

用这些算法搞训练集与测试集，按年龄分割成5组

注意对于不同数据集的rmse和mae, 要进行合并:

$$\text{RMSE}(X) = \sqrt{\frac{\sum_{t=1}^n (\hat{y}_t - y_t)^2}{n}}$$
$$\text{RMSE}(XY) = \sqrt{\frac{\text{RMSE}(X)^2 \times n_X + \text{RMSE}(Y)^2 \times n_Y}{n_X + n_Y}}$$
$$\text{MAE}(X) = \frac{\sum_{t=1}^n |\hat{y}_t - y_t|}{n}$$
$$\text{MAE}(XY) = \frac{\text{MAE}(X) \times n_X + \text{MAE}(Y) \times n_Y}{n_X + n_Y}$$

```
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)
```

```
def build_train_test(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()
    return data_train, data_test
```

```
In [2]: algorithms = {'SVD':SVD(), 'SVDpp':SVDpp(), 'SlopeOne':SlopeOne(), 'NMF':NMF(), 'NormalPredictor':NormalPredictor(),
                    'KNNBaseline':KNNBaseline(), 'KNNBasic':KNNBasic(), 'KNNWithMeans':KNNWithMeans(),
                    'KNNWithZScore':KNNWithZScore(), 'BaselineOnly':BaselineOnly(), 'CoClustering':CoClustering()}
```

```
In [3]: def train_single_algorithm(algorithm_name, data_train, data_test, save_model=False):
    algorithms = {'SVD':SVD(), 'SVDpp':SVDpp(), 'SlopeOne':SlopeOne(), 'NMF':NMF(), 'NormalPredictor':NormalPredictor(),
                '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]
    # print("{} training started!".format(algorithm_name))
    start_time = time()
    # results = cross_validate(algo, data, measures=['RMSE', 'MAE'], cv=5, verbose=True)
    algo.fit(data_train)
    # print("{} testing started!".format(algorithm_name))
    predictions = algo.test(data_test)
    result = {}
    # result['name'] = algorithm_name
    result['rmse'] = accuracy.rmse(predictions, verbose=True)
    result['mae'] = accuracy.mae(predictions, verbose=True)
    result['n_samples'] = len(data_test)
    if save_model:
        result['model'] = algo
    print("{:<20}|{:.2f} mins|rmse: {:.4f}|mae: {:.4f}".format(algorithm_name,
                                                                (time() - start_time) / 60.,
                                                                result['rmse'],
                                                                result['mae']
                                                                ))

    return result
```

```
In [4]: # note that this is for total dataset, not single gender!
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("{:<20}|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_mae)))
```

```
In [5]: # result_list [{rmse, mae, len}]
def combine_rmse_mae(result_list):
    part1_rmse = 0
    part1_mae = 0
    part2 = 0
    for i in range(len(result_list)):
        curr_rmse = result_list[i]['rmse']
        curr_mae = result_list[i]['mae']
        curr_len = result_list[i]['n_samples']
        part1_rmse += curr_rmse ** 2 * curr_len
        part1_mae += curr_mae * curr_len
        part2 += curr_len
    return np.sqrt(part1_rmse / part2), part1_mae / part2
```

```
In [6]: # note that this is for total dataset, not single age
def combine_age(algorithms, all_results_list):
    """
    all_results_list: all_results_age1, ..., all_results_age5
    """
    assert (len(all_results_list) == 5), "5 age groups!"
    all_results = {}
    for curr_algo_name in algorithms.keys():
        result_list = []
        # build curr result list for curr algo name
        for i in range(len(all_results_list)):
            result_list.append(all_results_list[i][curr_algo_name])
        # compute combined rmse and mae
        rmse_combined, mae_combined = combine_rmse_mae(result_list)
        all_results[curr_algo_name] = {'rmse':rmse_combined, 'mae':mae_combined}
    return all_results
```

u1

```
In [7]: # load
df_train = pd.read_csv("data/ml-100k_merged/u1.base")
df_test = pd.read_csv("data/ml-100k_merged/u1.test")
```

```
In [8]: # split by age
reader = Reader(rating_scale=(1, 5))
df_train_age1 = df_train[df_train['age'] < 20]
data_train_age1 = Dataset.load_from_df(df_train_age1[['user_id', 'movie_id', 'rating']], reader)
data_train_age1 = data_train_age1.build_full_trainset()
df_train_age2 = df_train[(df_train['age'] >= 20) & (df_train['age'] < 30)]
data_train_age2 = Dataset.load_from_df(df_train_age2[['user_id', 'movie_id', 'rating']], reader)
data_train_age2 = data_train_age2.build_full_trainset()
df_train_age3 = df_train[(df_train['age'] >= 30) & (df_train['age'] < 40)]
data_train_age3 = Dataset.load_from_df(df_train_age3[['user_id', 'movie_id', 'rating']], reader)
data_train_age3 = data_train_age3.build_full_trainset()
df_train_age4 = df_train[(df_train['age'] >= 40) & (df_train['age'] < 50)]
data_train_age4 = Dataset.load_from_df(df_train_age4[['user_id', 'movie_id', 'rating']], reader)
data_train_age4 = data_train_age4.build_full_trainset()
df_train_age5 = df_train[df_train['age'] >= 50]
data_train_age5 = Dataset.load_from_df(df_train_age5[['user_id', 'movie_id', 'rating']], reader)
data_train_age5 = data_train_age5.build_full_trainset()
```

```
In [9]: # split by age
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()
```

```
In [10]: all_results_age1 = {}
save_model = False
for algorithm_name in algorithms.keys():
    result = train_single_algorithm(algorithm_name, data_train_age1, data_test_age1, save_model)
    all_results_age1[algorithm_name] = result
print("=====")

all_results_age2 = {}
save_model = False
for algorithm_name in algorithms.keys():
    result = train_single_algorithm(algorithm_name, data_train_age2, data_test_age2, save_model)
    all_results_age2[algorithm_name] = result
print("=====")

all_results_age3 = {}
save_model = False
for algorithm_name in algorithms.keys():
    result = train_single_algorithm(algorithm_name, data_train_age3, data_test_age3, save_model)
    all_results_age3[algorithm_name] = result
print("=====")

all_results_age4 = {}
save_model = False
for algorithm_name in algorithms.keys():
    result = train_single_algorithm(algorithm_name, data_train_age4, data_test_age4, save_model)
    all_results_age4[algorithm_name] = result
print("=====")

all_results_age5 = {}
save_model = False
for algorithm_name in algorithms.keys():
    result = train_single_algorithm(algorithm_name, data_train_age5, data_test_age5, save_model)
    all_results_age5[algorithm_name] = result
print("=====")
```

```
RMSE: 1.0356
MAE: 0.8259
SVD |0.01 mins|rmse: 1.0356|mae: 0.8259
RMSE: 1.0153
MAE: 0.8042
SVDpp |0.22 mins|rmse: 1.0153|mae: 0.8042
RMSE: 1.0896
MAE: 0.8535
SlopeOne |0.00 mins|rmse: 1.0896|mae: 0.8535
RMSE: 1.1261
MAE: 0.8888
NMF |0.01 mins|rmse: 1.1261|mae: 0.8888
RMSE: 1.5718
MAE: 1.2661
NormalPredictor |0.00 mins|rmse: 1.5718|mae: 1.2661
Estimating biases using als...
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0727
MAE: 0.8458
KNNBaseline |0.00 mins|rmse: 1.0727|mae: 0.8458
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.1444
MAE: 0.9036
KNNBasic |0.00 mins|rmse: 1.1444|mae: 0.9036
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0844
MAE: 0.8562
KNNWithMeans |0.00 mins|rmse: 1.0844|mae: 0.8562
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0685
MAE: 0.8412
KNNWithZScore |0.00 mins|rmse: 1.0685|mae: 0.8412
Estimating biases using als...
RMSE: 1.0411
MAE: 0.8350
BaselineOnly |0.00 mins|rmse: 1.0411|mae: 0.8350
RMSE: 1.1278
MAE: 0.8869
CoClustering |0.00 mins|rmse: 1.1278|mae: 0.8869
=====
RMSE: 0.9748
MAE: 0.7699
SVD |0.02 mins|rmse: 0.9748|mae: 0.7699
RMSE: 0.9483
MAE: 0.7444
SVDpp |0.92 mins|rmse: 0.9483|mae: 0.7444
RMSE: 0.9780
MAE: 0.7617
SlopeOne |0.01 mins|rmse: 0.9780|mae: 0.7617
RMSE: 1.0208
MAE: 0.7978
NMF |0.02 mins|rmse: 1.0208|mae: 0.7978
RMSE: 1.5860
MAE: 1.2769
NormalPredictor |0.00 mins|rmse: 1.5860|mae: 1.2769
Estimating biases using als...
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9592
MAE: 0.7519
KNNBaseline |0.01 mins|rmse: 0.9592|mae: 0.7519
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0582
MAE: 0.8298
KNNBasic |0.01 mins|rmse: 1.0582|mae: 0.8298
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9846
MAE: 0.7677
KNNWithMeans |0.01 mins|rmse: 0.9846|mae: 0.7677
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9861
MAE: 0.7658
KNNWithZScore |0.01 mins|rmse: 0.9861|mae: 0.7658
Estimating biases using als...
RMSE: 0.9770
MAE: 0.7772
BaselineOnly |0.00 mins|rmse: 0.9770|mae: 0.7772
RMSE: 1.0142
MAE: 0.7864
CoClustering |0.01 mins|rmse: 1.0142|mae: 0.7864
=====
```

```
RMSE: 0.9855
MAE: 0.7775
SVD |0.02 mins|rmse: 0.9855|mae: 0.7775
RMSE: 0.9675
MAE: 0.7604
SVDpp |0.52 mins|rmse: 0.9675|mae: 0.7604
RMSE: 0.9947
MAE: 0.7784
SlopeOne |0.01 mins|rmse: 0.9947|mae: 0.7784
RMSE: 1.0394
MAE: 0.8107
NMF |0.01 mins|rmse: 1.0394|mae: 0.8107
RMSE: 1.5127
MAE: 1.2074
NormalPredictor |0.00 mins|rmse: 1.5127|mae: 1.2074
Estimating biases using als...
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9750
MAE: 0.7672
KNNBaseline |0.01 mins|rmse: 0.9750|mae: 0.7672
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0160
MAE: 0.8071
KNNBasic |0.00 mins|rmse: 1.0160|mae: 0.8071
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9954
MAE: 0.7826
KNNWithMeans |0.01 mins|rmse: 0.9954|mae: 0.7826
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9952
MAE: 0.7795
KNNWithZScore |0.01 mins|rmse: 0.9952|mae: 0.7795
Estimating biases using als...
RMSE: 0.9815
MAE: 0.7787
BaselineOnly |0.00 mins|rmse: 0.9815|mae: 0.7787
RMSE: 1.0288
MAE: 0.8041
CoClustering |0.01 mins|rmse: 1.0288|mae: 0.8041
=====
RMSE: 1.0267
MAE: 0.8153
SVD |0.01 mins|rmse: 1.0267|mae: 0.8153
RMSE: 1.0225
MAE: 0.8127
SVDpp |0.27 mins|rmse: 1.0225|mae: 0.8127
RMSE: 1.0412
MAE: 0.8186
SlopeOne |0.00 mins|rmse: 1.0412|mae: 0.8186
RMSE: 1.0928
MAE: 0.8643
NMF |0.01 mins|rmse: 1.0928|mae: 0.8643
RMSE: 1.4826
MAE: 1.1858
NormalPredictor |0.00 mins|rmse: 1.4826|mae: 1.1858
Estimating biases using als...
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0218
MAE: 0.8035
KNNBaseline |0.00 mins|rmse: 1.0218|mae: 0.8035
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0598
MAE: 0.8418
KNNBasic |0.00 mins|rmse: 1.0598|mae: 0.8418
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0443
MAE: 0.8175
KNNWithMeans |0.00 mins|rmse: 1.0443|mae: 0.8175
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0482
MAE: 0.8172
KNNWithZScore |0.00 mins|rmse: 1.0482|mae: 0.8172
Estimating biases using als...
RMSE: 1.0226
MAE: 0.8152
BaselineOnly |0.00 mins|rmse: 1.0226|mae: 0.8152
RMSE: 1.0767
MAE: 0.8442
CoClustering |0.00 mins|rmse: 1.0767|mae: 0.8442
=====
RMSE: 0.8827
```

```
MAE: 0.6981
SVD |0.01 mins|rmse: 0.8827|mae: 0.6981
RMSE: 0.8765
MAE: 0.6906
SVDpp |0.26 mins|rmse: 0.8765|mae: 0.6906
RMSE: 0.9115
MAE: 0.7126
SlopeOne |0.00 mins|rmse: 0.9115|mae: 0.7126
RMSE: 0.9671
MAE: 0.7519
NMF |0.01 mins|rmse: 0.9671|mae: 0.7519
RMSE: 1.3884
MAE: 1.1080
NormalPredictor |0.00 mins|rmse: 1.3884|mae: 1.1080
Estimating biases using als...
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9002
MAE: 0.7065
KNNBaseline |0.00 mins|rmse: 0.9002|mae: 0.7065
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9560
MAE: 0.7551
KNNBasic |0.00 mins|rmse: 0.9560|mae: 0.7551
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9136
MAE: 0.7148
KNNWithMeans |0.00 mins|rmse: 0.9136|mae: 0.7148
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9172
MAE: 0.7128
KNNWithZScore |0.00 mins|rmse: 0.9172|mae: 0.7128
Estimating biases using als...
RMSE: 0.8791
MAE: 0.6962
BaselineOnly |0.00 mins|rmse: 0.8791|mae: 0.6962
RMSE: 0.9440
MAE: 0.7361
CoClustering |0.00 mins|rmse: 0.9440|mae: 0.7361
=====
```



```
In [11]: print(all_results_age1)
print(all_results_age2)
print(all_results_age3)
print(all_results_age4)
print(all_results_age5)
```

```
{'SVD': {'rmse': 1.0355911530929198, 'mae': 0.8258977201563171, 'n_samples': 1427}, 'SVDpp': {'rmse': 1.0153449610921927, 'mae': 0.8041899891344562, 'n_samples': 1427}, 'SlopeOne': {'rmse': 1.0896435120173538, 'mae': 0.8534531857370777, 'n_samples': 1427}, 'NMF': {'rmse': 1.1260847637346978, 'mae': 0.8888242805331005, 'n_samples': 1427}, 'NormalPredictor': {'rmse': 1.5718327791429036, 'mae': 1.2661339873946773, 'n_samples': 1427}, 'KNNBaseline': {'rmse': 1.0727174299177427, 'mae': 0.8457949995417663, 'n_samples': 1427}, 'KNNBasic': {'rmse': 1.144402619709926, 'mae': 0.9035686632317225, 'n_samples': 1427}, 'KNNWithMeans': {'rmse': 1.0843960336783927, 'mae': 0.8562444290067666, 'n_samples': 1427}, 'KNNWithZScore': {'rmse': 1.0684960974536286, 'mae': 0.8412240321834963, 'n_samples': 1427}, 'BaselineOnly': {'rmse': 1.0410801699397199, 'mae': 0.8350379227215281, 'n_samples': 1427}, 'CoClustering': {'rmse': 1.1277582567327074, 'mae': 0.886889709225245, 'n_samples': 1427}}
{'SVD': {'rmse': 0.9748228487462752, 'mae': 0.7699316525430836, 'n_samples': 7444}, 'SVDpp': {'rmse': 0.9483199544312881, 'mae': 0.7444161542130608, 'n_samples': 7444}, 'SlopeOne': {'rmse': 0.9779621794099198, 'mae': 0.7617031762226866, 'n_samples': 7444}, 'NMF': {'rmse': 1.0207726419910903, 'mae': 0.797846785477622, 'n_samples': 7444}, 'NormalPredictor': {'rmse': 1.5860177033465441, 'mae': 1.2768612215275588, 'n_samples': 7444}, 'KNNBaseline': {'rmse': 0.9591673622543911, 'mae': 0.7519207610664174, 'n_samples': 7444}, 'KNNBasic': {'rmse': 1.0581902396309446, 'mae': 0.8297915343090865, 'n_samples': 7444}, 'KNNWithMeans': {'rmse': 0.9846453086440998, 'mae': 0.7677251408511204, 'n_samples': 7444}, 'KNNWithZScore': {'rmse': 0.9861307554949753, 'mae': 0.7657869818032177, 'n_samples': 7444}, 'BaselineOnly': {'rmse': 0.9770000099760291, 'mae': 0.7772213772353992, 'n_samples': 7444}, 'CoClustering': {'rmse': 1.0142096358207655, 'mae': 0.7864059771829947, 'n_samples': 7444}}
{'SVD': {'rmse': 0.9854876633227126, 'mae': 0.7775495632767723, 'n_samples': 5567}, 'SVDpp': {'rmse': 0.9674689726411999, 'mae': 0.7603557687387915, 'n_samples': 5567}, 'SlopeOne': {'rmse': 0.9946793073131991, 'mae': 0.7783723100273322, 'n_samples': 5567}, 'NMF': {'rmse': 1.0394261203901345, 'mae': 0.8107097401008034, 'n_samples': 5567}, 'NormalPredictor': {'rmse': 1.5127103056846458, 'mae': 1.2074442066376507, 'n_samples': 5567}, 'KNNBaseline': {'rmse': 0.974982923492738, 'mae': 0.767185893711634, 'n_samples': 5567}, 'KNNBasic': {'rmse': 1.0159937100292407, 'mae': 0.8070567314816268, 'n_samples': 5567}, 'KNNWithMeans': {'rmse': 0.9953527442472163, 'mae': 0.7826388099496471, 'n_samples': 5567}, 'KNNWithZScore': {'rmse': 0.9952388527720478, 'mae': 0.7794529662710552, 'n_samples': 5567}, 'BaselineOnly': {'rmse': 0.9814998978865292, 'mae': 0.7786590304040875, 'n_samples': 5567}, 'CoClustering': {'rmse': 1.0288169219738237, 'mae': 0.8040995795796035, 'n_samples': 5567}}
{'SVD': {'rmse': 1.026722799669155, 'mae': 0.8152623370050023, 'n_samples': 3199}, 'SVDpp': {'rmse': 1.0225236269613127, 'mae': 0.8126745421536259, 'n_samples': 3199}, 'SlopeOne': {'rmse': 1.0411716855155697, 'mae': 0.8185854730387749, 'n_samples': 3199}, 'NMF': {'rmse': 1.0927558042029457, 'mae': 0.8642613458546583, 'n_samples': 3199}, 'NormalPredictor': {'rmse': 1.4826101332887753, 'mae': 1.185828574690116, 'n_samples': 3199}, 'KNNBaseline': {'rmse': 1.0218448486549936, 'mae': 0.8034973116094617, 'n_samples': 3199}, 'KNNBasic': {'rmse': 1.059775855457691, 'mae': 0.8418120710420678, 'n_samples': 3199}, 'KNNWithMeans': {'rmse': 1.044310605257662, 'mae': 0.8174782827483614, 'n_samples': 3199}, 'KNNWithZScore': {'rmse': 1.0482407352691039, 'mae': 0.8171755861357297, 'n_samples': 3199}, 'BaselineOnly': {'rmse': 1.0226201619246706, 'mae': 0.8152111156324071, 'n_samples': 3199}, 'CoClustering': {'rmse': 1.076706045046876, 'mae': 0.8442330653590135, 'n_samples': 3199}}
{'SVD': {'rmse': 0.8827117339733748, 'mae': 0.6980919971614856, 'n_samples': 2363}, 'SVDpp': {'rmse': 0.8764988210292656, 'mae': 0.6906091246282824, 'n_samples': 2363}, 'SlopeOne': {'rmse': 0.9115474377057277, 'mae': 0.7126361882003286, 'n_samples': 2363}, 'NMF': {'rmse': 0.9671205933533394, 'mae': 0.7519480822253247, 'n_samples': 2363}, 'NormalPredictor': {'rmse': 1.388382605350215, 'mae': 1.107959982967561, 'n_samples': 2363}, 'KNNBaseline': {'rmse': 0.9001849404778313, 'mae': 0.706494937435021, 'n_samples': 2363}, 'KNNBasic': {'rmse': 0.9560113113232773, 'mae': 0.7550767181326061, 'n_samples': 2363}, 'KNNWithMeans': {'rmse': 0.91358476100146, 'mae': 0.7148439882186207, 'n_samples': 2363}, 'KNNWithZScore': {'rmse': 0.9172192611700174, 'mae': 0.7128351444150244, 'n_samples': 2363}, 'BaselineOnly': {'rmse': 0.8790946158042358, 'mae': 0.6961903181688068, 'n_samples': 2363}, 'CoClustering': {'rmse': 0.9439807457970445, 'mae': 0.7361463378762891, 'n_samples': 2363}}
```

```
In [12]: all_results_list = [all_results_age1,all_results_age2,all_results_age3,all_results_age4,all_results_age5]
all_results = combine_age(algorithms, all_results_list)
all_results
```

```
Out[12]: {'SVD': {'rmse': 0.9804161710823103, 'mae': 0.7748080646163581},
'SVDpp': {'rmse': 0.9627147763783225, 'mae': 0.7576784376435913},
'SlopeOne': {'rmse': 0.9938069023152546, 'mae': 0.7761904515369531},
'NMF': {'rmse': 1.0394895783285045, 'mae': 0.813118510312241},
'NormalPredictor': {'rmse': 1.5260461957717841, 'mae': 1.2222572540800594},
'KNNBaseline': {'rmse': 0.975644068708112, 'mae': 0.76575034585074},
'KNNBasic': {'rmse': 1.0417168474378293, 'mae': 0.8218224294098824},
'KNNWithMeans': {'rmse': 0.996784585900515, 'mae': 0.7799023187175346},
'KNNWithZScore': {'rmse': 0.9971285019041994, 'mae': 0.7769366898000435},
'BaselineOnly': {'rmse': 0.9794537911449268, 'mae': 0.7782493975432223},
'CoClustering': {'rmse': 1.029045589736446, 'mae': 0.8018117720609722}}
```



```
In [13]: df_train = pd.read_csv("data/ml-100k_merged/u2.base")
df_test = pd.read_csv("data/ml-100k_merged/u2.test")
# split by age
reader = Reader(rating_scale=(1, 5))
df_train_age1 = df_train[df_train['age'] < 20]
data_train_age1 = Dataset.load_from_df(df_train_age1[['user_id', 'movie_id', 'rating']], reader)
data_train_age1 = data_train_age1.build_full_trainset()
df_train_age2 = df_train[(df_train['age'] >= 20) & (df_train['age'] < 30)]
data_train_age2 = Dataset.load_from_df(df_train_age2[['user_id', 'movie_id', 'rating']], reader)
data_train_age2 = data_train_age2.build_full_trainset()
df_train_age3 = df_train[(df_train['age'] >= 30) & (df_train['age'] < 40)]
data_train_age3 = Dataset.load_from_df(df_train_age3[['user_id', 'movie_id', 'rating']], reader)
data_train_age3 = data_train_age3.build_full_trainset()
df_train_age4 = df_train[(df_train['age'] >= 40) & (df_train['age'] < 50)]
data_train_age4 = Dataset.load_from_df(df_train_age4[['user_id', 'movie_id', 'rating']], reader)
data_train_age4 = data_train_age4.build_full_trainset()
df_train_age5 = df_train[df_train['age'] >= 50]
data_train_age5 = Dataset.load_from_df(df_train_age5[['user_id', 'movie_id', 'rating']], reader)
data_train_age5 = data_train_age5.build_full_trainset()

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()
```

```
In [14]: all_results_age1_u2 = {}
save_model = False
for algorithm_name in algorithms.keys():
    result = train_single_algorithm(algorithm_name, data_train_age1, data_test_age1, save_model)
    all_results_age1_u2[algorithm_name] = result
print("==== =")

all_results_age2_u2 = {}
save_model = False
for algorithm_name in algorithms.keys():
    result = train_single_algorithm(algorithm_name, data_train_age2, data_test_age2, save_model)
    all_results_age2_u2[algorithm_name] = result
print("==== =")

all_results_age3_u2 = {}
save_model = False
for algorithm_name in algorithms.keys():
    result = train_single_algorithm(algorithm_name, data_train_age3, data_test_age3, save_model)
    all_results_age3_u2[algorithm_name] = result
print("==== =")

all_results_age4_u2 = {}
save_model = False
for algorithm_name in algorithms.keys():
    result = train_single_algorithm(algorithm_name, data_train_age4, data_test_age4, save_model)
    all_results_age4_u2[algorithm_name] = result
print("==== =")

all_results_age5_u2 = {}
save_model = False
for algorithm_name in algorithms.keys():
    result = train_single_algorithm(algorithm_name, data_train_age5, data_test_age5, save_model)
    all_results_age5_u2[algorithm_name] = result
print("==== =")
```

```
RMSE: 1.0387
MAE: 0.8305
SVD |0.01 mins|rmse: 1.0387|mae: 0.8305
RMSE: 1.0135
MAE: 0.8071
SVDpp |0.19 mins|rmse: 1.0135|mae: 0.8071
RMSE: 1.0788
MAE: 0.8540
SlopeOne |0.00 mins|rmse: 1.0788|mae: 0.8540
RMSE: 1.1266
MAE: 0.8836
NMF |0.01 mins|rmse: 1.1266|mae: 0.8836
RMSE: 1.6011
MAE: 1.2815
NormalPredictor |0.00 mins|rmse: 1.6011|mae: 1.2815
Estimating biases using als...
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0383
MAE: 0.8180
KNNBaseline |0.00 mins|rmse: 1.0383|mae: 0.8180
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.1091
MAE: 0.8662
KNNBasic |0.00 mins|rmse: 1.1091|mae: 0.8662
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0414
MAE: 0.8223
KNNWithMeans |0.00 mins|rmse: 1.0414|mae: 0.8223
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0365
MAE: 0.8105
KNNWithZScore |0.00 mins|rmse: 1.0365|mae: 0.8105
Estimating biases using als...
RMSE: 1.0378
MAE: 0.8372
BaselineOnly |0.00 mins|rmse: 1.0378|mae: 0.8372
RMSE: 1.0815
MAE: 0.8555
CoClustering |0.00 mins|rmse: 1.0815|mae: 0.8555
=====
RMSE: 0.9696
MAE: 0.7617
SVD |0.02 mins|rmse: 0.9696|mae: 0.7617
RMSE: 0.9550
MAE: 0.7480
SVDpp |0.89 mins|rmse: 0.9550|mae: 0.7480
RMSE: 0.9882
MAE: 0.7673
SlopeOne |0.01 mins|rmse: 0.9882|mae: 0.7673
RMSE: 1.0080
MAE: 0.7835
NMF |0.02 mins|rmse: 1.0080|mae: 0.7835
RMSE: 1.5756
MAE: 1.2670
NormalPredictor |0.00 mins|rmse: 1.5756|mae: 1.2670
Estimating biases using als...
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9647
MAE: 0.7536
KNNBaseline |0.01 mins|rmse: 0.9647|mae: 0.7536
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0539
MAE: 0.8312
KNNBasic |0.01 mins|rmse: 1.0539|mae: 0.8312
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9939
MAE: 0.7742
KNNWithMeans |0.01 mins|rmse: 0.9939|mae: 0.7742
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9961
MAE: 0.7716
KNNWithZScore |0.01 mins|rmse: 0.9961|mae: 0.7716
Estimating biases using als...
RMSE: 0.9703
MAE: 0.7701
BaselineOnly |0.00 mins|rmse: 0.9703|mae: 0.7701
RMSE: 1.0190
MAE: 0.7914
CoClustering |0.01 mins|rmse: 1.0190|mae: 0.7914
=====
```

```
RMSE: 0.9625
MAE: 0.7606
SVD |0.02 mins|rmse: 0.9625|mae: 0.7606
RMSE: 0.9525
MAE: 0.7464
SVDpp |0.52 mins|rmse: 0.9525|mae: 0.7464
RMSE: 0.9797
MAE: 0.7631
SlopeOne |0.01 mins|rmse: 0.9797|mae: 0.7631
RMSE: 1.0042
MAE: 0.7836
NMF |0.01 mins|rmse: 1.0042|mae: 0.7836
RMSE: 1.4958
MAE: 1.1984
NormalPredictor |0.00 mins|rmse: 1.4958|mae: 1.1984
Estimating biases using als...
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9637
MAE: 0.7537
KNNBaseline |0.01 mins|rmse: 0.9637|mae: 0.7537
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0106
MAE: 0.7955
KNNBasic |0.00 mins|rmse: 1.0106|mae: 0.7955
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9818
MAE: 0.7686
KNNWithMeans |0.00 mins|rmse: 0.9818|mae: 0.7686
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9876
MAE: 0.7681
KNNWithZScore |0.01 mins|rmse: 0.9876|mae: 0.7681
Estimating biases using als...
RMSE: 0.9609
MAE: 0.7610
BaselineOnly |0.00 mins|rmse: 0.9609|mae: 0.7610
RMSE: 1.0098
MAE: 0.7882
CoClustering |0.01 mins|rmse: 1.0098|mae: 0.7882
=====
RMSE: 0.9755
MAE: 0.7747
SVD |0.01 mins|rmse: 0.9755|mae: 0.7747
RMSE: 0.9663
MAE: 0.7665
SVDpp |0.28 mins|rmse: 0.9663|mae: 0.7665
RMSE: 0.9910
MAE: 0.7793
SlopeOne |0.00 mins|rmse: 0.9910|mae: 0.7793
RMSE: 1.0256
MAE: 0.8006
NMF |0.01 mins|rmse: 1.0256|mae: 0.8006
RMSE: 1.4803
MAE: 1.1888
NormalPredictor |0.00 mins|rmse: 1.4803|mae: 1.1888
Estimating biases using als...
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9729
MAE: 0.7672
KNNBaseline |0.00 mins|rmse: 0.9729|mae: 0.7672
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0115
MAE: 0.7942
KNNBasic |0.00 mins|rmse: 1.0115|mae: 0.7942
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9884
MAE: 0.7798
KNNWithMeans |0.00 mins|rmse: 0.9884|mae: 0.7798
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9879
MAE: 0.7774
KNNWithZScore |0.00 mins|rmse: 0.9879|mae: 0.7774
Estimating biases using als...
RMSE: 0.9709
MAE: 0.7741
BaselineOnly |0.00 mins|rmse: 0.9709|mae: 0.7741
RMSE: 1.0162
MAE: 0.7975
CoClustering |0.00 mins|rmse: 1.0162|mae: 0.7975
=====
RMSE: 0.9138
```

MAE: 0.7217
SVD |0.01 mins|rmse: 0.9138|mae: 0.7217
RMSE: 0.9079
MAE: 0.7153
SVDpp |0.27 mins|rmse: 0.9079|mae: 0.7153
RMSE: 0.9405
MAE: 0.7374
SlopeOne |0.00 mins|rmse: 0.9405|mae: 0.7374
RMSE: 1.0067
MAE: 0.7943
NMF |0.01 mins|rmse: 1.0067|mae: 0.7943
RMSE: 1.4021
MAE: 1.1262
NormalPredictor |0.00 mins|rmse: 1.4021|mae: 1.1262
Estimating biases using als...
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9202
MAE: 0.7229
KNNBaseline |0.00 mins|rmse: 0.9202|mae: 0.7229
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9813
MAE: 0.7767
KNNBasic |0.00 mins|rmse: 0.9813|mae: 0.7767
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9273
MAE: 0.7263
KNNWithMeans |0.00 mins|rmse: 0.9273|mae: 0.7263
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9295
MAE: 0.7266
KNNWithZScore |0.00 mins|rmse: 0.9295|mae: 0.7266
Estimating biases using als...
RMSE: 0.9143
MAE: 0.7235
BaselineOnly |0.00 mins|rmse: 0.9143|mae: 0.7235
RMSE: 0.9685
MAE: 0.7562
CoClustering |0.00 mins|rmse: 0.9685|mae: 0.7562
=====

```
In [15]: all_results_list = [all_results_age1_u2,all_results_age2_u2,all_results_age3_u2,all_results_age4_u2,all_results_age5_u2]
all_results2 = combine_age(algorithms, all_results_list)
all_results2
```

```
Out[15]: {'SVD': {'rmse': 0.9676044521897827, 'mae': 0.7637774963750655},
'SVDpp': {'rmse': 0.9551090504142203, 'mae': 0.7509129222918828},
'SlopeOne': {'rmse': 0.9881675076639034, 'mae': 0.7711544998794113},
'NMF': {'rmse': 1.0193893466620267, 'mae': 0.7953111636980491},
'NormalPredictor': {'rmse': 1.5229530839301966, 'mae': 1.2216425222613434},
'KNNBaseline': {'rmse': 0.9662754912659093, 'mae': 0.7569096507027763},
'KNNBasic': {'rmse': 1.032528432830125, 'mae': 0.8126495659098283},
'KNNWithMeans': {'rmse': 0.9858654642778015, 'mae': 0.771456337596699},
'KNNWithZScore': {'rmse': 0.9879634146704217, 'mae': 0.7690420176543943},
'BaselineOnly': {'rmse': 0.9667866513496378, 'mae': 0.7678967163048456},
'CoClustering': {'rmse': 1.0152042114800102, 'mae': 0.7921536303614217}}
```

u3

```
In [16]: df_train = pd.read_csv("data/ml-100k_merged/u3.base")
df_test = pd.read_csv("data/ml-100k_merged/u3.test")
reader = Reader(rating_scale=(1, 5))
df_train_age1 = df_train[df_train['age'] < 20]
data_train_age1 = Dataset.load_from_df(df_train_age1[['user_id', 'movie_id', 'rating']], reader)
data_train_age1 = data_train_age1.build_full_trainset()
df_train_age2 = df_train[(df_train['age'] >= 20) & (df_train['age'] < 30)]
data_train_age2 = Dataset.load_from_df(df_train_age2[['user_id', 'movie_id', 'rating']], reader)
data_train_age2 = data_train_age2.build_full_trainset()
df_train_age3 = df_train[(df_train['age'] >= 30) & (df_train['age'] < 40)]
data_train_age3 = Dataset.load_from_df(df_train_age3[['user_id', 'movie_id', 'rating']], reader)
data_train_age3 = data_train_age3.build_full_trainset()
df_train_age4 = df_train[(df_train['age'] >= 40) & (df_train['age'] < 50)]
data_train_age4 = Dataset.load_from_df(df_train_age4[['user_id', 'movie_id', 'rating']], reader)
data_train_age4 = data_train_age4.build_full_trainset()
df_train_age5 = df_train[df_train['age'] >= 50]
data_train_age5 = Dataset.load_from_df(df_train_age5[['user_id', 'movie_id', 'rating']], reader)
data_train_age5 = data_train_age5.build_full_trainset()

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()
```

```
In [17]: all_results_age1_u3 = {}
save_model = False
for algorithm_name in algorithms.keys():
    result = train_single_algorithm(algorithm_name, data_train_age1, data_test_age1, save_model)
    all_results_age1_u3[algorithm_name] = result
print("==== = ")
```

```
all_results_age2_u3 = {}
save_model = False
for algorithm_name in algorithms.keys():
    result = train_single_algorithm(algorithm_name, data_train_age2, data_test_age2, save_model)
    all_results_age2_u3[algorithm_name] = result
print("==== = ")
```

```
all_results_age3_u3 = {}
save_model = False
for algorithm_name in algorithms.keys():
    result = train_single_algorithm(algorithm_name, data_train_age3, data_test_age3, save_model)
    all_results_age3_u3[algorithm_name] = result
print("==== = ")
```

```
all_results_age4_u3 = {}
save_model = False
for algorithm_name in algorithms.keys():
    result = train_single_algorithm(algorithm_name, data_train_age4, data_test_age4, save_model)
    all_results_age4_u3[algorithm_name] = result
print("==== = ")
```

```
all_results_age5_u3 = {}
save_model = False
for algorithm_name in algorithms.keys():
    result = train_single_algorithm(algorithm_name, data_train_age5, data_test_age5, save_model)
    all_results_age5_u3[algorithm_name] = result
print("==== = ")
```



```
RMSE: 1.0718
MAE: 0.8623
SVD |0.01 mins|rmse: 1.0718|mae: 0.8623
RMSE: 1.0470
MAE: 0.8300
SVDpp |0.18 mins|rmse: 1.0470|mae: 0.8300
RMSE: 1.1102
MAE: 0.8728
SlopeOne |0.00 mins|rmse: 1.1102|mae: 0.8728
RMSE: 1.1773
MAE: 0.9220
NMF |0.01 mins|rmse: 1.1773|mae: 0.9220
RMSE: 1.5640
MAE: 1.2474
NormalPredictor |0.00 mins|rmse: 1.5640|mae: 1.2474
Estimating biases using als...
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0852
MAE: 0.8535
KNNBaseline |0.00 mins|rmse: 1.0852|mae: 0.8535
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.1439
MAE: 0.9099
KNNBasic |0.00 mins|rmse: 1.1439|mae: 0.9099
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0968
MAE: 0.8575
KNNWithMeans |0.00 mins|rmse: 1.0968|mae: 0.8575
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0962
MAE: 0.8496
KNNWithZScore |0.00 mins|rmse: 1.0962|mae: 0.8496
Estimating biases using als...
RMSE: 1.0623
MAE: 0.8556
BaselineOnly |0.00 mins|rmse: 1.0623|mae: 0.8556
RMSE: 1.1569
MAE: 0.9088
CoClustering |0.00 mins|rmse: 1.1569|mae: 0.9088
=====
RMSE: 0.9478
MAE: 0.7497
SVD |0.02 mins|rmse: 0.9478|mae: 0.7497
RMSE: 0.9341
MAE: 0.7328
SVDpp |0.92 mins|rmse: 0.9341|mae: 0.7328
RMSE: 0.9635
MAE: 0.7568
SlopeOne |0.01 mins|rmse: 0.9635|mae: 0.7568
RMSE: 1.0009
MAE: 0.7830
NMF |0.02 mins|rmse: 1.0009|mae: 0.7830
RMSE: 1.5453
MAE: 1.2373
NormalPredictor |0.00 mins|rmse: 1.5453|mae: 1.2373
Estimating biases using als...
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9392
MAE: 0.7386
KNNBaseline |0.01 mins|rmse: 0.9392|mae: 0.7386
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0157
MAE: 0.7976
KNNBasic |0.01 mins|rmse: 1.0157|mae: 0.7976
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9619
MAE: 0.7559
KNNWithMeans |0.01 mins|rmse: 0.9619|mae: 0.7559
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9611
MAE: 0.7528
KNNWithZScore |0.01 mins|rmse: 0.9611|mae: 0.7528
Estimating biases using als...
RMSE: 0.9475
MAE: 0.7521
BaselineOnly |0.00 mins|rmse: 0.9475|mae: 0.7521
RMSE: 0.9944
MAE: 0.7753
CoClustering |0.01 mins|rmse: 0.9944|mae: 0.7753
=====
```

```
RMSE: 0.9354
MAE: 0.7457
SVD |0.02 mins|rmse: 0.9354|mae: 0.7457
RMSE: 0.9165
MAE: 0.7263
SVDpp |0.55 mins|rmse: 0.9165|mae: 0.7263
RMSE: 0.9419
MAE: 0.7466
SlopeOne |0.01 mins|rmse: 0.9419|mae: 0.7466
RMSE: 0.9818
MAE: 0.7739
NMF |0.02 mins|rmse: 0.9818|mae: 0.7739
RMSE: 1.5079
MAE: 1.2191
NormalPredictor |0.00 mins|rmse: 1.5079|mae: 1.2191
Estimating biases using als...
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9285
MAE: 0.7376
KNNBaseline |0.01 mins|rmse: 0.9285|mae: 0.7376
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9954
MAE: 0.7916
KNNBasic |0.00 mins|rmse: 0.9954|mae: 0.7916
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9425
MAE: 0.7478
KNNWithMeans |0.00 mins|rmse: 0.9425|mae: 0.7478
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9391
MAE: 0.7432
KNNWithZScore |0.01 mins|rmse: 0.9391|mae: 0.7432
Estimating biases using als...
RMSE: 0.9348
MAE: 0.7493
BaselineOnly |0.00 mins|rmse: 0.9348|mae: 0.7493
RMSE: 0.9750
MAE: 0.7738
CoClustering |0.01 mins|rmse: 0.9750|mae: 0.7738
=====
RMSE: 0.9901
MAE: 0.7862
SVD |0.01 mins|rmse: 0.9901|mae: 0.7862
RMSE: 0.9732
MAE: 0.7693
SVDpp |0.28 mins|rmse: 0.9732|mae: 0.7693
RMSE: 1.0423
MAE: 0.8172
SlopeOne |0.00 mins|rmse: 1.0423|mae: 0.8172
RMSE: 1.0686
MAE: 0.8438
NMF |0.01 mins|rmse: 1.0686|mae: 0.8438
RMSE: 1.5091
MAE: 1.2103
NormalPredictor |0.00 mins|rmse: 1.5091|mae: 1.2103
Estimating biases using als...
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9971
MAE: 0.7835
KNNBaseline |0.00 mins|rmse: 0.9971|mae: 0.7835
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0480
MAE: 0.8206
KNNBasic |0.00 mins|rmse: 1.0480|mae: 0.8206
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0176
MAE: 0.7972
KNNWithMeans |0.00 mins|rmse: 1.0176|mae: 0.7972
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0177
MAE: 0.7924
KNNWithZScore |0.00 mins|rmse: 1.0177|mae: 0.7924
Estimating biases using als...
RMSE: 0.9816
MAE: 0.7797
BaselineOnly |0.00 mins|rmse: 0.9816|mae: 0.7797
RMSE: 1.0564
MAE: 0.8250
CoClustering |0.00 mins|rmse: 1.0564|mae: 0.8250
=====
RMSE: 0.9182
```

MAE: 0.7123
SVD |0.01 mins|rmse: 0.9182|mae: 0.7123
RMSE: 0.9091
MAE: 0.7057
SVDpp |0.24 mins|rmse: 0.9091|mae: 0.7057
RMSE: 0.9555
MAE: 0.7441
SlopeOne |0.00 mins|rmse: 0.9555|mae: 0.7441
RMSE: 0.9966
MAE: 0.7768
NMF |0.01 mins|rmse: 0.9966|mae: 0.7768
RMSE: 1.4217
MAE: 1.1445
NormalPredictor |0.00 mins|rmse: 1.4217|mae: 1.1445
Estimating biases using als...
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9292
MAE: 0.7193
KNNBaseline |0.00 mins|rmse: 0.9292|mae: 0.7193
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0050
MAE: 0.7906
KNNBasic |0.00 mins|rmse: 1.0050|mae: 0.7906
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9479
MAE: 0.7345
KNNWithMeans |0.00 mins|rmse: 0.9479|mae: 0.7345
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9486
MAE: 0.7329
KNNWithZScore |0.00 mins|rmse: 0.9486|mae: 0.7329
Estimating biases using als...
RMSE: 0.9125
MAE: 0.7085
BaselineOnly |0.00 mins|rmse: 0.9125|mae: 0.7085
RMSE: 0.9753
MAE: 0.7593
CoClustering |0.00 mins|rmse: 0.9753|mae: 0.7593
=====

```
In [18]: all_results_list = [all_results_age1_u3,all_results_age2_u3,all_results_age3_u3,all_results_age4_u3,all_results_age5_u3]
all_results3 = combine_age(algorithms, all_results_list)
all_results3
```

Out[18]: {'SVD': {'rmse': 0.959344868604272, 'mae': 0.7595901688456372},
'SVDpp': {'rmse': 0.9433128986025361, 'mae': 0.7419806922943417},
'SlopeOne': {'rmse': 0.9834414762723772, 'mae': 0.7721179509810501},
'NMF': {'rmse': 1.023120295085004, 'mae': 0.8015328714593942},
'NormalPredictor': {'rmse': 1.5173494301094996, 'mae': 1.2180029364638136},
'KNNBaseline': {'rmse': 0.9581842018114709, 'mae': 0.7529757023435228},
'KNNBasic': {'rmse': 1.026429951181892, 'mae': 0.8087399376686832},
'KNNWithMeans': {'rmse': 0.9767886121121776, 'mae': 0.7664965149621841},
'KNNWithZScore': {'rmse': 0.9756962777246977, 'mae': 0.7625195167611328},
'BaselineOnly': {'rmse': 0.9561513515779696, 'mae': 0.7593768798050246},
'CoClustering': {'rmse': 1.0124118177399923, 'mae': 0.7923201753581459}}

u4

```
In [19]: df_train = pd.read_csv("data/ml-100k_merged/u4.base")
df_test = pd.read_csv("data/ml-100k_merged/u4.test")
reader = Reader(rating_scale=(1, 5))
df_train_age1 = df_train[df_train['age'] < 20]
data_train_age1 = Dataset.load_from_df(df_train_age1[['user_id', 'movie_id', 'rating']], reader)
data_train_age1 = data_train_age1.build_full_trainset()
df_train_age2 = df_train[(df_train['age'] >= 20) & (df_train['age'] < 30)]
data_train_age2 = Dataset.load_from_df(df_train_age2[['user_id', 'movie_id', 'rating']], reader)
data_train_age2 = data_train_age2.build_full_trainset()
df_train_age3 = df_train[(df_train['age'] >= 30) & (df_train['age'] < 40)]
data_train_age3 = Dataset.load_from_df(df_train_age3[['user_id', 'movie_id', 'rating']], reader)
data_train_age3 = data_train_age3.build_full_trainset()
df_train_age4 = df_train[(df_train['age'] >= 40) & (df_train['age'] < 50)]
data_train_age4 = Dataset.load_from_df(df_train_age4[['user_id', 'movie_id', 'rating']], reader)
data_train_age4 = data_train_age4.build_full_trainset()
df_train_age5 = df_train[df_train['age'] >= 50]
data_train_age5 = Dataset.load_from_df(df_train_age5[['user_id', 'movie_id', 'rating']], reader)
data_train_age5 = data_train_age5.build_full_trainset()

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()
```

```
In [20]: all_results_age1_u4 = {}
save_model = False
for algorithm_name in algorithms.keys():
    result = train_single_algorithm(algorithm_name, data_train_age1, data_test_age1, save_model)
    all_results_age1_u4[algorithm_name] = result
print("==== =")

all_results_age2_u4 = {}
save_model = False
for algorithm_name in algorithms.keys():
    result = train_single_algorithm(algorithm_name, data_train_age2, data_test_age2, save_model)
    all_results_age2_u4[algorithm_name] = result
print("==== =")

all_results_age3_u4 = {}
save_model = False
for algorithm_name in algorithms.keys():
    result = train_single_algorithm(algorithm_name, data_train_age3, data_test_age3, save_model)
    all_results_age3_u4[algorithm_name] = result
print("==== =")

all_results_age4_u4 = {}
save_model = False
for algorithm_name in algorithms.keys():
    result = train_single_algorithm(algorithm_name, data_train_age4, data_test_age4, save_model)
    all_results_age4_u4[algorithm_name] = result
print("==== =")

all_results_age5_u4 = {}
save_model = False
for algorithm_name in algorithms.keys():
    result = train_single_algorithm(algorithm_name, data_train_age5, data_test_age5, save_model)
    all_results_age5_u4[algorithm_name] = result
print("==== =")
```

```
RMSE: 1.0514
MAE: 0.8355
SVD |0.01 mins|rmse: 1.0514|mae: 0.8355
RMSE: 1.0376
MAE: 0.8155
SVDpp |0.18 mins|rmse: 1.0376|mae: 0.8155
RMSE: 1.1118
MAE: 0.8651
SlopeOne |0.00 mins|rmse: 1.1118|mae: 0.8651
RMSE: 1.1610
MAE: 0.9048
NMF |0.01 mins|rmse: 1.1610|mae: 0.9048
RMSE: 1.5516
MAE: 1.2486
NormalPredictor |0.00 mins|rmse: 1.5516|mae: 1.2486
Estimating biases using als...
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0701
MAE: 0.8361
KNNBaseline |0.00 mins|rmse: 1.0701|mae: 0.8361
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.1257
MAE: 0.8940
KNNBasic |0.00 mins|rmse: 1.1257|mae: 0.8940
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0834
MAE: 0.8432
KNNWithMeans |0.00 mins|rmse: 1.0834|mae: 0.8432
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0776
MAE: 0.8330
KNNWithZScore |0.00 mins|rmse: 1.0776|mae: 0.8330
Estimating biases using als...
RMSE: 1.0520
MAE: 0.8417
BaselineOnly |0.00 mins|rmse: 1.0520|mae: 0.8417
RMSE: 1.1265
MAE: 0.8753
CoClustering |0.00 mins|rmse: 1.1265|mae: 0.8753
=====
RMSE: 0.9518
MAE: 0.7493
SVD |0.02 mins|rmse: 0.9518|mae: 0.7493
RMSE: 0.9318
MAE: 0.7300
SVDpp |0.92 mins|rmse: 0.9318|mae: 0.7300
RMSE: 0.9614
MAE: 0.7504
SlopeOne |0.01 mins|rmse: 0.9614|mae: 0.7504
RMSE: 0.9997
MAE: 0.7790
NMF |0.02 mins|rmse: 0.9997|mae: 0.7790
RMSE: 1.5338
MAE: 1.2354
NormalPredictor |0.00 mins|rmse: 1.5338|mae: 1.2354
Estimating biases using als...
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9458
MAE: 0.7393
KNNBaseline |0.01 mins|rmse: 0.9458|mae: 0.7393
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0167
MAE: 0.7964
KNNBasic |0.01 mins|rmse: 1.0167|mae: 0.7964
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9614
MAE: 0.7512
KNNWithMeans |0.01 mins|rmse: 0.9614|mae: 0.7512
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9613
MAE: 0.7492
KNNWithZScore |0.01 mins|rmse: 0.9613|mae: 0.7492
Estimating biases using als...
RMSE: 0.9500
MAE: 0.7532
BaselineOnly |0.00 mins|rmse: 0.9500|mae: 0.7532
RMSE: 0.9930
MAE: 0.7703
CoClustering |0.01 mins|rmse: 0.9930|mae: 0.7703
=====
```

```
RMSE: 0.9640
MAE: 0.7712
SVD |0.02 mins|rmse: 0.9640|mae: 0.7712
RMSE: 0.9542
MAE: 0.7583
SVDpp |0.55 mins|rmse: 0.9542|mae: 0.7583
RMSE: 0.9809
MAE: 0.7759
SlopeOne |0.01 mins|rmse: 0.9809|mae: 0.7759
RMSE: 1.0158
MAE: 0.7984
NMF |0.02 mins|rmse: 1.0158|mae: 0.7984
RMSE: 1.5164
MAE: 1.2146
NormalPredictor |0.00 mins|rmse: 1.5164|mae: 1.2146
Estimating biases using als...
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9644
MAE: 0.7640
KNNBaseline |0.01 mins|rmse: 0.9644|mae: 0.7640
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0256
MAE: 0.8154
KNNBasic |0.00 mins|rmse: 1.0256|mae: 0.8154
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9843
MAE: 0.7773
KNNWithMeans |0.00 mins|rmse: 0.9843|mae: 0.7773
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9837
MAE: 0.7731
KNNWithZScore |0.01 mins|rmse: 0.9837|mae: 0.7731
Estimating biases using als...
RMSE: 0.9634
MAE: 0.7749
BaselineOnly |0.00 mins|rmse: 0.9634|mae: 0.7749
RMSE: 1.0151
MAE: 0.7983
CoClustering |0.01 mins|rmse: 1.0151|mae: 0.7983
=====
RMSE: 0.9559
MAE: 0.7561
SVD |0.01 mins|rmse: 0.9559|mae: 0.7561
RMSE: 0.9474
MAE: 0.7464
SVDpp |0.29 mins|rmse: 0.9474|mae: 0.7464
RMSE: 1.0011
MAE: 0.7850
SlopeOne |0.00 mins|rmse: 1.0011|mae: 0.7850
RMSE: 1.0369
MAE: 0.8119
NMF |0.01 mins|rmse: 1.0369|mae: 0.8119
RMSE: 1.4251
MAE: 1.1381
NormalPredictor |0.00 mins|rmse: 1.4251|mae: 1.1381
Estimating biases using als...
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9644
MAE: 0.7551
KNNBaseline |0.00 mins|rmse: 0.9644|mae: 0.7551
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0170
MAE: 0.7980
KNNBasic |0.00 mins|rmse: 1.0170|mae: 0.7980
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9785
MAE: 0.7634
KNNWithMeans |0.00 mins|rmse: 0.9785|mae: 0.7634
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9760
MAE: 0.7573
KNNWithZScore |0.00 mins|rmse: 0.9760|mae: 0.7573
Estimating biases using als...
RMSE: 0.9483
MAE: 0.7516
BaselineOnly |0.00 mins|rmse: 0.9483|mae: 0.7516
RMSE: 1.0217
MAE: 0.7952
CoClustering |0.00 mins|rmse: 1.0217|mae: 0.7952
=====
RMSE: 0.8994
```


MAE: 0.7140
SVD | 0.01 mins | rmse: 0.8994 | mae: 0.7140
RMSE: 0.8939
MAE: 0.7050
SVDpp | 0.28 mins | rmse: 0.8939 | mae: 0.7050
RMSE: 0.9567
MAE: 0.7524
SlopeOne | 0.00 mins | rmse: 0.9567 | mae: 0.7524
RMSE: 0.9870
MAE: 0.7798
NMF | 0.01 mins | rmse: 0.9870 | mae: 0.7798
RMSE: 1.4122
MAE: 1.1348
NormalPredictor | 0.00 mins | rmse: 1.4122 | mae: 1.1348
Estimating biases using als...
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9205
MAE: 0.7265
KNNBaseline | 0.00 mins | rmse: 0.9205 | mae: 0.7265
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9981
MAE: 0.7961
KNNBasic | 0.00 mins | rmse: 0.9981 | mae: 0.7961
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9372
MAE: 0.7404
KNNWithMeans | 0.00 mins | rmse: 0.9372 | mae: 0.7404
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9391
MAE: 0.7357
KNNWithZScore | 0.00 mins | rmse: 0.9391 | mae: 0.7357
Estimating biases using als...
RMSE: 0.8931
MAE: 0.7075
BaselineOnly | 0.00 mins | rmse: 0.8931 | mae: 0.7075
RMSE: 0.9799
MAE: 0.7795
CoClustering | 0.00 mins | rmse: 0.9799 | mae: 0.7795
=====

```
In [21]: all_results_list = [all_results_age1_u4,all_results_age2_u4,all_results_age3_u4,all_results_age4_u4,all_results_age5_u4]
all_results4 = combine_age(algorithms, all_results_list)
all_results4
```

```
Out[21]: {'SVD': {'rmse': 0.9592003094696314, 'mae': 0.7596219587601744},
'SVDpp': {'rmse': 0.9455677397935904, 'mae': 0.7444168104937434},
'SlopeOne': {'rmse': 0.9855715175440845, 'mae': 0.7721072353091022},
'NMF': {'rmse': 1.0228644268015323, 'mae': 0.7997458565558258},
'NormalPredictor': {'rmse': 1.502855565168063, 'mae': 1.20626025329921},
'KNNBaseline': {'rmse': 0.9619922724055916, 'mae': 0.7549293601819816},
'KNNBasic': {'rmse': 1.0268934316652445, 'mae': 0.8098133938572302},
'KNNWithMeans': {'rmse': 0.9783620541956775, 'mae': 0.7664434730592435},
'KNNWithZScore': {'rmse': 0.9774775589160908, 'mae': 0.7623055069152185},
'BaselineOnly': {'rmse': 0.9566532210166214, 'mae': 0.7613178593726582},
'CoClustering': {'rmse': 1.013692859858577, 'mae': 0.7911098102798972}}
```

```
In [22]: df_train = pd.read_csv("data/ml-100k_merged/u5.base")
df_test = pd.read_csv("data/ml-100k_merged/u5.test")
reader = Reader(rating_scale=(1, 5))
df_train_age1 = df_train[df_train['age'] < 20]
data_train_age1 = Dataset.load_from_df(df_train_age1[['user_id', 'movie_id', 'rating']], reader)
data_train_age1 = data_train_age1.build_full_trainset()
df_train_age2 = df_train[(df_train['age'] >= 20) & (df_train['age'] < 30)]
data_train_age2 = Dataset.load_from_df(df_train_age2[['user_id', 'movie_id', 'rating']], reader)
data_train_age2 = data_train_age2.build_full_trainset()
df_train_age3 = df_train[(df_train['age'] >= 30) & (df_train['age'] < 40)]
data_train_age3 = Dataset.load_from_df(df_train_age3[['user_id', 'movie_id', 'rating']], reader)
data_train_age3 = data_train_age3.build_full_trainset()
df_train_age4 = df_train[(df_train['age'] >= 40) & (df_train['age'] < 50)]
data_train_age4 = Dataset.load_from_df(df_train_age4[['user_id', 'movie_id', 'rating']], reader)
data_train_age4 = data_train_age4.build_full_trainset()
df_train_age5 = df_train[df_train['age'] >= 50]
data_train_age5 = Dataset.load_from_df(df_train_age5[['user_id', 'movie_id', 'rating']], reader)
data_train_age5 = data_train_age5.build_full_trainset()

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()
```

```
In [23]: all_results_age1_u5 = {}
save_model = False
for algorithm_name in algorithms.keys():
    result = train_single_algorithm(algorithm_name, data_train_age1, data_test_age1, save_model)
    all_results_age1_u5[algorithm_name] = result
print("==== =")

all_results_age2_u5 = {}
save_model = False
for algorithm_name in algorithms.keys():
    result = train_single_algorithm(algorithm_name, data_train_age2, data_test_age2, save_model)
    all_results_age2_u5[algorithm_name] = result
print("==== =")

all_results_age3_u5 = {}
save_model = False
for algorithm_name in algorithms.keys():
    result = train_single_algorithm(algorithm_name, data_train_age3, data_test_age3, save_model)
    all_results_age3_u5[algorithm_name] = result
print("==== =")

all_results_age4_u5 = {}
save_model = False
for algorithm_name in algorithms.keys():
    result = train_single_algorithm(algorithm_name, data_train_age4, data_test_age4, save_model)
    all_results_age4_u5[algorithm_name] = result
print("==== =")

all_results_age5_u5 = {}
save_model = False
for algorithm_name in algorithms.keys():
    result = train_single_algorithm(algorithm_name, data_train_age5, data_test_age5, save_model)
    all_results_age5_u5[algorithm_name] = result
print("==== =")
```

```
RMSE: 1.0737
MAE: 0.8557
SVD |0.01 mins|rmse: 1.0737|mae: 0.8557
RMSE: 1.0464
MAE: 0.8261
SVDpp |0.18 mins|rmse: 1.0464|mae: 0.8261
RMSE: 1.1264
MAE: 0.8757
SlopeOne |0.00 mins|rmse: 1.1264|mae: 0.8757
RMSE: 1.1849
MAE: 0.9286
NMF |0.01 mins|rmse: 1.1849|mae: 0.9286
RMSE: 1.5700
MAE: 1.2618
NormalPredictor |0.00 mins|rmse: 1.5700|mae: 1.2618
Estimating biases using als...
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0968
MAE: 0.8595
KNNBaseline |0.00 mins|rmse: 1.0968|mae: 0.8595
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.1486
MAE: 0.9097
KNNBasic |0.00 mins|rmse: 1.1486|mae: 0.9097
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.1109
MAE: 0.8682
KNNWithMeans |0.00 mins|rmse: 1.1109|mae: 0.8682
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.1103
MAE: 0.8624
KNNWithZScore |0.00 mins|rmse: 1.1103|mae: 0.8624
Estimating biases using als...
RMSE: 1.0655
MAE: 0.8550
BaselineOnly |0.00 mins|rmse: 1.0655|mae: 0.8550
RMSE: 1.1404
MAE: 0.8904
CoClustering |0.00 mins|rmse: 1.1404|mae: 0.8904
=====
RMSE: 0.9399
MAE: 0.7517
SVD |0.02 mins|rmse: 0.9399|mae: 0.7517
RMSE: 0.9253
MAE: 0.7358
SVDpp |0.93 mins|rmse: 0.9253|mae: 0.7358
RMSE: 0.9503
MAE: 0.7523
SlopeOne |0.01 mins|rmse: 0.9503|mae: 0.7523
RMSE: 0.9951
MAE: 0.7864
NMF |0.02 mins|rmse: 0.9951|mae: 0.7864
RMSE: 1.5623
MAE: 1.2638
NormalPredictor |0.00 mins|rmse: 1.5623|mae: 1.2638
Estimating biases using als...
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9338
MAE: 0.7452
KNNBaseline |0.01 mins|rmse: 0.9338|mae: 0.7452
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0190
MAE: 0.8080
KNNBasic |0.01 mins|rmse: 1.0190|mae: 0.8080
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9510
MAE: 0.7564
KNNWithMeans |0.01 mins|rmse: 0.9510|mae: 0.7564
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9506
MAE: 0.7535
KNNWithZScore |0.01 mins|rmse: 0.9506|mae: 0.7535
Estimating biases using als...
RMSE: 0.9445
MAE: 0.7593
BaselineOnly |0.00 mins|rmse: 0.9445|mae: 0.7593
RMSE: 0.9902
MAE: 0.7812
CoClustering |0.01 mins|rmse: 0.9902|mae: 0.7812
=====
```

```
RMSE: 0.9631
MAE: 0.7619
SVD |0.02 mins|rmse: 0.9631|mae: 0.7619
RMSE: 0.9445
MAE: 0.7423
SVDpp |0.56 mins|rmse: 0.9445|mae: 0.7423
RMSE: 0.9742
MAE: 0.7641
SlopeOne |0.01 mins|rmse: 0.9742|mae: 0.7641
RMSE: 1.0010
MAE: 0.7897
NMF |0.02 mins|rmse: 1.0010|mae: 0.7897
RMSE: 1.5056
MAE: 1.2017
NormalPredictor |0.00 mins|rmse: 1.5056|mae: 1.2017
Estimating biases using als...
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9536
MAE: 0.7508
KNNBaseline |0.01 mins|rmse: 0.9536|mae: 0.7508
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0140
MAE: 0.8038
KNNBasic |0.00 mins|rmse: 1.0140|mae: 0.8038
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9662
MAE: 0.7560
KNNWithMeans |0.00 mins|rmse: 0.9662|mae: 0.7560
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9658
MAE: 0.7540
KNNWithZScore |0.01 mins|rmse: 0.9658|mae: 0.7540
Estimating biases using als...
RMSE: 0.9606
MAE: 0.7635
BaselineOnly |0.00 mins|rmse: 0.9606|mae: 0.7635
RMSE: 0.9947
MAE: 0.7772
CoClustering |0.01 mins|rmse: 0.9947|mae: 0.7772
=====
RMSE: 0.9610
MAE: 0.7618
SVD |0.01 mins|rmse: 0.9610|mae: 0.7618
RMSE: 0.9485
MAE: 0.7492
SVDpp |0.30 mins|rmse: 0.9485|mae: 0.7492
RMSE: 1.0096
MAE: 0.7930
SlopeOne |0.00 mins|rmse: 1.0096|mae: 0.7930
RMSE: 1.0446
MAE: 0.8192
NMF |0.01 mins|rmse: 1.0446|mae: 0.8192
RMSE: 1.4562
MAE: 1.1651
NormalPredictor |0.00 mins|rmse: 1.4562|mae: 1.1651
Estimating biases using als...
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9775
MAE: 0.7641
KNNBaseline |0.00 mins|rmse: 0.9775|mae: 0.7641
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0272
MAE: 0.8048
KNNBasic |0.00 mins|rmse: 1.0272|mae: 0.8048
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9951
MAE: 0.7788
KNNWithMeans |0.00 mins|rmse: 0.9951|mae: 0.7788
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9915
MAE: 0.7762
KNNWithZScore |0.00 mins|rmse: 0.9915|mae: 0.7762
Estimating biases using als...
RMSE: 0.9544
MAE: 0.7591
BaselineOnly |0.00 mins|rmse: 0.9544|mae: 0.7591
RMSE: 1.0273
MAE: 0.8076
CoClustering |0.00 mins|rmse: 1.0273|mae: 0.8076
=====
RMSE: 0.9476
```

```
MAE: 0.7460
SVD |0.01 mins|rmse: 0.9476|mae: 0.7460
RMSE: 0.9360
MAE: 0.7398
SVDpp |0.29 mins|rmse: 0.9360|mae: 0.7398
RMSE: 0.9828
MAE: 0.7820
SlopeOne |0.00 mins|rmse: 0.9828|mae: 0.7820
RMSE: 1.0279
MAE: 0.8142
NMF |0.01 mins|rmse: 1.0279|mae: 0.8142
RMSE: 1.4310
MAE: 1.1557
NormalPredictor |0.00 mins|rmse: 1.4310|mae: 1.1557
Estimating biases using als...
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9658
MAE: 0.7604
KNNBaseline |0.00 mins|rmse: 0.9658|mae: 0.7604
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 1.0359
MAE: 0.8217
KNNBasic |0.00 mins|rmse: 1.0359|mae: 0.8217
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9740
MAE: 0.7677
KNNWithMeans |0.00 mins|rmse: 0.9740|mae: 0.7677
Computing the msd similarity matrix...
Done computing similarity matrix.
RMSE: 0.9771
MAE: 0.7647
KNNWithZScore |0.00 mins|rmse: 0.9771|mae: 0.7647
Estimating biases using als...
RMSE: 0.9403
MAE: 0.7442
BaselineOnly |0.00 mins|rmse: 0.9403|mae: 0.7442
RMSE: 1.0082
MAE: 0.7950
CoClustering |0.00 mins|rmse: 1.0082|mae: 0.7950
=====
```

```
In [24]: all_results_list = [all_results_age1_u5,all_results_age2_u5,all_results_age3_u5,all_results_age4_u5,all_results_age5_u5]
all_results5 = combine_age(algorithms, all_results_list)
all_results5
```

```
Out[24]: {'SVD': {'rmse': 0.9616959192625387, 'mae': 0.7640022430110269},
'SVDpp': {'rmse': 0.945482650168212, 'mae': 0.7475113011563863},
'SlopeOne': {'rmse': 0.9844126316227517, 'mae': 0.7747622826269642},
'NMF': {'rmse': 1.024705373333123, 'mae': 0.8070147729483735},
'NormalPredictor': {'rmse': 1.5207391881509587, 'mae': 1.2227622736973291},
'KNNBaseline': {'rmse': 0.9633168764596013, 'mae': 0.7607095949908101},
'KNNBasic': {'rmse': 1.0324929109480314, 'mae': 0.8166350318422743},
'KNNWithMeans': {'rmse': 0.9782092380804165, 'mae': 0.7703000818587739},
'KNNWithZScore': {'rmse': 0.9776901967020852, 'mae': 0.7673899302864872},
'BaselineOnly': {'rmse': 0.96045785980103, 'mae': 0.7669526148990355},
'CoClustering': {'rmse': 1.0122509407569489, 'mae': 0.794790692270618}}
```

看下平均水平

```
In [25]: all_results_list = [all_results, all_results2, all_results3, all_results4, all_results5]
get_mean_results(algorithms, all_results_list)
```

SVD	rmse: 0.9657+-0.0080 mae: 0.7644+-0.0056
SVDpp	rmse: 0.9504+-0.0074 mae: 0.7485+-0.0055
SlopeOne	rmse: 0.9871+-0.0037 mae: 0.7733+-0.0019
NMF	rmse: 1.0259+-0.0070 mae: 0.8033+-0.0062
NormalPredictor	rmse: 1.5180+-0.0081 mae: 1.2182+-0.0062
KNNBaseline	rmse: 0.9651+-0.0059 mae: 0.7583+-0.0045
KNNBasic	rmse: 1.0320+-0.0055 mae: 0.8139+-0.0048
KNNWithMeans	rmse: 0.9832+-0.0075 mae: 0.7709+-0.0049
KNNWithZScore	rmse: 0.9832+-0.0082 mae: 0.7676+-0.0054
BaselineOnly	rmse: 0.9639+-0.0087 mae: 0.7668+-0.0066
CoClustering	rmse: 1.0165+-0.0064 mae: 0.7944+-0.0039

In [26]:

```
# age1
all_results_list = [all_results_age1, all_results_age1_u2, all_results_age1_u3, all_results_age1_u4, all_results_age1_u5]
get_mean_results(algorithms, all_results_list)
```

SVD	rmse: 1.0542+-0.0160	mae: 0.8420+-0.0144
SVDpp	rmse: 1.0320+-0.0147	mae: 0.8166+-0.0102
SlopeOne	rmse: 1.1034+-0.0170	mae: 0.8642+-0.0092
NMF	rmse: 1.1552+-0.0248	mae: 0.9056+-0.0177
NormalPredictor	rmse: 1.5717+-0.0163	mae: 1.2611+-0.0126
KNNBaseline	rmse: 1.0726+-0.0196	mae: 0.8426+-0.0146
KNNBasic	rmse: 1.1343+-0.0149	mae: 0.8967+-0.0163
KNNWithMeans	rmse: 1.0834+-0.0233	mae: 0.8495+-0.0157
KNNWithZScore	rmse: 1.0778+-0.0252	mae: 0.8393+-0.0174
BaselineOnly	rmse: 1.0517+-0.0110	mae: 0.8449+-0.0087
CoClustering	rmse: 1.1266+-0.0251	mae: 0.8834+-0.0176

In [27]:

```
# age2
all_results_list = [all_results_age2, all_results_age2_u2, all_results_age2_u3, all_results_age2_u4, all_results_age2_u5]
get_mean_results(algorithms, all_results_list)
```

SVD	rmse: 0.9568+-0.0133	mae: 0.7564+-0.0081
SVDpp	rmse: 0.9389+-0.0110	mae: 0.7382+-0.0069
SlopeOne	rmse: 0.9683+-0.0133	mae: 0.7577+-0.0062
NMF	rmse: 1.0049+-0.0089	mae: 0.7859+-0.0064
NormalPredictor	rmse: 1.5606+-0.0191	mae: 1.2561+-0.0167
KNNBaseline	rmse: 0.9485+-0.0117	mae: 0.7457+-0.0062
KNNBasic	rmse: 1.0327+-0.0191	mae: 0.8126+-0.0152
KNNWithMeans	rmse: 0.9706+-0.0160	mae: 0.7611+-0.0085
KNNWithZScore	rmse: 0.9711+-0.0171	mae: 0.7586+-0.0086
BaselineOnly	rmse: 0.9579+-0.0132	mae: 0.7624+-0.0098
CoClustering	rmse: 1.0022+-0.0120	mae: 0.7809+-0.0075

In [28]:

```
# age3
all_results_list = [all_results_age3, all_results_age3_u2, all_results_age3_u3, all_results_age3_u4, all_results_age3_u5]
get_mean_results(algorithms, all_results_list)
```

SVD	rmse: 0.9621+-0.0159	mae: 0.7634+-0.0108
SVDpp	rmse: 0.9470+-0.0170	mae: 0.7467+-0.0123
SlopeOne	rmse: 0.9743+-0.0175	mae: 0.7656+-0.0113
NMF	rmse: 1.0085+-0.0190	mae: 0.7913+-0.0126
NormalPredictor	rmse: 1.5077+-0.0070	mae: 1.2082+-0.0077
KNNBaseline	rmse: 0.9570+-0.0158	mae: 0.7546+-0.0105
KNNBasic	rmse: 1.0123+-0.0098	mae: 0.8026+-0.0085
KNNWithMeans	rmse: 0.9740+-0.0183	mae: 0.7665+-0.0130
KNNWithZScore	rmse: 0.9743+-0.0201	mae: 0.7636+-0.0132
BaselineOnly	rmse: 0.9603+-0.0149	mae: 0.7655+-0.0105
CoClustering	rmse: 1.0047+-0.0184	mae: 0.7883+-0.0117

In [29]:

```
# age4
all_results_list = [all_results_age4, all_results_age4_u2, all_results_age4_u3, all_results_age4_u4, all_results_age4_u5]
get_mean_results(algorithms, all_results_list)
```

SVD	rmse: 0.9819+-0.0254	mae: 0.7788+-0.0210
SVDpp	rmse: 0.9716+-0.0274	mae: 0.7688+-0.0237
SlopeOne	rmse: 1.0170+-0.0210	mae: 0.7986+-0.0163
NMF	rmse: 1.0537+-0.0241	mae: 0.8279+-0.0230
NormalPredictor	rmse: 1.4707+-0.0283	mae: 1.1776+-0.0244
KNNBaseline	rmse: 0.9868+-0.0206	mae: 0.7747+-0.0171
KNNBasic	rmse: 1.0327+-0.0184	mae: 0.8119+-0.0175
KNNWithMeans	rmse: 1.0048+-0.0236	mae: 0.7873+-0.0185
KNNWithZScore	rmse: 1.0043+-0.0259	mae: 0.7841+-0.0199
BaselineOnly	rmse: 0.9756+-0.0263	mae: 0.7759+-0.0221
CoClustering	rmse: 1.0396+-0.0232	mae: 0.8139+-0.0185

In [30]:

```
# age5
all_results_list = [all_results_age5, all_results_age5_u2, all_results_age5_u3, all_results_age5_u4, all_results_age5_u5]
get_mean_results(algorithms, all_results_list)
```

SVD	rmse: 0.9123+-0.0216	mae: 0.7184+-0.0158
SVDpp	rmse: 0.9047+-0.0196	mae: 0.7113+-0.0163
SlopeOne	rmse: 0.9494+-0.0233	mae: 0.7457+-0.0225
NMF	rmse: 0.9971+-0.0202	mae: 0.7834+-0.0206
NormalPredictor	rmse: 1.4111+-0.0149	mae: 1.1338+-0.0163
KNNBaseline	rmse: 0.9272+-0.0215	mae: 0.7271+-0.0180
KNNBasic	rmse: 0.9953+-0.0264	mae: 0.7880+-0.0220
KNNWithMeans	rmse: 0.9400+-0.0204	mae: 0.7367+-0.0177
KNNWithZScore	rmse: 0.9423+-0.0203	mae: 0.7346+-0.0170
BaselineOnly	rmse: 0.9079+-0.0208	mae: 0.7160+-0.0166
CoClustering	rmse: 0.9752+-0.0207	mae: 0.7652+-0.0203