242739887,
           'rmse_age4': 0.9794641370389955,
           'mae_age4': 0.7659052571375615,
           'rmse_age5': 0.9215784500759234,
           'mae_age5': 0.7218243511803312},
          'NormalPredictor': {'rmse': 1.5018835287800265,
           'mae': 1.204908480045924,
           'rmse_age1': 1.558244433131279,
           'mae_age1': 1.2512531156336957,
           'rmse_age2': 1.5262323168643,
           'mae_age2': 1.230904788457827,
           'rmse_age3': 1.5043674219200123,
           'mae_age3': 1.2024764262444763,
           'rmse_age4': 1.4641318671243169,
           'mae_age4': 1.171128439261796,
           'rmse_age5': 1.485872421737157,
           'mae_age5': 1.1948225169959787},
          'KNNBaseline': {'rmse': 0.9291944905970315,
           'mae': 0.7316578581069132,
           'rmse_age1': 1.0070162648138137,
           'mae_age1': 0.7956928340990657,
           'rmse_age2': 0.9286401744886168,
           'mae_age2': 0.73230845087309,
           'rmse_age3': 0.9075876260742347,
           'mae_age3': 0.7193005309511112,
           'rmse_age4': 0.948260612011482,
           'mae_age4': 0.7433083513896773,
           'rmse_age5': 0.8914796396877934,
           'mae_age5': 0.6943881042193254},
          'KNNBasic': {'rmse': 0.97709795253605,
           'mae': 0.7708818274785253,
           'rmse_age1': 1.0323368334971463,
           'mae_age1': 0.8249129637691067,
           'rmse_age2': 0.9877289051839043,
           'mae_age2': 0.7794528486595249,
           'rmse_age3': 0.9573443243287554,
           'mae_age3': 0.7579290063823599,
           'rmse_age4': 0.9789048300521922,
           'mae_age4': 0.7644715555848683,
           'rmse_age5': 0.939459120467521,
           'mae_age5': 0.7392277074111637},
          'KNNWithMeans': {'rmse': 0.9498546994004081,
           'mae': 0.7473468557726317,
           'rmse_age1': 1.0346964932890503,
           'mae_age1': 0.8220924196094083,
  'rmse_age2': 0.9505981527646352,
  'mae_age2': 0.7518417076656633,
  'rmse_age3': 0.925452864584297,
  'mae_age3': 0.7322547040653032,
  'rmse_age4': 0.9689914414486058,
  'mae_age4': 0.7557348877795373,
  'rmse_age5': 0.907928966508694,
  'mae_age5': 0.6998949568047236},
 'KNNWithZScore': {'rmse': 0.9501250111896186,
  'mae': 0.7444131337643888,
  'rmse_age1': 1.0398022951038484,
  'mae_age1': 0.8209974704823247,
  'rmse_age2': 0.9504230986004776,
  'mae_age2': 0.7495712555941925,
  'rmse_age3': 0.9252407551835149,
  'mae_age3': 0.7284587619260424,
  'rmse_age4': 0.9721281401516518,
  'mae_age4': 0.7542534102624522,
  'rmse_age5': 0.902957276143092,
  'mae_age5': 0.6935610163685376},
 'BaselineOnly': {'rmse': 0.9405230282786979,
  'mae': 0.7445158474930392,
  'rmse_age1': 1.0259465713636362,
  'mae_age1': 0.8192668752864052,
  'rmse_age2': 0.9400369010818292,
  'mae_age2': 0.7446874384742277,
  'rmse_age3': 0.9173008792481192,
  'mae_age3': 0.7321471446707648,
  'rmse_age4': 0.9642419564880065,
  'mae_age4': 0.7587014427700733,
  'rmse_age5': 0.8937254141641122,
  'mae_age5': 0.6980121888103226},
 'CoClustering': {'rmse': 0.9763043948300641,
  'mae': 0.7640989354958094,
  'rmse_age1': 1.0547937160774925,
  'mae_age1': 0.8376481457341626,
  'rmse_age2': 0.9740190062191303,
  'mae_age2': 0.7652937763876154,
  'rmse_age3': 0.9506003741766978,
  'mae_age3': 0.7450411018522141,
  'rmse_age4': 1.0033920870525128,
  'mae_age4': 0.7790852689123556,
  'rmse_age5': 0.9418199091932368,
  'mae_age5': 0.7276949774667439}}
```

u4

In [38]:
```
# load
df_train = pd.read_csv("data/ml-100k_merged/u4.base")
df_test = pd.read_csv("data/ml-100k_merged/u4.test")
data_train, data_test, data_test_age_list = build_train_test_age(df_train, df_test)
all_results4 = {}
save_model = False
for algorithm_name in algorithms.keys():
    result = train_single_algorithm_age(algorithm_name, data_train, data_test, data_test_age_list, save_model
)
    all_results4[algorithm_name] = result
```

```
RMSE: 0.9327
MAE:  0.7349
RMSE: 0.9889
MAE:  0.7776
RMSE: 0.9328
MAE:  0.7314
RMSE: 0.9390
MAE:  0.7449
RMSE: 0.9259
MAE:  0.7300
RMSE: 0.8772
MAE:  0.6962
===== ===== ===== ===== =====
SVD            |0.07 mins|rmse: 0.9327|mae: 0.7349
AgeGroup      1|rmse: 0.9889|mae: 0.7776
AgeGroup      2|rmse: 0.9328|mae: 0.7314
AgeGroup      3|rmse: 0.9390|mae: 0.7449
AgeGroup      4|rmse: 0.9259|mae: 0.7300
AgeGroup      5|rmse: 0.8772|mae: 0.6962
===== ===== ===== ===== =====
RMSE: 0.9165
MAE:  0.7185
RMSE: 0.9668
MAE:  0.7533
RMSE: 0.9144
MAE:  0.7147
RMSE: 0.9237
MAE:  0.7295
RMSE: 0.9113
MAE:  0.7111
RMSE: 0.8704
MAE:  0.6883
===== ===== ===== ===== =====
SVDpp          |2.46 mins|rmse: 0.9165|mae: 0.7185
AgeGroup      1|rmse: 0.9668|mae: 0.7533
AgeGroup      2|rmse: 0.9144|mae: 0.7147
AgeGroup      3|rmse: 0.9237|mae: 0.7295
AgeGroup      4|rmse: 0.9113|mae: 0.7111
AgeGroup      5|rmse: 0.8704|mae: 0.6883
===== ===== ===== ===== =====
RMSE: 0.9432
MAE:  0.7402
RMSE: 0.9944
MAE:  0.7764
RMSE: 0.9428
MAE:  0.7369
RMSE: 0.9516
MAE:  0.7540
RMSE: 0.9394
MAE:  0.7360
RMSE: 0.8836
MAE:  0.6956
===== ===== ===== ===== =====
SlopeOne       |0.06 mins|rmse: 0.9432|mae: 0.7402
AgeGroup      1|rmse: 0.9944|mae: 0.7764
AgeGroup      2|rmse: 0.9428|mae: 0.7369
AgeGroup      3|rmse: 0.9516|mae: 0.7540
AgeGroup      4|rmse: 0.9394|mae: 0.7360
AgeGroup      5|rmse: 0.8836|mae: 0.6956
===== ===== ===== ===== =====
RMSE: 0.9605
MAE:  0.7546
RMSE: 1.0311
MAE:  0.8048
RMSE: 0.9614
MAE:  0.7522
RMSE: 0.9660
MAE:  0.7639
RMSE: 0.9483
MAE:  0.7477
RMSE: 0.8992
MAE:  0.7094
===== ===== ===== ===== =====
NMF            |0.07 mins|rmse: 0.9605|mae: 0.7546
AgeGroup      1|rmse: 1.0311|mae: 0.8048
AgeGroup      2|rmse: 0.9614|mae: 0.7522
AgeGroup      3|rmse: 0.9660|mae: 0.7639
AgeGroup      4|rmse: 0.9483|mae: 0.7477
AgeGroup      5|rmse: 0.8992|mae: 0.7094
===== ===== ===== ===== =====
RMSE: 1.5145
MAE:  1.2172
RMSE: 1.5565
MAE:  1.2498
RMSE: 1.5188
MAE:  1.2213
RMSE: 1.5078
MAE:  1.2086
```

```
RMSE: 1.4791
MAE:  1.1908
RMSE: 1.4481
MAE:  1.1651
===== ===== ===== ===== =====
NormalPredictor|0.01 mins|rmse: 1.5145|mae: 1.2172
AgeGroup      1|rmse: 1.5565|mae: 1.2498
AgeGroup      2|rmse: 1.5188|mae: 1.2213
AgeGroup      3|rmse: 1.5078|mae: 1.2086
AgeGroup      4|rmse: 1.4791|mae: 1.1908
AgeGroup      5|rmse: 1.4481|mae: 1.1651
===== ===== ===== ===== =====
Estimating biases using als...
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9260
MAE:  0.7301
RMSE: 0.9784
MAE:  0.7742
RMSE: 0.9276
MAE:  0.7272
RMSE: 0.9321
MAE:  0.7423
RMSE: 0.9164
MAE:  0.7189
RMSE: 0.8726
MAE:  0.6910
===== ===== ===== ===== =====
KNNBaseline    |0.11 mins|rmse: 0.9260|mae: 0.7301
AgeGroup      1|rmse: 0.9784|mae: 0.7742
AgeGroup      2|rmse: 0.9276|mae: 0.7272
AgeGroup      3|rmse: 0.9321|mae: 0.7423
AgeGroup      4|rmse: 0.9164|mae: 0.7189
AgeGroup      5|rmse: 0.8726|mae: 0.6910
===== ===== ===== ===== =====
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9704
MAE:  0.7671
RMSE: 1.0100
MAE:  0.8073
RMSE: 0.9810
MAE:  0.7705
RMSE: 0.9736
MAE:  0.7769
RMSE: 0.9429
MAE:  0.7396
RMSE: 0.9260
MAE:  0.7364
===== ===== ===== ===== =====
KNNBasic       |0.09 mins|rmse: 0.9704|mae: 0.7671
AgeGroup      1|rmse: 1.0100|mae: 0.8073
AgeGroup      2|rmse: 0.9810|mae: 0.7705
AgeGroup      3|rmse: 0.9736|mae: 0.7769
AgeGroup      4|rmse: 0.9429|mae: 0.7396
AgeGroup      5|rmse: 0.9260|mae: 0.7364
===== ===== ===== ===== =====
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9448
MAE:  0.7444
RMSE: 1.0022
MAE:  0.7999
RMSE: 0.9457
MAE:  0.7428
RMSE: 0.9536
MAE:  0.7560
RMSE: 0.9320
MAE:  0.7298
RMSE: 0.8875
MAE:  0.6978
===== ===== ===== ===== =====
KNNWithMeans   |0.10 mins|rmse: 0.9448|mae: 0.7444
AgeGroup      1|rmse: 1.0022|mae: 0.7999
AgeGroup      2|rmse: 0.9457|mae: 0.7428
AgeGroup      3|rmse: 0.9536|mae: 0.7560
AgeGroup      4|rmse: 0.9320|mae: 0.7298
AgeGroup      5|rmse: 0.8875|mae: 0.6978
===== ===== ===== ===== =====
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9452
MAE:  0.7413
RMSE: 1.0035
MAE:  0.7918
RMSE: 0.9475
MAE:  0.7423
RMSE: 0.9540
MAE:  0.7531
```

```
RMSE: 0.9312
MAE:  0.7247
RMSE: 0.8828
MAE:  0.6901
===== ===== ===== ===== =====
KNNWithZScore  |0.10 mins|rmse: 0.9452|mae: 0.7413
AgeGroup      1|rmse: 1.0035|mae: 0.7918
AgeGroup      2|rmse: 0.9475|mae: 0.7423
AgeGroup      3|rmse: 0.9540|mae: 0.7531
AgeGroup      4|rmse: 0.9312|mae: 0.7247
AgeGroup      5|rmse: 0.8828|mae: 0.6901
===== ===== ===== ===== =====
Estimating biases using als...
RMSE: 0.9383
MAE:  0.7442
RMSE: 1.0042
MAE:  0.7956
RMSE: 0.9406
MAE:  0.7440
RMSE: 0.9455
MAE:  0.7574
RMSE: 0.9275
MAE:  0.7324
RMSE: 0.8682
MAE:  0.6873
===== ===== ===== ===== =====
BaselineOnly   |0.00 mins|rmse: 0.9383|mae: 0.7442
AgeGroup      1|rmse: 1.0042|mae: 0.7956
AgeGroup      2|rmse: 0.9406|mae: 0.7440
AgeGroup      3|rmse: 0.9455|mae: 0.7574
AgeGroup      4|rmse: 0.9275|mae: 0.7324
AgeGroup      5|rmse: 0.8682|mae: 0.6873
===== ===== ===== ===== =====
RMSE: 0.9614
MAE:  0.7510
RMSE: 1.0128
MAE:  0.7975
RMSE: 0.9588
MAE:  0.7454
RMSE: 0.9711
MAE:  0.7633
RMSE: 0.9522
MAE:  0.7424
RMSE: 0.9162
MAE:  0.7162
===== ===== ===== ===== =====
CoClustering   |0.02 mins|rmse: 0.9614|mae: 0.7510
AgeGroup      1|rmse: 1.0128|mae: 0.7975
AgeGroup      2|rmse: 0.9588|mae: 0.7454
AgeGroup      3|rmse: 0.9711|mae: 0.7633
AgeGroup      4|rmse: 0.9522|mae: 0.7424
AgeGroup      5|rmse: 0.9162|mae: 0.7162
===== ===== ===== ===== =====
In [39]: all_results4
```

```
Out[39]: {'SVD': {'rmse': 0.9326901866357193,
           'mae': 0.7349211664993027,
           'rmse_age1': 0.9888506432525179,
           'mae_age1': 0.7775561042289055,
           'rmse_age2': 0.9328208957751281,
           'mae_age2': 0.7313549661425284,
           'rmse_age3': 0.9389531914832907,
           'mae_age3': 0.7448838799866787,
           'rmse_age4': 0.9258673401373062,
           'mae_age4': 0.7300243946492211,
           'rmse_age5': 0.8771732202306594,
           'mae_age5': 0.6962207022892283},
          'SVDpp': {'rmse': 0.9165000127174876,
           'mae': 0.7185304443215897,
           'rmse_age1': 0.9668221824459928,
           'mae_age1': 0.7533479234043579,
           'rmse_age2': 0.9144422360714523,
           'mae_age2': 0.714741687877907,
           'rmse_age3': 0.923715978507645,
           'mae_age3': 0.7294595460099375,
           'rmse_age4': 0.9113084742365055,
           'mae_age4': 0.7111366926520004,
           'rmse_age5': 0.8703756318387476,
           'mae_age5': 0.6883272123028583},
          'SlopeOne': {'rmse': 0.9431506535440269,
           'mae': 0.7402308642861269,
           'rmse_age1': 0.9944260115329252,
           'mae_age1': 0.7764412953392191,
           'rmse_age2': 0.942821722403938,
           'mae_age2': 0.7368810308729963,
           'rmse_age3': 0.9516418234296757,
           'mae_age3': 0.754005522080426,
           'rmse_age4': 0.9393535227064899,
           'mae_age4': 0.7360440675637884,
           'rmse_age5': 0.883632636379239,
           'mae_age5': 0.6955751367796851},
          'NMF': {'rmse': 0.9605404582864492,
           'mae': 0.7545780236790011,
           'rmse_age1': 1.0310540073123784,
           'mae_age1': 0.8047703406219595,
           'rmse_age2': 0.9613515096548126,
           'mae_age2': 0.7522433406873511,
           'rmse_age3': 0.9659748242329016,
           'mae_age3': 0.7638696058598734,
           'rmse_age4': 0.9482978301929846,
           'mae_age4': 0.7476898789266331,
           'rmse_age5': 0.8992483723320222,
           'mae_age5': 0.7093570973812423},
          'NormalPredictor': {'rmse': 1.5144820564898698,
           'mae': 1.217225052688765,
           'rmse_age1': 1.5565026537950086,
           'mae_age1': 1.2497552982575186,
           'rmse_age2': 1.518830602881918,
           'mae_age2': 1.2213291105339796,
           'rmse_age3': 1.5077935394340332,
           'mae_age3': 1.2086075088522334,
           'rmse_age4': 1.4791098131490847,
           'mae_age4': 1.1908021752061166,
           'rmse_age5': 1.448052293643547,
           'mae_age5': 1.1651448909137603},
          'KNNBaseline': {'rmse': 0.9260155325581058,
           'mae': 0.7300987706659998,
           'rmse_age1': 0.9783960318742752,
           'mae_age1': 0.7741857608416494,
           'rmse_age2': 0.9275970916453422,
           'mae_age2': 0.7272080289142238,
           'rmse_age3': 0.9321301464739279,
           'mae_age3': 0.7423031170498507,
           'rmse_age4': 0.9163738550144453,
           'mae_age4': 0.7189428134417627,
           'rmse_age5': 0.8725963678371004,
           'mae_age5': 0.6910097852521457},
          'KNNBasic': {'rmse': 0.9704489972276842,
           'mae': 0.7671418382414986,
           'rmse_age1': 1.0099839771990506,
           'mae_age1': 0.8072606742534543,
           'rmse_age2': 0.9809816146480498,
           'mae_age2': 0.7704922029659271,
           'rmse_age3': 0.973602397863352,
           'mae_age3': 0.7769177381401484,
           'rmse_age4': 0.9429115184998004,
           'mae_age4': 0.7395816448869137,
           'rmse_age5': 0.9260124381710026,
           'mae_age5': 0.7364379116084234},
          'KNNWithMeans': {'rmse': 0.9448167988667769,
           'mae': 0.7444309286996698,
           'rmse_age1': 1.0022433822022874,
           'mae_age1': 0.799878077688803 7,
                    'rmse_age2': 0.9457146435899199,
                    'mae_age2': 0.7427840223891808,
                    'rmse_age3': 0.9535878851029961,
                    'mae_age3': 0.7559572133990116,
                    'rmse_age4': 0.9319620062107408,
                    'mae_age4': 0.7297900944757262,
                    'rmse_age5': 0.887452269184076,
                    'mae_age5': 0.6977981036222801},
                   'KNNWithZScore': {'rmse': 0.9452156164912281,
                    'mae': 0.7412841328617357,
                    'rmse_age1': 1.0035379058378973,
                    'mae_age1': 0.7918493514018637,
                    'rmse_age2': 0.9475198922096211,
                    'mae_age2': 0.7423245712131903,
                    'rmse_age3': 0.9540096716243703,
                    'mae_age3': 0.7531416295282003,
                    'rmse_age4': 0.9312263456036458,
                    'mae_age4': 0.7247454982672915,
                    'rmse_age5': 0.8828000124731217,
                    'mae_age5': 0.6900910956723189},
                   'BaselineOnly': {'rmse': 0.938284026686687,
                    'mae': 0.7442326440918581,
                    'rmse_age1': 1.004220413683713,
                    'mae_age1': 0.7955826450174796,
                    'rmse_age2': 0.9405671948547226,
                    'mae_age2': 0.744033522114975,
                    'rmse_age3': 0.9454814742357062,
                    'mae_age3': 0.7573970096870294,
                    'rmse_age4': 0.9274788918961356,
                    'mae_age4': 0.7323911041686982,
                    'rmse_age5': 0.8681632118255633,
                    'mae_age5': 0.6872877350888784},
                   'CoClustering': {'rmse': 0.9614161677899277,
                    'mae': 0.7509515727576168,
                    'rmse_age1': 1.0128102185104795,
                    'mae_age1': 0.7975167771095403,
                    'rmse_age2': 0.958803993680256,
                    'mae_age2': 0.7453531220299355,
                    'rmse_age3': 0.9711140813889769,
                    'mae_age3': 0.7633116865028281,
                    'rmse_age4': 0.9521918567269994,
                    'mae_age4': 0.7424428744644536,
                    'rmse_age5': 0.9162421337992028,
                    'mae_age5': 0.7162092729715555}}
```

u5

In [40]:

```
# load
df_train = pd.read_csv("data/ml-100k_merged/u5.base")
df_test = pd.read_csv("data/ml-100k_merged/u5.test")
data_train, data_test, data_test_age_list = build_train_test_age(df_train, df_test)
all_results5 = {}
save_model = False
for algorithm_name in algorithms.keys():
    result = train_single_algorithm_age(algorithm_name, data_train, data_test, data_test_age_list, save_model
)
    all_results5[algorithm_name] = result
```

```
RMSE: 0.9303
MAE:  0.7369
RMSE: 0.9967
MAE:  0.7873
RMSE: 0.9193
MAE:  0.7354
RMSE: 0.9346
MAE:  0.7358
RMSE: 0.9271
MAE:  0.7270
RMSE: 0.9110
MAE:  0.7179
===== ===== ===== ===== =====
SVD             |0.07 mins|rmse: 0.9303|mae: 0.7369
AgeGroup       1|rmse: 0.9967|mae: 0.7873
AgeGroup       2|rmse: 0.9193|mae: 0.7354
AgeGroup       3|rmse: 0.9346|mae: 0.7358
AgeGroup       4|rmse: 0.9271|mae: 0.7270
AgeGroup       5|rmse: 0.9110|mae: 0.7179
===== ===== ===== ===== =====
RMSE: 0.9169
MAE:  0.7228
RMSE: 0.9823
MAE:  0.7679
RMSE: 0.9082
MAE:  0.7204
RMSE: 0.9186
MAE:  0.7223
RMSE: 0.9115
MAE:  0.7148
RMSE: 0.8987
MAE:  0.7078
===== ===== ===== ===== =====
SVDpp           |2.65 mins|rmse: 0.9169|mae: 0.7228
AgeGroup       1|rmse: 0.9823|mae: 0.7679
AgeGroup       2|rmse: 0.9082|mae: 0.7204
AgeGroup       3|rmse: 0.9186|mae: 0.7223
AgeGroup       4|rmse: 0.9115|mae: 0.7148
AgeGroup       5|rmse: 0.8987|mae: 0.7078
===== ===== ===== ===== =====
RMSE: 0.9408
MAE:  0.7436
RMSE: 1.0034
MAE:  0.7901
RMSE: 0.9299
MAE:  0.7388
RMSE: 0.9433
MAE:  0.7412
RMSE: 0.9384
MAE:  0.7412
RMSE: 0.9281
MAE:  0.7337
===== ===== ===== ===== =====
SlopeOne        |0.06 mins|rmse: 0.9408|mae: 0.7436
AgeGroup       1|rmse: 1.0034|mae: 0.7901
AgeGroup       2|rmse: 0.9299|mae: 0.7388
AgeGroup       3|rmse: 0.9433|mae: 0.7412
AgeGroup       4|rmse: 0.9384|mae: 0.7412
AgeGroup       5|rmse: 0.9281|mae: 0.7337
===== ===== ===== ===== =====
RMSE: 0.9631
MAE:  0.7628
RMSE: 1.0319
MAE:  0.8144
RMSE: 0.9548
MAE:  0.7599
RMSE: 0.9598
MAE:  0.7557
RMSE: 0.9648
MAE:  0.7629
RMSE: 0.9429
MAE:  0.7483
===== ===== ===== ===== =====
NMF             |0.06 mins|rmse: 0.9631|mae: 0.7628
AgeGroup       1|rmse: 1.0319|mae: 0.8144
AgeGroup       2|rmse: 0.9548|mae: 0.7599
AgeGroup       3|rmse: 0.9598|mae: 0.7557
AgeGroup       4|rmse: 0.9648|mae: 0.7629
AgeGroup       5|rmse: 0.9429|mae: 0.7483
===== ===== ===== ===== =====
RMSE: 1.5148
MAE:  1.2133
RMSE: 1.5536
MAE:  1.2491
RMSE: 1.5225
MAE:  1.2296
RMSE: 1.5276
MAE:  1.2245
```

```
RMSE: 1.4851
MAE:  1.1954
RMSE: 1.4853
MAE:  1.1875
===== ===== ===== ===== =====
NormalPredictor|0.01 mins|rmse: 1.5148|mae: 1.2133
AgeGroup      1|rmse: 1.5536|mae: 1.2491
AgeGroup      2|rmse: 1.5225|mae: 1.2296
AgeGroup      3|rmse: 1.5276|mae: 1.2245
AgeGroup      4|rmse: 1.4851|mae: 1.1954
AgeGroup      5|rmse: 1.4853|mae: 1.1875
===== ===== ===== ===== =====
Estimating biases using als...
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9299
MAE:  0.7370
RMSE: 0.9966
MAE:  0.7900
RMSE: 0.9215
MAE:  0.7373
RMSE: 0.9247
MAE:  0.7280
RMSE: 0.9273
MAE:  0.7306
RMSE: 0.9225
MAE:  0.7226
===== ===== ===== ===== =====
KNNBaseline    |0.11 mins|rmse: 0.9299|mae: 0.7370
AgeGroup      1|rmse: 0.9966|mae: 0.7900
AgeGroup      2|rmse: 0.9215|mae: 0.7373
AgeGroup      3|rmse: 0.9247|mae: 0.7280
AgeGroup      4|rmse: 0.9273|mae: 0.7306
AgeGroup      5|rmse: 0.9225|mae: 0.7226
===== ===== ===== ===== =====
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9792
MAE:  0.7756
RMSE: 1.0260
MAE:  0.8188
RMSE: 0.9887
MAE:  0.7872
RMSE: 0.9649
MAE:  0.7623
RMSE: 0.9556
MAE:  0.7509
RMSE: 0.9678
MAE:  0.7588
===== ===== ===== ===== =====
KNNBasic       |0.09 mins|rmse: 0.9792|mae: 0.7756
AgeGroup      1|rmse: 1.0260|mae: 0.8188
AgeGroup      2|rmse: 0.9887|mae: 0.7872
AgeGroup      3|rmse: 0.9649|mae: 0.7623
AgeGroup      4|rmse: 0.9556|mae: 0.7509
AgeGroup      5|rmse: 0.9678|mae: 0.7588
===== ===== ===== ===== =====
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9471
MAE:  0.7505
RMSE: 1.0294
MAE:  0.8269
RMSE: 0.9380
MAE:  0.7517
RMSE: 0.9369
MAE:  0.7337
RMSE: 0.9501
MAE:  0.7460
RMSE: 0.9325
MAE:  0.7291
===== ===== ===== ===== =====
KNNWithMeans   |0.09 mins|rmse: 0.9471|mae: 0.7505
AgeGroup      1|rmse: 1.0294|mae: 0.8269
AgeGroup      2|rmse: 0.9380|mae: 0.7517
AgeGroup      3|rmse: 0.9369|mae: 0.7337
AgeGroup      4|rmse: 0.9501|mae: 0.7460
AgeGroup      5|rmse: 0.9325|mae: 0.7291
===== ===== ===== ===== =====
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9471
MAE:  0.7476
RMSE: 1.0326
MAE:  0.8191
RMSE: 0.9371
MAE:  0.7487
RMSE: 0.9367
MAE:  0.7315
```

```
RMSE: 0.9531
MAE:  0.7459
RMSE: 0.9292
MAE:  0.7257
===== ===== ===== ===== =====
KNNWithZScore  |0.10 mins|rmse: 0.9471|mae: 0.7476
AgeGroup      1|rmse: 1.0326|mae: 0.8191
AgeGroup      2|rmse: 0.9371|mae: 0.7487
AgeGroup      3|rmse: 0.9367|mae: 0.7315
AgeGroup      4|rmse: 0.9531|mae: 0.7459
AgeGroup      5|rmse: 0.9292|mae: 0.7257
===== ===== ===== ===== =====
Estimating biases using als...
RMSE: 0.9423
MAE:  0.7499
RMSE: 1.0140
MAE:  0.8058
RMSE: 0.9352
MAE:  0.7511
RMSE: 0.9437
MAE:  0.7465
RMSE: 0.9325
MAE:  0.7364
RMSE: 0.9190
MAE:  0.7259
===== ===== ===== ===== =====
BaselineOnly   |0.01 mins|rmse: 0.9423|mae: 0.7499
AgeGroup      1|rmse: 1.0140|mae: 0.8058
AgeGroup      2|rmse: 0.9352|mae: 0.7511
AgeGroup      3|rmse: 0.9437|mae: 0.7465
AgeGroup      4|rmse: 0.9325|mae: 0.7364
AgeGroup      5|rmse: 0.9190|mae: 0.7259
===== ===== ===== ===== =====
RMSE: 0.9593
MAE:  0.7545
RMSE: 1.0324
MAE:  0.8234
RMSE: 0.9474
MAE:  0.7522
RMSE: 0.9530
MAE:  0.7403
RMSE: 0.9647
MAE:  0.7542
RMSE: 0.9522
MAE:  0.7416
===== ===== ===== ===== =====
CoClustering   |0.02 mins|rmse: 0.9593|mae: 0.7545
AgeGroup      1|rmse: 1.0324|mae: 0.8234
AgeGroup      2|rmse: 0.9474|mae: 0.7522
AgeGroup      3|rmse: 0.9530|mae: 0.7403
AgeGroup      4|rmse: 0.9647|mae: 0.7542
AgeGroup      5|rmse: 0.9522|mae: 0.7416
===== ===== ===== ===== =====
```

In [41]:
```
all_results5
```

```
Out[41]: {'SVD': {'rmse': 0.9302550791697272,
           'mae': 0.7369181704895869,
           'rmse_age1': 0.9967117506442609,
           'mae_age1': 0.7873257778938304,
           'rmse_age2': 0.919283057013431,
           'mae_age2': 0.7353601512509883,
           'rmse_age3': 0.934572903825904,
           'mae_age3': 0.7357868164135016,
           'rmse_age4': 0.927123613266339,
           'mae_age4': 0.727019943385983,
           'rmse_age5': 0.9109969889258668,
           'mae_age5': 0.7179126563201715},
          'SVDpp': {'rmse': 0.9168752989688315,
           'mae': 0.7228209631412362,
           'rmse_age1': 0.982306952030302,
           'mae_age1': 0.7678777280868988,
           'rmse_age2': 0.9081873304153129,
           'mae_age2': 0.7203959019866903,
           'rmse_age3': 0.9186047118828828,
           'mae_age3': 0.7222767957063508,
           'rmse_age4': 0.911508896825789,
           'mae_age4': 0.7147644317634871,
           'rmse_age5': 0.8987060185302762,
           'mae_age5': 0.7077617117816378},
          'SlopeOne': {'rmse': 0.9407889914875407,
           'mae': 0.7436115185387002,
           'rmse_age1': 1.0034009820702283,
           'mae_age1': 0.7901380237666238,
           'rmse_age2': 0.9299045729444685,
           'mae_age2': 0.7388107412653722,
           'rmse_age3': 0.9432761361528191,
           'mae_age3': 0.7411568303051247,
           'rmse_age4': 0.9383704907780491,
           'mae_age4': 0.7412075724437422,
           'rmse_age5': 0.928105642400173,
           'mae_age5': 0.7336796184279739},
          'NMF': {'rmse': 0.9630947001669077,
           'mae': 0.7627500115908242,
           'rmse_age1': 1.0318541203442693,
           'mae_age1': 0.814394253669284,
           'rmse_age2': 0.9547803034683047,
           'mae_age2': 0.7598898901508853,
           'rmse_age3': 0.959813786963584,
           'mae_age3': 0.755688358443859,
           'rmse_age4': 0.9647879675109018,
           'mae_age4': 0.7629222180244305,
           'rmse_age5': 0.9429327279358611,
           'mae_age5': 0.7483442189101316},
          'NormalPredictor': {'rmse': 1.5147719773870745,
           'mae': 1.2133478769284152,
           'rmse_age1': 1.5535565853292819,
           'mae_age1': 1.249102456944196,
           'rmse_age2': 1.522546709174688,
           'mae_age2': 1.2296353990590538,
           'rmse_age3': 1.5276276443600652,
           'mae_age3': 1.2245449107296267,
           'rmse_age4': 1.4850986490364613,
           'mae_age4': 1.1954075018523123,
           'rmse_age5': 1.4852634018635253,
           'mae_age5': 1.1874867477505937},
          'KNNBaseline': {'rmse': 0.9299180021351924,
           'mae': 0.7369989764410888,
           'rmse_age1': 0.9965633924889542,
           'mae_age1': 0.7899877761517242,
           'rmse_age2': 0.9214966909113633,
           'mae_age2': 0.7372741695249583,
           'rmse_age3': 0.9247205347579577,
           'mae_age3': 0.7279971426387197,
           'rmse_age4': 0.9273214504183495,
           'mae_age4': 0.7305772154634026,
           'rmse_age5': 0.9225217414947833,
           'mae_age5': 0.7225777052146644},
          'KNNBasic': {'rmse': 0.9792464045505916,
           'mae': 0.7755651888896984,
           'rmse_age1': 1.0260108648127904,
           'mae_age1': 0.8188173326057968,
           'rmse_age2': 0.9886762984621237,
           'mae_age2': 0.7872158651727248,
           'rmse_age3': 0.9648832252531108,
           'mae_age3': 0.7622916619321729,
           'rmse_age4': 0.9556296875630326,
           'mae_age4': 0.7508670781901363,
           'rmse_age5': 0.9677852207800417,
           'mae_age5': 0.7588100616846855},
          'KNNWithMeans': {'rmse': 0.9470911255392472,
           'mae': 0.7505104648653849,
           'rmse_age1': 1.0294049053737357,
           'mae_age1': 0.8268985685820716,
  'rmse_age2': 0.9380203399946441,
  'mae_age2': 0.7517194824189923,
  'rmse_age3': 0.9368685058362621,
  'mae_age3': 0.7337087239330491,
  'rmse_age4': 0.9501288511596375,
  'mae_age4': 0.7460376557343746,
  'rmse_age5': 0.9325466766230293,
  'mae_age5': 0.7290805027663628},
 'KNNWithZScore': {'rmse': 0.9470581257878241,
  'mae': 0.7476369812311432,
  'rmse_age1': 1.0325941606595177,
  'mae_age1': 0.8191192917240964,
  'rmse_age2': 0.9371236545332965,
  'mae_age2': 0.7486814347721936,
  'rmse_age3': 0.9367124443447742,
  'mae_age3': 0.7314922727511709,
  'rmse_age4': 0.9530636601666047,
  'mae_age4': 0.7458609433979401,
  'rmse_age5': 0.9292202680526047,
  'mae_age5': 0.7256566897878381},
 'BaselineOnly': {'rmse': 0.9422794835917605,
  'mae': 0.7499396915080997,
  'rmse_age1': 1.0139878340197224,
  'mae_age1': 0.8058268767285596,
  'rmse_age2': 0.9351612651282349,
  'mae_age2': 0.7511211110653658,
  'rmse_age3': 0.943739551127565,
  'mae_age3': 0.7465171667207136,
  'rmse_age4': 0.9324624765595914,
  'mae_age4': 0.7363833671021532,
  'rmse_age5': 0.9189724533110473,
  'mae_age5': 0.7259444638244974},
 'CoClustering': {'rmse': 0.9593310536356667,
  'mae': 0.7545255878904495,
  'rmse_age1': 1.0323621989070229,
  'mae_age1': 0.8234005746840297,
  'rmse_age2': 0.947377015252222,
  'mae_age2': 0.7521961573700833,
  'rmse_age3': 0.9529669370671255,
  'mae_age3': 0.7402783250419818,
  'rmse_age4': 0.9647471740654063,
  'mae_age4': 0.7542289923044843,
  'rmse_age5': 0.9521627414774616,
  'mae_age5': 0.7416091946846889}}
```

现在看下在这5个数据集下的综合水平

In [42]:

```
all_results_list = [all_results, all_results2, all_results3, all_results4, all_results5]
get_mean_results(algorithms, all_results_list)
```

```
SVD            |rmse: 0.9374+-0.0068|mae: 0.7391+-0.0053
SVDpp          |rmse: 0.9221+-0.0061|mae: 0.7232+-0.0037
SlopeOne       |rmse: 0.9470+-0.0055|mae: 0.7439+-0.0035
NMF            |rmse: 0.9650+-0.0065|mae: 0.7588+-0.0049
NormalPredictor|rmse: 1.5193+-0.0163|mae: 1.2193+-0.0139
KNNBaseline    |rmse: 0.9323+-0.0055|mae: 0.7345+-0.0041
KNNBasic       |rmse: 0.9801+-0.0063|mae: 0.7744+-0.0054
KNNWithMeans   |rmse: 0.9527+-0.0073|mae: 0.7505+-0.0050
KNNWithZScore  |rmse: 0.9527+-0.0069|mae: 0.7473+-0.0047
BaselineOnly   |rmse: 0.9457+-0.0077|mae: 0.7499+-0.0063
CoClustering   |rmse: 0.9692+-0.0086|mae: 0.7580+-0.0067
```

In [43]:

```
# 全训练集，不同年龄组
age_group_list = ['<20','20-29','30-39','40-49','>=50']
all_results_list = [all_results, all_results2, all_results3, all_results4, all_results5]
for i in range(5):
    print("Age {}".format(age_group_list[i]))
    get_mean_results_age(algorithms, all_results_list, i+1)
```

```
Age <20
SVD            |rmse: 0.9946+-0.0111|mae: 0.7879+-0.0102
SVDpp          |rmse: 0.9673+-0.0158|mae: 0.7563+-0.0121
SlopeOne       |rmse: 0.9997+-0.0115|mae: 0.7875+-0.0095
NMF            |rmse: 1.0209+-0.0143|mae: 0.8032+-0.0095
NormalPredictor|rmse: 1.5531+-0.0138|mae: 1.2430+-0.0170
KNNBaseline    |rmse: 0.9870+-0.0134|mae: 0.7823+-0.0096
KNNBasic       |rmse: 1.0208+-0.0097|mae: 0.8151+-0.0100
KNNWithMeans   |rmse: 1.0136+-0.0168|mae: 0.8119+-0.0122
KNNWithZScore  |rmse: 1.0133+-0.0203|mae: 0.8041+-0.0148
BaselineOnly   |rmse: 1.0081+-0.0111|mae: 0.8024+-0.0094
CoClustering   |rmse: 1.0259+-0.0200|mae: 0.8151+-0.0176
Age 20-29
SVD            |rmse: 0.9387+-0.0129|mae: 0.7401+-0.0076
SVDpp          |rmse: 0.9234+-0.0124|mae: 0.7242+-0.0077
SlopeOne       |rmse: 0.9489+-0.0134|mae: 0.7442+-0.0067
NMF            |rmse: 0.9682+-0.0129|mae: 0.7616+-0.0086
NormalPredictor|rmse: 1.5375+-0.0213|mae: 1.2368+-0.0169
KNNBaseline    |rmse: 0.9346+-0.0111|mae: 0.7365+-0.0061
KNNBasic       |rmse: 1.0015+-0.0195|mae: 0.7913+-0.0160
KNNWithMeans   |rmse: 0.9560+-0.0146|mae: 0.7545+-0.0078
KNNWithZScore  |rmse: 0.9570+-0.0157|mae: 0.7527+-0.0077
BaselineOnly   |rmse: 0.9488+-0.0128|mae: 0.7538+-0.0093
CoClustering   |rmse: 0.9689+-0.0139|mae: 0.7584+-0.0082
Age 30-39
SVD            |rmse: 0.9320+-0.0117|mae: 0.7365+-0.0085
SVDpp          |rmse: 0.9152+-0.0099|mae: 0.7190+-0.0074
SlopeOne       |rmse: 0.9421+-0.0114|mae: 0.7416+-0.0096
NMF            |rmse: 0.9585+-0.0125|mae: 0.7536+-0.0095
NormalPredictor|rmse: 1.5156+-0.0092|mae: 1.2153+-0.0088
KNNBaseline    |rmse: 0.9249+-0.0102|mae: 0.7298+-0.0090
KNNBasic       |rmse: 0.9626+-0.0070|mae: 0.7618+-0.0096
KNNWithMeans   |rmse: 0.9441+-0.0119|mae: 0.7437+-0.0101
KNNWithZScore  |rmse: 0.9441+-0.0115|mae: 0.7407+-0.0096
BaselineOnly   |rmse: 0.9409+-0.0134|mae: 0.7472+-0.0103
CoClustering   |rmse: 0.9629+-0.0117|mae: 0.7524+-0.0108
Age 40-49
SVD            |rmse: 0.9482+-0.0263|mae: 0.7465+-0.0232
SVDpp          |rmse: 0.9362+-0.0273|mae: 0.7341+-0.0234
SlopeOne       |rmse: 0.9590+-0.0257|mae: 0.7545+-0.0207
NMF            |rmse: 0.9745+-0.0236|mae: 0.7676+-0.0172
NormalPredictor|rmse: 1.4913+-0.0318|mae: 1.1983+-0.0274
KNNBaseline    |rmse: 0.9412+-0.0250|mae: 0.7408+-0.0206
KNNBasic       |rmse: 0.9665+-0.0220|mae: 0.7605+-0.0203
KNNWithMeans   |rmse: 0.9618+-0.0280|mae: 0.7545+-0.0223
KNNWithZScore  |rmse: 0.9617+-0.0285|mae: 0.7511+-0.0223
BaselineOnly   |rmse: 0.9540+-0.0270|mae: 0.7547+-0.0236
CoClustering   |rmse: 0.9812+-0.0307|mae: 0.7655+-0.0244
Age >=50
SVD            |rmse: 0.8856+-0.0203|mae: 0.6964+-0.0142
SVDpp          |rmse: 0.8782+-0.0181|mae: 0.6903+-0.0147
SlopeOne       |rmse: 0.8935+-0.0242|mae: 0.7022+-0.0199
NMF            |rmse: 0.9110+-0.0216|mae: 0.7169+-0.0197
NormalPredictor|rmse: 1.4711+-0.0158|mae: 1.1797+-0.0104
KNNBaseline    |rmse: 0.8860+-0.0236|mae: 0.6952+-0.0166
KNNBasic       |rmse: 0.9311+-0.0248|mae: 0.7339+-0.0175
KNNWithMeans   |rmse: 0.9004+-0.0219|mae: 0.7026+-0.0155
KNNWithZScore  |rmse: 0.8965+-0.0218|mae: 0.6975+-0.0163
BaselineOnly   |rmse: 0.8849+-0.0225|mae: 0.6981+-0.0165
CoClustering   |rmse: 0.9236+-0.0225|mae: 0.7177+-0.0162
```