Data Mining Technology for Business and Society

Homework

Team Members:

- **Loretucci Lorenzo** (ID. 1903794)
- **Nana Teukam Yves Gaetan** (ID. 1741352)

Part 1.1

We exploited 6 CPU-cores on our machine specify it with *"n_jobs = 6"* on *cross_validate* algorithm in Surprise library.

Basic Surprise Method

NormalPredictor()

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE	1.5143	1.5120	1.5111	1.5248	1.5108	**1.5146**	0.0053
Fit time	0.13	0.12	0.12	0.13	0.13	0.13	0.00
Test time	0.17	0.16	0.17	0.16	0.17	0.17	0.01

BaselineOnly()

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE	0.9139	0.9191	0.9135	0.9182	0.9192	**0.9168**	0.0026
Fit time	0.68	0.69	0.74	0.70	0.66	0.69	0.03
Test time	0.16	0.14	0.15	0.14	0.13	0.15	0.01

Neighborhood Surprise Methods

KNNBasic()

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE	1.0040	1.0032	1.0066	1.0093	1.0117	**1.0070**	0.0032
Fit time	8.02	8.72	9.11	9.30	8.80	8.79	0.44
Test time	10.66	9.79	9.43	8.95	9.13	9.59	0.60

KNNWithMeans()

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE	0.9125	0.9186	0.9159	0.9197	0.9211	**0.9176**	0.0030
Fit time	8.23	8.82	9.26	9.34	9.17	8.96	0.41
Test time	11.54	11.10	11.21	10.41	9.80	10.81	0.63

KNNWithZScore()

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE	0.9150	0.9207	0.9189	0.9225	0.9240	**0.9202**	0.0031
Fit time	8.45	9.24	9.26	9.58	9.30	9.17	0.38
Test time	12.09	11.70	11.65	10.60	9.73	11.15	0.87

KNNBaseline()

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE	0.9081	0.9142	0.9113	0.9136	0.9143	**0.9123**	0.0024
Fit time	8.97	9.61	9.53	9.46	9.33	9.38	0.23
Test time	11.44	11.69	11.80	11.05	10.88	11.37	0.35

Matrix Factorization-based Surprise Methods

SVD()

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE	0.9058	0.9107	0.9042	0.9073	0.9131	**0.9082**	0.0033
Fit time	8.29	8.51	8.20	8.15	7.93	8.21	0.19
Test time	0.24	0.22	0.26	0.21	0.19	0.22	0.03

SVDpp()

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE	0.8941	0.8960	0.8921	0.8974	0.8947	**0.8949**	0.0018
Fit time	726.38	727.40	729.19	727.99	731.74	728.54	1.84
Test time	12.41	12.08	11.46	11.24	9.04	11.25	1.18

NMF()

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE	0.9321	0.9386	0.9343	0.9373	0.9384	**0.9361**	0.0025
Fit time	8.74	9.38	9.46	9.29	8.66	9.11	0.34
Test time	0.24	0.26	0.37	0.19	0.16	0.24	0.07

Rank the algorithm:

1	2	3	4	5	6	7	8	9
SVDpp	SVD	KNNBaseline	KNNWithMeans	BaselineOnly	KNNWithZScore	NMF	KNNBasic	NormalPredictor

Part 1.2

Random-Search-Cross-Validation + KNNBaseline

Grid-of-Parameter: param_grid = {'bsl_options': {'method': ['als', 'sgd'], 'reg': [1, 2]},
'k': [i for i in range(1, 50)],
'min_k': [a for a in range(1, 21)],
'n_factors':[w for w in range(1, 51)],
'n_epochs':[e for e in range(1, 51)],
'lr_all': uniform(0.002, 0.005),
'reg_all':uniform(0.5, 0.9),
'sim_options': {'name': ['pearson_baseline', 'cosine'], 'min_support': [1, 5],
'user_based': [False, True]}}

Best Configuration: {'bsl_options': {'method': 'als', 'reg': 2}, 'k': 45, 'lr_all': 0.0038009798534072914, 'min_k': 2, 'n_epochs': 30, 'n_factors': 43, 'reg_all': 0.5117999168589823, 'sim_options': {'name': 'pearson_baseline', 'min_support': 1, 'user_based': False}}

Average-RMSE associated with the best estimators: *rmse = 0.8867*

Total time required to select the best estimators: 2.8min

Grid-Search-Cross-Validation + SVD

For SVD we decide to change only the Hyperparameter below.

Grid-of-Parameter: grid = {'n_factors':[100,150, 170],'n_epochs':[50, 100, 120], 'lr_all':[0.005, 0.002],'reg_all':[0.02, 0.1]}

Best Configuration: {'n_factors': 150, 'n_epochs': 120, 'lr_all': 0.005, 'reg_all': 0.1}

Average-RMSE associated with the best estimators: *rmse = 0.8814*

Total time required to select the best estimators: 36.6min

CPU

We exploited 6 CPU-cores on our machine specify it with *"n_jobs = 6"* on *RandomizedSearchCV* and *GridSearchCV* function in Surprise library.

Part 2.1

Below we have, in sequence, a table with:the book file name, character name, the dumping factor of the best configuration, the exponent for our best configuration, number of character inside the community belonging Baratheon, Lannister, Stark, Targaryen family and the total number of character inside the community.

Books	Char.	Dump. Fac.	Expo.	Cond.	Barath.	Lan.	Stark.	Targ.	# of Characters
book_1	Daenerys-Targaryen	0.95	1.0	0.078	0	0	0	3	22
book_1	Jon-Snow	0.95	1.0	0.079	6	6	11	5	166
book_1	Samwell-Tarly	0.95	1.0	0.079	6	6	11	5	166
book_1	Tyrion-Lannister	0.95	1.0	0.079	6	6	11	5	166
book_2	Daenerys-Targaryen	0.95	0.8	0.099	0	0	0	3	18
book_2	Jon-Snow	0.95	0.8	0.085	0	0	1	4	28
book_2	Samwell-Tarly	0.95	0.6	0.085	0	0	1	4	28
book_2	Tyrion-Lannister	0.95	1.0	0.099	8	6	5	6	160
book_3	Daenerys-Targaryen	0.95	0.8	0.071	0	0	0	2	25
book_3	Jon-Snow	0.95	1.0	0.062	0	0	1	1	74
book_3	Samwell-Tarly	0.95	1.0	0.061	0	0	1	1	71
book_3	Tyrion-Lannister	0.95	1.0	0.079	7	9	10	10	204
book_4	Daenerys-Targaryen	0.95	1.0	0.072	5	8	4	11	298
book_4	Jon-Snow	0.95	1.0	0.061	2	0	6	2	181
book_4	Samwell-Tarly	0.95	1.0	0.086	2	0	6	2	177
book_4	Tyrion-Lannister	0.95	1.0	0.044	5	8	4	11	286