Recommender Systems for suggesting the financial reports to users, based on the usage pattern for Finance Data Lake Products

November 6, 2019

Siddhartha Banerjee
BITS 4th Sem Dissertation
2017HT131225

Contents

Problem statement & objectives

Problem Statement

1. There are over 22000+ standard reports & custom reports usage on Data Lake products to cater the needs of GE wide user base across all business segments.

2. A user must depend on book marking reports or remember the name of the report in order to use the same or select from standard catalogue

3. The problem statement was to overcome the explosive growth of reports where each user was trying to create a customized version of an existing report or creating a custom version of the standard report or was not knowing about existence of various other functional reports.

4. The project aims to overcome this issue through effective recommendation to encourage the usage of existing reports & make them known to user groups through effective recommendation

Objectives

1. Implement logging of reports usage in order to analyze usage patterns

2. Design & develop the recommendation engine for the usage of the 3 products based on historical usage pattern

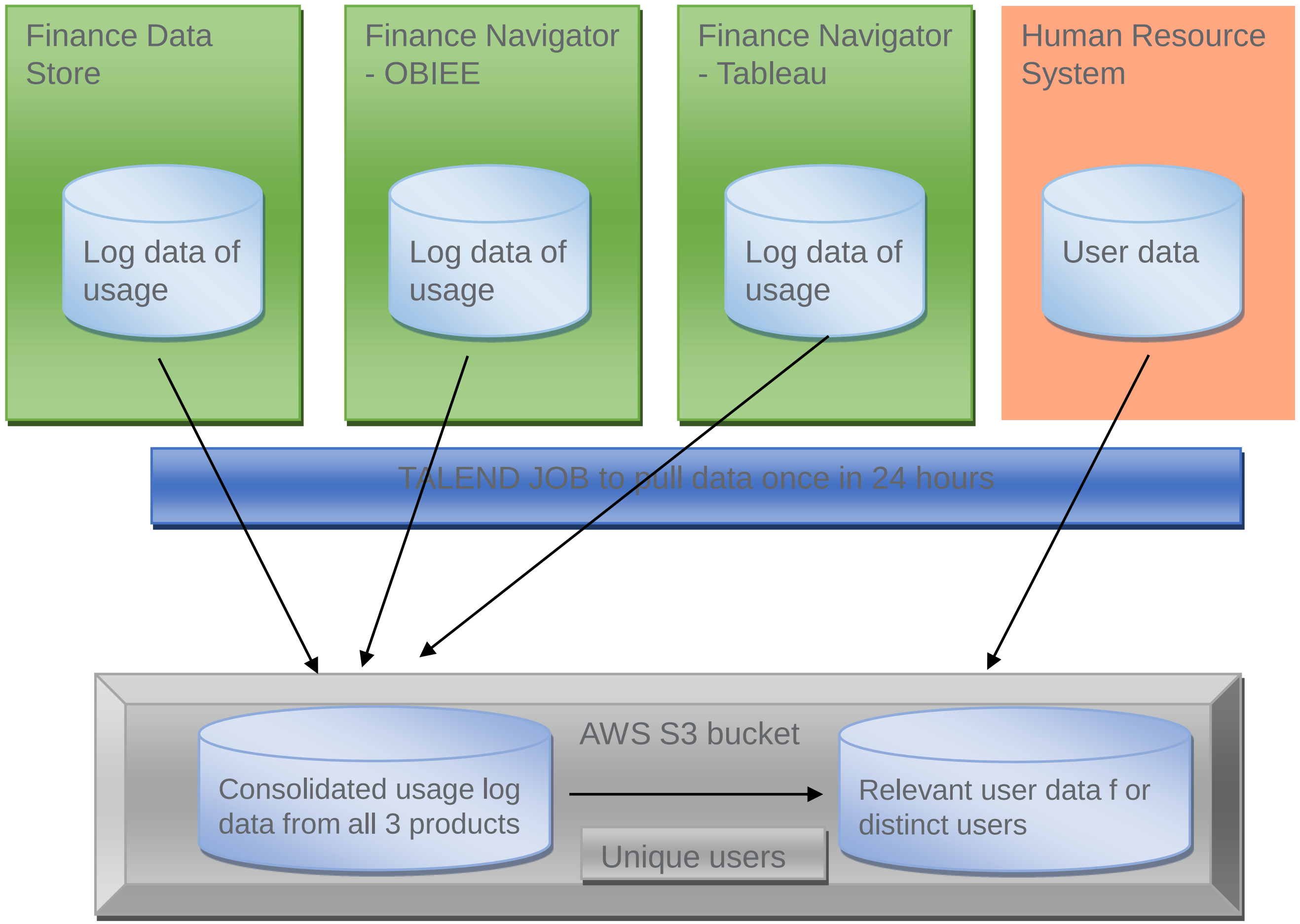
3. Build recommendation systems using 3 different approach
    1. Content based recommendation – For recommending similar reports to what is being used
    2. User profile-based recommendation – For recommending similar reports to what vertical & horizontal peers have used
    3. Collaborative based recommendation – For recommending reports which is most apt for the specific user

Data collation for model generation

Steps to get the required data

1. 3 reporting products were considered for recommendation project.
2. The 3 products were chosen as they were critical for business users and most used
3. Each of 3 products had a logging mechanism to generate the usage logs for the users.
4. Each session is logged where the User-Product-Report_name combination is unique.
5. Talend ETL/API pull/Cron jobs were used to ingested & store the data on a common S3 bucket for aggregation
6. Employee information is extracted from HR systems
7. The employee detailed information is extracted from HR table for the unique users that are present in the reports for user profiling

High level Architecture



Data preprocessing

Clean up data

1. The report data was cleansed to eliminate report names that had words like “Test” or “Try” or “Obsolete” or “Backup” to ensure that only active production reports are being used for model recommendation.
2. Users who have left the company or are no longer active, their usage records were deleted to ensure only active current user base is considered
3. Historical data of Jan 2019-June 2019 was considered for mode. Earlier data was not considered to ensure the recommendation is relevant to the current users and was being done on the recent usage history

Min-Max scaling

1. The total access count of a report is similar to rating of a movie. This number had lot of variance, and was needed to be scaled down for model effectiveness.
2. The access count varied from 1 to 754, so it was required to be scaled down to from 1 to 5

BEFORE SCALING

count	10707.000000
mean	6.194732
std	17.493825
min	1.000000
25%	1.000000
50%	3.000000
75%	7.000000
max	754.000000

AFTER SCALING

count	10707.000000
mean	1.447785
std	0.713293
min	1.000000
25%	1.000000
50%	1.195122
75%	1.585366
max	5.000000

score_df - DataFrame

Index	user_sso	report_id	total_access	score
14660	[redacted]	545989877	5	1.39024
14661	[redacted]	545989877	1	1
14662	[redacted]	545989877	41	4.90244
14663	[redacted]	545989877	2	1.09756
14664	[redacted]	581898547	1	1
14665	[redacted]	823153395	1	1
14666	[redacted]	823302349	3	1.19512
14667	[redacted]	823302349	1	1
14668	[redacted]	823302349	37	4.5122
14669	[redacted]	823302349	1	1
14670	[redacted]	823302350	6	1.4878

Content based recommendation

Content based similarity examines the current content of the articles or reports preferred by the users. The recommender systems then attempt to identify similar reports or articles that was preferred by the user.

Model used

Bag of word model

The report name is combined with all the relevant parameters or features to create a combined name. So, the first step is to generate tokens out of the combined report name or document name along with its relevant attributes.

Cosine similarity

1. Cosine similarity is the COS of the angle between 2 vectors. In other words, it is projection of one vector over the other.
2. The cosine similarity generates a matrix of 1981 X 1981 where each unique Product-Report name is compared with every other Product-Report name. Sample Cosine matrix generated is as follows

The matrix generated gives the score of each report to all other reports. The scores are sorted in descending order to pick the top 5 reports that match closely

report_cosine_similar - NumPy array

	0	1	2	3	4
0	1	0.353553	0.408248	0.5	0.5
1	0.353553	1	0.288675	0.353553	0.353553
2	0.408248	0.288675	1	0.408248	0.408248
3	0.5	0.353553	0.408248	1	1
4	0.5	0.353553	0.408248	1	1
5	0.408248	0.288675	0.333333	0.816497	0.816497
6	0.408248	0.288675	0.333333	0.816497	0.816497
7	0.408248	0.288675	0.333333	0.816497	0.816497
8	0.408248	0.288675	0.333333	0.816497	0.816497
9	0.5	0.353553	0.408248	0.5	0.5
10	0.408248	0.288675	0.333333	0.408248	0.408248
11	0.316228	0.223607	0.258199	0.316228	0.316228
12	0.353553	0.25	0.288675	0.353553	0.353553

Output

candidate_df - DataFrame

Index	report_id	candidate_id	similarity	report_name	candidate_name	rank
4	0	1024	0.894427	FNOBIEE BALANCES ANALYSIS - YTD	FNOBIEE BALANCES ANALYSIS - YTD-CC_NC5025	1
2	0	136	0.75	FNOBIEE BALANCES ANALYSIS - YTD	FNOBIEE BALANCES ANALYSIS - QTD	2
3	0	67	0.75	FNOBIEE BALANCES ANALYSIS - YTD	FNOBIEE BALANCES ANALYSIS - PTD	3
1	0	1560	0.707107	FNOBIEE BALANCES ANALYSIS - YTD	FNOBIEE YTD	4
0	0	1105	0.57735	FNOBIEE BALANCES ANALYSIS - YTD	FNOBIEE AHCM BALANCES P&L	5
6	1	1664	0.666667	FNOBIEE BANK BALANCE	FNOBIEE TRIAL BALANCE - D&D AS	1
7	1	1138	0.666667	FNOBIEE BANK BALANCE	FNOBIEE BALANCE ANALYSIS	2
8	1	954	0.666667	FNOBIEE BANK BALANCE	FNOBIEE TRIAL BALANCE - D&D	3
9	1	56	0.666667	FNOBIEE BANK BALANCE	FNOBIEE TRIAL BALANCE	4
5	1	1615	0.57735	FNOBIEE BANK BALANCE	FNOBIEE BANK DATA POD	5
14	2	3	1	FNOBIEE BASIC	FNOBIEE BASIC C&R	1
10	2	5	0.816497	FNOBIEE BASIC	FNOBIEE BASIC C&R UNDIS	2
11	2	4	0.816497	FNOBIEE BASIC	FNOBIEE BASIC C&R NET_PAY	3
12	2	38	0.816497	FNOBIEE BASIC	FNOBIEE MY_JE_EXTRACT - BASIC	4
13	2	6	0.816497	FNOBIEE BASIC	FNOBIEE BASIC C&R UNDIS.	5

1. The above outputs shows the top 5 named matching report to each of the report that the user uses.
2. This will be used to give the recommendation for similar reports that the user uses

User profile based collaborative recommendation

Cluster 1 -Inter business

1. The attributes of Band of the employee, Primary Business & the Sub-business is used to cluster user groups.
2. This type of clustering assumes that users within a business would have similar usage patterns.
3. This clustering used for recommendations prioritizes usage pattern within the same business for the recommendation.

bu - DataFrame

Index	corp_bnd	level_2	level_1	report_id	total_access	rank_in_bucket
1902	DEFAULT	DEFAULT	GE Renewable Energy	1438629676	120	1
1828	DEFAULT	DEFAULT	GE Renewable Energy	743710267	114	2
1862	DEFAULT	DEFAULT	GE Renewable Energy	1017574377	101	3
1846	DEFAULT	DEFAULT	GE Renewable Energy	864000368	75	4
1847	DEFAULT	DEFAULT	GE Renewable Energy	864000369	75	5
3877	LPB	Power Steam Power	GE Power	1961483755	11	1
3876	LPB	Power Steam Power	GE Power	889872928	1	2
3878	LPB	Power Steam Power	GE Power	2037707659	1	3
2430	DEFAULT	Power Steam Power	GE Power	1961483755	11	1
2429	DEFAULT	Power Steam Power	GE Power	889872928	1	2
2431	DEFAULT	Power Steam Power	GE Power	2037707659	1	3
3875	SPB	Power Power Services	GE Power	1961483755	11	1
3873	SPB	Power Power Services	GE Power	24557354	1	2
3874	SPB	Power Power Services	GE Power	1048734042	1	3
3859	PB	Power Power Services	GE Power	1101896797	219	1
3856	PB	Power Power Services	GE Power	1017574377	125	2
3851	PB	Power Power Services	GE Power	864000368	111	3
3852	PB	Power Power Services	GE Power	864000369	111	4
3858	PB	Power Power Services	GE Power	1080507679	111	5
3835	No Source Data	Power Power Services	GE Power	1654374459	195	1
3823	No Source Data	Power Power Services	GE Power	1017574377	194	2
3825	No Source Data	Power Power Services	GE Power	1101896797	186	3
3817	No Source Data	Power Power Services	GE Power	864000368	160	4
3818	No Source Data	Power Power Services	GE Power	864000369	160	5

Cluster 2 - Intra business

1. The attributes of the Band, Job function & Job family is used to cluster user groups.
2. This type of clustering assumes that the users belonging to a job function & job family across the business has similar usage pattern.
3. This clustering used for recommendations prioritizes usage pattern across business for similar job roles

process - DataFrame

Index	corp_bnd	level_2	level_1	report_id	total_access	rank_in_bucket
1412	SPB	Controllership	Finance	849046182	64	1
1408	SPB	Controllership	Finance	788385311	61	2
1479	SPB	Controllership	Finance	1961483755	57	3
1423	SPB	Controllership	Finance	1017574377	54	4
1375	SPB	Controllership	Finance	24557354	52	5
1279	PB	Controllership	Finance	1101896797	2026	1
1267	PB	Controllership	Finance	1017574377	1300	2
1341	PB	Controllership	Finance	1654374459	1119	3
1242	PB	Controllership	Finance	864000369	1079	4
1276	PB	Controllership	Finance	1080507679	1079	5
1125	OTHSAL	Controllership	Finance	1017574377	130	1
1132	OTHSAL	Controllership	Finance	1101896797	79	2
1118	OTHSAL	Controllership	Finance	864000368	73	3
1119	OTHSAL	Controllership	Finance	864000369	73	4
1131	OTHSAL	Controllership	Finance	1080507679	73	5
1066	No Source Data	Controllership	Finance	1017574377	137	1
1058	No Source Data	Controllership	Finance	864000368	109	2
1059	No Source Data	Controllership	Finance	864000369	109	3
1070	No Source Data	Controllership	Finance	1080507679	109	4
1088	No Source Data	Controllership	Finance	1654374459	107	5

Collaborative recommendation – SVD method

The Single Value Decomposition (SVD) model help decompose the usage matrix into latent vectors which can be used for recommendation

Model output

The recommended data frame generated is as below. We take only the top 5 recommendation to be displayed to the end user.

recommendation_df_svd - DataFrame

Index	user_sso	report_id	score_estimate	rank
19	[redacted]	1242064080	1.99344	1
749	[redacted]	597286461	1.96182	2
349	[redacted]	1351390390	1.93345	3
357	[redacted]	1725955002	1.9173	4
430	[redacted]	1493663810	1.91547	5
4324	[redacted]	1725955002	1.82725	1
4143	[redacted]	414251741	1.81992	2
4194	[redacted]	849046182	1.80886	3
4137	[redacted]	1364401219	1.76691	4
3986	[redacted]	1242064080	1.76653	5
1112235	[redacted]	1654374459	2.19199	1
1112936	[redacted]	597286461	2.06295	2
1112360	[redacted]	1932582669	2.05983	3
1112538	[redacted]	1375497469	2.02267	4
1112764	[redacted]	957029259	1.95592	5
1116215	[redacted]	1017574377	2.0458	1
1116218	[redacted]	1654374459	1.97643	2
1116521	[redacted]	1375497469	1.90622	3
1116919	[redacted]	597286461	1.89685	4
1116331	[redacted]	418729141	1.87594	5

Model performance

5 fold cross validation is run to assess the performance of the model. . We get a mean RMSE of 0.6059 & mean MAE of 0.3885

```
score = cross_validate(algo_svd, svd_data, measures=['RMSE', 'MAE'], cv=5,
                       verbose=True)
Evaluating RMSE, MAE of algorithm SVD on 5 split(s).

                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std
RMSE (testset)    0.6408  0.6169  0.6193  0.5588  0.5938  0.6059  0.0279
MAE (testset)     0.4072  0.3872  0.3930  0.3685  0.3867  0.3885  0.0125
Fit time          0.56    0.78    0.53    0.68    0.80    0.67    0.11
Test time         0.01    0.01    0.01    0.02    0.02    0.02    0.00
```

Collaborative recommendation – SVD++ method

Single Value Decomposition ++ (SVD++) in addition to core SVD, also takes into account the user local preference and works well on sparse matrix.

Model output

The recommended data frame generated is as below. We take only the top 5 recommendation to be displayed to the end user.

recommendation_df_svdpp - DataFrame

Index	user_sso	report_id	score_estimate	rank
435	[redacted]	512256810	1.87936	1
349	[redacted]	1351390390	1.87815	2
529	[redacted]	1150722472	1.85347	3
577	[redacted]	957029259	1.84585	4
749	[redacted]	597286461	1.83816	5
4716	[redacted]	597286461	2.07184	1
3986	[redacted]	1242064080	1.93399	2
4316	[redacted]	1351390390	1.92054	3
4022	[redacted]	1473357027	1.88211	4
4402	[redacted]	512256810	1.8198	5
1112617	[redacted]	1493663810	2.14777	1
1112206	[redacted]	1242064080	2.14647	2
1112936	[redacted]	597286461	2.13571	3
1112764	[redacted]	957029259	2.09994	4
1112232	[redacted]	1017574377	2.03789	5
1116919	[redacted]	597286461	2.06106	1
1116699	[redacted]	1150722472	1.98174	2
1117109	[redacted]	257315482	1.89971	3
1116225	[redacted]	1473357027	1.84433	4
1116407	[redacted]	468633286	1.82787	5

Model performance

5 fold cross validation is run to assess the performance of the model. . We get a mean RMSE of 0.5781 & mean MAE of 0.3602

```
score = cross_validate(algo_svdpp, svd_data, measures=['RMSE', 'MAE'], cv=5,
                       verbose=True)
Evaluating RMSE, MAE of algorithm SVDpp on 5 split(s).

                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std
RMSE (testset)    0.5869  0.5625  0.5472  0.5715  0.6225  0.5781  0.0257
MAE (testset)     0.3732  0.3530  0.3474  0.3498  0.3775  0.3602  0.0126
Fit time          2.53    2.48    2.44    2.63    2.65    2.55    0.08
Test time         0.06    0.05    0.06    0.06    0.06    0.06    0.00
```

Collaborative recommendation – NMF method

The Non Negative Matrix Factorization (NMF) model help decompose the usage matrix weight & component latent vectors which can be used for recommendation

Model output

The recommended data frame generated is as below. We take only the top 5 recommendation to be displayed to the end user.

recommendation_df_nmf - DataFrame

Index	user_sso	report_id	score_estimate	rank
939	[redacted]	257315482	2.49657	1
537	[redacted]	1449349248	2.26921	2
237	[redacted]	468633286	2.2593	3
749	[redacted]	597286461	2.25919	4
577	[redacted]	957029259	2.03637	5
4204	[redacted]	468633286	3.45253	1
4906	[redacted]	257315482	3.12865	2
4716	[redacted]	597286461	3.09678	3
4504	[redacted]	1449349248	2.81755	4
4544	[redacted]	957029259	2.63825	5
1112424	[redacted]	468633286	3.0142	1
1112936	[redacted]	597286461	3.00128	2
1113126	[redacted]	257315482	2.90694	3
1112764	[redacted]	957029259	2.54127	4
1112245	[redacted]	189755771	2.35848	5
1116169	[redacted]	823153395	1.44725	1
1116172	[redacted]	1602502046	1.44725	2
1116238	[redacted]	780707070	1.44725	3
1116267	[redacted]	1277810363	1.44725	4

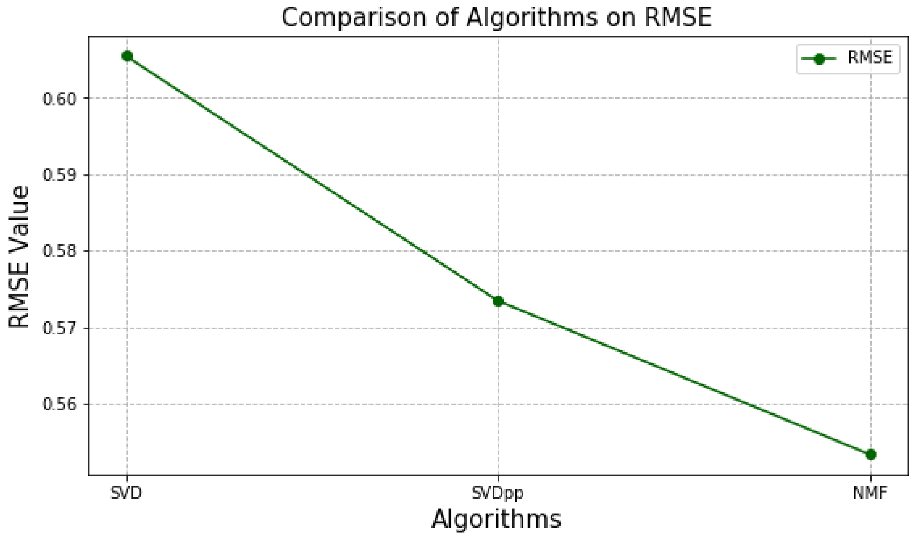
Model performance

5 fold cross validation is run to assess the performance of the model. . We get a mean RMSE of 0.5562 & mean MAE of 0.2931

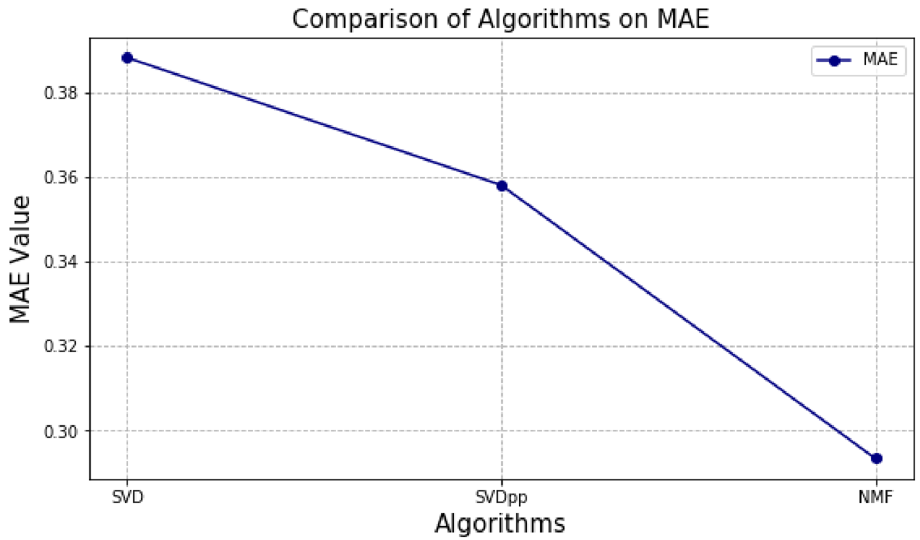
```
score = cross_validate(algo_nmf, svd_data, measures=['RMSE', 'MAE'], cv=5,
verbose=True)
Evaluating RMSE, MAE of algorithm NMF on 5 split(s).

                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std
RMSE (testset)    0.5515  0.5213  0.5481  0.5939  0.5660  0.5562  0.0238
MAE (testset)     0.2884  0.2782  0.2860  0.3125  0.3002  0.2931  0.0120
Fit time          0.63    0.65    0.70    0.71    0.65    0.67    0.03
Test time         0.01    0.01    0.01    0.01    0.01    0.01    0.00
```

Performance measure – SVD / SVD++/NMF

Root Mean Square Error



Mean Absolute Error



1. We observe that the NMF (Non negative Matrix Factorization) has the better performance in both RMSE (Root Mean Square Error) & MAE (Mean Absolute Error) as it has least error.

2. NMF performs better as it uses gradient descent function find the optimal matrix decomposition for recommendation

3. Therefore as a conclusion we move forward with NMF (Non Negative Matrix Factorization) recommendation to move ahead.

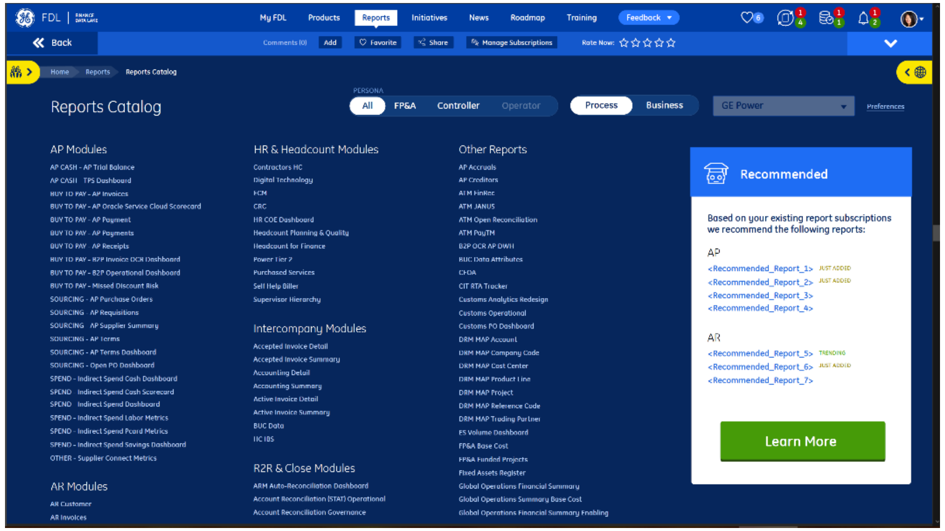
Output integration for user display (Work in Progress)

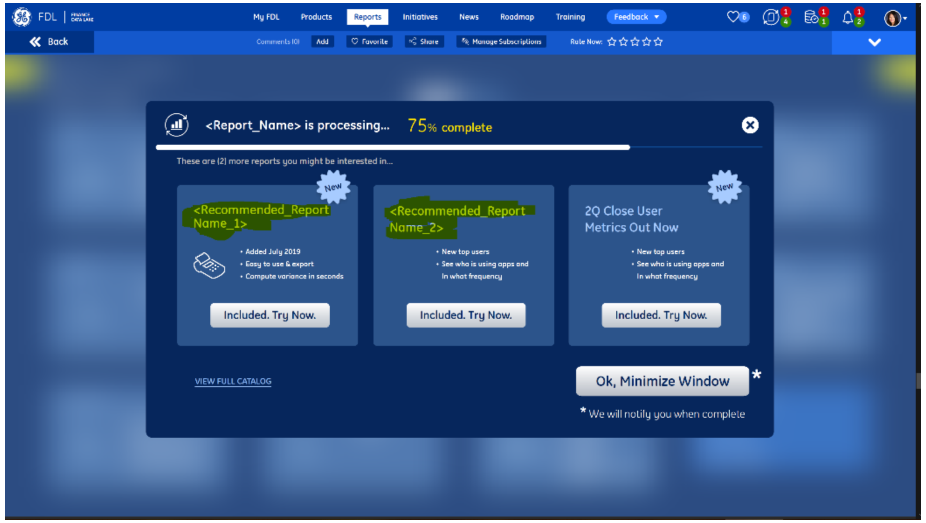
Content & Profile based recommendation

The content based and user profile based collaborative recommendation would be displayed on the right pane when the user is browsing through the reports. The recommendation would be related to the report he is currently browsing.

Collaborative recommendation using matrix factorization

The collaborative recommendation to be displayed on the main console of the FDL Wrapper page. The top 5 recommended reports for the user would be displayed on the console while his current report is being run.





Conclusion

Summary

This analysis & the recommendation engine helps in 2-fold purpose

1. Users recommendation- Core purpose of recommendation of reports to user base so that they discover relevant reporters that their peer have used within the business & across the business, thereby eliminating manual work of replicating the report functionality. This also helps user adoption of the products.

2. Product usage analysis– The underlying data of the recommendation engine helps the product managers in the following aspect - Identify patterns of reports usage across various user profiles thereby cross training & socializing relevant reports & Identify reports that are sparsely to analyze the fitment of the functionality

Future scope of work

The future scope of work would be integration of all the 3 recommendation and display recommendation reports to user at various stages of the product wrapper usage.

1. Generate recommendation using live data on a weekly basis
2. Measure the hit ratio of the recommended reports to ensure reuse & collaboration of reports
3. Optimize the recommendation model using neural networks to provide customized recommendation for each user
4. Identify trends & change in usage patterns of users to predict the most suitable report at that point of time or at that region.
5. Incorporate other attributes in the recommendation algorithm to make more effective recommendation with higher hit ratio