Data Mining Technology for Business and Society Homework 2

Beatrice Nobile (1908315) Katsiaryna Zavadskaya (1847985)

May 2020

1 Recommendation-System

1.1

Basic algorithms:

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE (testset)	1.5075	1.4957	1.5085	1.4986	1.5080	1.5037	0.0054
Fit time	0.32	0.32	0.33	0.32	0.17	0.29	0.06
Test time	0.52	0.51	0.54	0.53	0.31	0.48	0.09

Table 1: Evaluating RMSE of algorithm NormalPredictor on 5 split(s).

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE (testset)	0.9176	0.9201	0.9174	0.9209	0.9225	0.9197	0.0020
Fit time	0.15	0.15	0.15	0.15	0.10	0.14	0.02
Test time	0.38	0.38	0.38	0.40	0.22	0.35	0.06

Table 2: Evaluating RMSE of algorithm BaselineOnly on 5 split(s).

k-NN inspired algorithms:

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE (testset)	0.9468	0.9517	0.9505	0.9530	0.9560	0.9516	0.0030
Fit time	1.12	1.80	1.92	1.88	1.06	1.56	0.38
Test time	13.29	14.60	14.76	12.28	7.78	12.54	2.55

Table 3: Evaluating RMSE of algorithm KNNBasic on 5 split(s).

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE (testset)	0.9287	0.9347	0.9312	0.9331	0.9344	0.9324	0.0022
Fit time	1.16	1.90	2.02	1.80	1.16	1.61	0.37
Test time	13.97	15.47	15.22	13.20	8.23	13.22	2.63

Table 4: Evaluating RMSE of algorithm KNNWithMeans on 5 split(s).

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE (testset)	0.9272	0.9338	0.9307	0.9322	0.9330	0.9314	0.0023
Fit time	1.30	1.93	2.38	2.00	1.34	1.79	0.41
Test time	14.79	16.58	16.25	14.09	8.52	14.04	2.91

Table 5: Evaluating RMSE of algorithm KNNWithZScore on 5 split(s).

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE (testset)	0.9051	0.9101	0.9065	0.9100	0.9107	0.9085	0.0022
Fit time	1.30	1.88	2.48	2.31	1.21	1.84	0.51
Test time	15.56	17.31	17.18	14.47	8.93	14.69	3.07

Table 6: Evaluating RMSE of algorithm KNNBaseline on 5 split(s).

Matrix Factorization-based algorithms:

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE (testset)	0.9062	0.9108	0.9067	0.9096	0.9107	0.9088	0.0020
Fit time	19.27	22.33	22.27	19.73	11.78	19.08	3.86
Test time	0.53	0.80	0.55	0.53	0.33	0.55	0.15

Table 7: Evaluating RMSE of algorithm SVD on 5 split(s).

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE (testset)	0.8937	0.8922	0.8935	0.8993	0.8945	0.8946	0.0024
Fit time	1224.46	1227.38	1224.14	1222.93	614.74	1102.73	244.00
Test time	21.88	28.00	22.12	22.69	11.59	21.26	5.33

Table 8: Evaluating RMSE of algorithm SVDpp on 5 split(s).

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE (testset)	0.9309	0.9394	0.9331	0.9388	0.9405	0.9366	0.0038
Fit time	20.13	21.05	22.35	19.28	11.66	18.89	3.76
Test time	0.45	0.71	0.44	0.45	0.28	0.47	0.14

Table 9: Evaluating RMSE of algorithm NMF on 5 split(s).

Slope One:

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE (testset)	0.9199	0.9253	0.9209	0.9236	0.9253	0.9230	0.0022
Fit time	5.92	8.01	7.51	7.84	4.92	6.84	1.21
Test time	19.70	20.17	19.80	18.35	10.46	17.70	3.67

Table 10: Evaluating RMSE of algorithm SlopeOne on 5 split(s).

Co-clustering:

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE (testset)	0.9367	0.9466	0.9353	0.9423	0.9509	0.9424	0.0059
Fit time	2.79	2.77	2.82	2.79	1.50	2.53	0.52
Test time	0.43	0.42	0.42	0.41	0.25	0.39	0.07

Table 11: Evaluating RMSE of algorithm CoClustering on 5 split(s).

Rank of all the tested recommendation algorithms:

Rank	Algorithm	Mean_RMSE
1	SVDpp	0.8946
2	KNNBaseline	0.9085
3	SVD	0.9088
4	BaselineOnly	0.9197
5	SlopeOne	0.9230
6	KNNWithZScore	0.9314
7	KNNWithMeans	0.9324
8	NMF	0.9366
9	CoClustering	0.9424
10	KNNBasic	0.9516
11	NormalPredictor	1.5037

Table 12: Rank of the recommendation algorithms according to the MEAN_RMSE metric value: from the best to the worst algorithm

To use all the CPU-cores, the parameter n_jobs must be -1 or 2 if we specify explicitly. To discover the number of CPU-cores we have, we used cpu_count() command.

1.2

Complete "Grid-of-Parameters":

Parameter	Values
k	[20, 40, 60, 80]
min_k	[1, 5]
user_based	[False, True]
name	['pearson_baseline']
shrinkage	[30, 60, 100]

Table 13: Grid of parameters for KNNBaseline

Parameter	Values
n_factors	[100, 200]
n_epochs	[10, 20, 50]
lr_all	[0.001, 0.005, 0.1]
reg_all	[0.0, 0.05, 0.1]

Table 14: Grid of parameters for SVD

Algorithm	Cross-Validation approach	Best configuration	RMSE	Total time
KNNBaseline	RandomizedSearchCV	k: 40, min_k: 1, name: 'pearson_baseline', shrinkage: 60, user_based: False	0.88759	662.08 s
SVD	GridSearchCV	lr_all: 0.005, n_epochs: 50, n_factors: 200, reg_all: 0.1	0.88764	5065.14 s

Table 15: The best tuned estimators

The number of CPU-cores: 2. We used both of the CPU-cores available by setting parameter n_jobs = -1, which allows us to exploit multiprocessing in order to improve processing time comparing to usage of only 1 CPU-core.

2 Local Community Detection with Personalized PageRank

The following are the results of our local communities analysis on the social networks of some characters of the well-known novel series called "A song of Ice and Fire", here considered divided into four books.The four books are, in order: "A Game of Thrones", "A Clash of Kings", "A Storm of Swords", and "A Feast for Crows" (which is considered merged with "A Dance of Dragons"). The characters taken into consideration are also four, namely, "Daenerys-Targaryen", "Jon-Snow", "Samwell-Tarly" and "Tyrion-Lannister".

Following are details of the local communities around the above-mentioned characters for each book, including the two tuning parameters (i.e. the dumping factor, alpha, and exponent), the conductance value, and information on the members of the communities.

Book	Characters	Alpha	Exponent	Conductance	Baratheon	Lannister	Stark	Targaryen	Total # Members
Game of Thrones	Daenerys-Targaryen	0.95	1.0	0.078	0	0	0	3	22
Game of Thrones	Jon-Snow	0.95	1.0	0.079	6	6	13	5	166
Game of Thrones	Samwell-Tarly	0.95	1.0	0.079	6	6	13	5	166
Game of Thrones	Tyrion-Lannister	0.95	1.0	0.079	6	6	13	5	166
Clash of Kings	Daenerys-Targaryen	0.95	0.8	0.099	0	0	0	3	18
Clash of Kings	Jon-Snow	0.95	0.8	0.085	0	0	1	4	28
Clash of Kings	Samwell-Tarly	0.95	0.6	0.085	0	0	1	4	28
Clash of Kings	Tyrion-Lannister	0.95	1.0	0.099	8	6	6	6	160
Storm of Swords	Daenerys-Targaryen.	0.95	0.8	0.071	0	0	0	2	25
Storm of Swords	Jon-Snow	0.95	1.0	0.062	0	0	1	1	74
Storm of Swords	Samwell-Tarly	0.95	1.0	0.061	0	0	1	1	71
Storm of Swords	Tyrion-Lannister	0.95	1.0	0.079	7	9	11	10	204
Feast for Crows	Daenerys-Targaryen	0.95	1.0	0.072	5	8	4	11	298
Feast for Crows	Jon-Snow	0.95	1.0	0.061	2	0	10	2	181
Feast for Crows	Samwell-Tarly	0.95	1.0	0.086	2	0	10	2	177
Feast for Crows	Tyrion-Lannister	0.95	1.0	0.044	5	8	4	11	286

Table 16: