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

BITS ZG628T: Dissertation

by

Siddhartha Banerjee

2017HT13125

Dissertation work carried out at

GENERAL ELECTRIC Company, Bangalore

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE
PILANI (RAJASTHAN)

November 2019

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

BITS ZG628T: Dissertation

by

Siddhartha Banerjee

2017HT13125

Dissertation work carried out at

GENERAL ELECTRIC Company, Bangalore

Submitted in partial fulfillment of MTech. Software Systems degree programme

Under the Supervision of
Ramji Sarangarajan
General Electric Company, Bangalore

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE
PILANI (RAJASTHAN)

November 2019

CERTIFICATE

This is to certify that the Dissertation entitled ***Recommender Systems for suggesting the financial reports to users based on the usage pattern for Finance Data Lake Products*** and submitted by **Siddhartha Banerjee** having ID-No. **2017HT13125** for the partial fulfillment of the requirements of M.Tech. Software Systems – Data and Analytics degree of BITS, embodies the bonafide work done by him/her under my supervision.

Ramji Sarangarajan

Place: BANGALORE

Date: NOV 4, 2019

General Electric Company, Bangalore

Birla Institute of Technology & Science, Pilani

Work-Integrated Learning Programmed Division

First Semester 2019-2020

BITS ZG628T: Dissertation

ABSTRACT

BITS ID No.	**: 2017HT13125**
NAME OF THE STUDENT	**: SIDDHARTHA BANERJEE**
EMAIL ADDRESS	**: banerjee.siddhartha.sb@gmail.com**
STUDENT'S EMPLOYING ORGANIZATION & LOCATION	**: GENERAL ELECTRIC Company, Bangalore**
SUPERVISOR'S NAME	**: Ramji Sarangarajan**
SUPERVISOR'S EMPLOYING ORGANIZATION & LOCATION	**: GENERAL ELECTRIC Company, Bangalore**
SUPERVISOR'S EMAIL ADDRESS	**: ramji.sarangarajan@ge.com**
DISSERTATION TITLE	: Recommender Systems for suggesting the reports to business users based on the Data Lake usage pattern

ABSTRACT : Finance Data Lake is aggregation of all the financial data from various systems of General Electric company across 5 business. There are multiple products that provide standardized reporting that caters to the business user needs. The recommenation engine will leverage usage metrics from 3 top used products to build the model, as these products are standard reporting platform.

The project is aimed at providing a customized recommended set of reports to business users across the 3 products that are most likely to be suitable for their requirements. The recommendation engine will recommend based on 3 distinct algorithms namely — Content based filtering based on past usage, Collaborative filtering based on similar usage & user profile similarity based for grouping users.

The recommendation engine will leverage the histoircal usage data of Q2'19 & Q3'19 to generate the model. The recommendation model will lay the path for integration of the 3 models by adjusting the weightage of the 3 model to generate a hybrid recommendation model leveraging the recommendation output from each of them. This integration will be based on the effective usage of the recommended set of reports.

The recommendation output apart from helping the users, will also be leveraged by the product managers & the sponsors to determine the effective utlization of the reproting platform and take proactive actions for underutilized reports or products.

Broad Academic Area of Work: Data mining & Information retrival

Key words : Recommender systems, Content based filtering, Collaborative filtering, Hybrid, document similarity, matrix factorization, co-occurrence matrix

Siddhartha Banerjee

Signature of the Student
Name: Siddhartha Banerjee

Date: 4-Nov-19
Place: Bangalore

Signature of the Supervisor
Name: Ramji Sarangarajan

Date: Nov 4, 2019
Place: BANGALORE

Follow up from mid-sem evaluation

1. **What is the overhead due to the logging process proposed?**

Out of the 3 products, 2 of them were standard tech stack. Finance Navigator ***OBIEE*** was based on Oracle Business Intelligence as result it had implicit logging of the usage. Finance Navigator ***Tableau*** also had implicit logging of user data. The 3rd application was Web based application and it leveraged DENODO for query transformation. DENODO also allowed logging of all queries. So, this was enabled thereby addressing the 3rd product. Thus, all the 3 products had provision for logging which was either already enabled OR had to be enabled.

This prevented creating any overhead as logging was being done by the applications after they had submitted their query to the database and was in a wait state for the results to return. So overhead was avoided as logging was done the during wait state. The logging involved writing just 1 line of record for – User -Product-Report Name-Usage count combination to local DB, thereby taking minimal time & resource.

The aggregation of the data on S3 bucket was a nightly process, and only incremental records where extracted. This mean only records that were created in past 24 hours or had changed past 24 hours was extracted. This reduced the overall volume of the data pull which in turn reduced network latency.

2. **Can you mention the use-cases for the three models you propose?**

There is wireframe design on the use case on page 33 on 2 of the use case of the recommendation. Here is the rationale of the various recommendation models:

Content based recommendation – This will be recommended to users when they directly go to a named report or access a bookmarked link which takes them straight to a specific report. This recommendation will be displayed to users to notify them of the top 5 similar reports to what they just accessed.

Profile based collaborative recommendation – There is dedicated section where we display the reports accessed by their peers or similar job profiles, and state that they may find it useful. This will encourage collaboration & reuse of reports. This will be effective for folks

that visit the landing page and explore various reports. (As this is deployed on production, taking screen shot or image is restricted).

Collaborative recommendation – This will be recommended to the user group when any of the report is submitted and the screen is waiting for the results back from DB. Thus, this recommendation is independent of the report being used, and is purely depend on the collaborative recommendation. The use case is to showcase to the user the most recommended reports.

3. **For Figures, numbering and labels must be as per the BITS format. And a Fig must be referred to by the main text of report.**

Yes, I agree. I have updated the same. Please let me know if this adheres to the BITS format.

4. **How do you handle formatting differences while extracting data from different sources?**

The common elements were extracted from each of the product, thus there was no need for immediate formatting of the data.

1. *User name* – This is standard SSO (Single Sign On ID) or the employee ID
2. *Product Name* – This is a standard text, and each of the product had their own respective name
3. *Report name* – This is also a standard text of the standard reports that are deployed on the products
4. *Usage count* – This was a numeric count field that was common to all reports
5. Other parameters like the report path, report domain, report link..etc. were not considered for the first phase.

5. **What if the user preferences change at later point of time? How do you handle the situation?**

The recommendation is not static. The recommendation will be computed on a weekly basis based on past 6 months of data. The logging is being done on a continuous basis. So, on a weekly frequency, we will be executing all the algorithms on the last 6 months data to compute the new recommendation. This will also consider new reports are being added to the products.

Table of Contents

List of Figures (with figure number, figure titles and page numbers)

List of Tables with table number, table title and page number

1. Introduction

1.1. Overview

Recommendation systems are widely used in commercial world and entertainment industry. These systems recommend possible choices to the users that helps the user to discover items or entertainment of their choice, which otherwise would take multiple iterations or search to find. Thus, recommendation system helps the user find items as per their preference, and helps the commercial company increase the sales or hit count of their products offerings.

On similar lines, this project aims at implementing recommendation systems for the Finance Data Lake to recommend the various products & their reports to cater the needs of the user base. The user base is varied and is spread across multiple business units, job functions & geographical locations.

This recommendation system will help the user find all the relevant reports across 3 products that are most relevant to their job function. The recommendation project also aims at increasing the usage of the Finance Data Lake products there by increasing the usage of report offering across the products.

The idea was conceived based on the proven fact that Amazon or Netflix increased their sales and usage by introducing the recommendation of products/entertainment for the users.

1.2. Problem statement

Finance Data Lake has 3 distinct reporting platforms – Finance Data Store, Finance Navigator – OBIEE & Finance Navigator – Tableau. There are over 22000+ standard reports & custom reports usage on these products to cater the needs of GE wide user base across all business segments & geographical regions.

A user must depend on book marking reports or remember the name of the report in order to use the same. This results in skewed usage of reports that are either most common or standard. This practice also hinders the sharing of user report where a custom report

created by a user in a geographical region can be used & known to another user in different geographical region.

In order to bridge the gap, the current process relies heavily on manual user training & encouraging users to explore existing reports before creating a new one.

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

1.3.Objective

The objective is that that the recommendation system will recommend reports to the users using 3 different methods to suit the need of the diverse user base. The project objectives are as follows:

a) Implement logging of reports usage in order to analyze usage patterns
b) Design & develop the recommendation engine for the usage of the 3 products
c) Build recommendation systems using 3 different mechanism
    a. Content based recommendation
    b. User profile-based recommendation
    c. Collaborative based recommendation
d) Performance measure for the Collaborative recommendation
e) Use case for each of the 3 methods of recommendation

1.4. Out of scope of work

The out of scope for the project includes the following:

a) Integration of the recommendation engine with the products wrapper page for display of the recommendation to the users
b) GUI design & development on the wrapper for the recommendation engine

2. Recommender Models

2.1. Content based similarity recommendation

The following reference was used to start building the recommendation models - https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-recommendation-engine-python/ & https://towardsdatascience.com/how-youtube-recommends-videos-b6e003a5ab2f

2.1.1. What is 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. The key factor here is if a user has liked or preferred 50 reports or documents, not all the reports have equal preference. So, for the content-based recommender system to work, the preference of the user needs to be sorted in descending order of most preferred to least preferred report/document. This sorting needs to be done based on rating.

So, rating of a document by a user is another key factor that determines the effectiveness of the content-based recommender system. Content based filtering considers only the highly rated document/reports in order to recommend similar reports/documents.

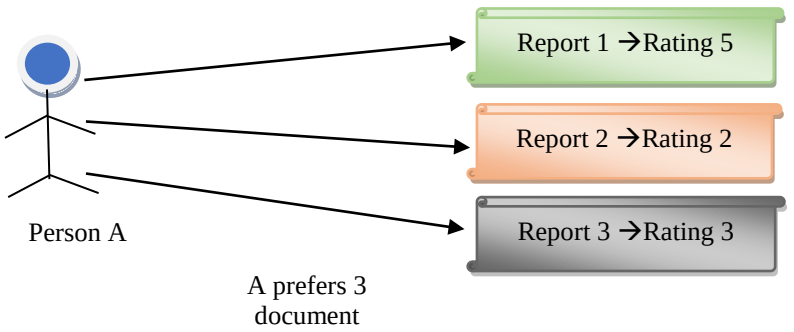


Fig 1 – User preference of report

Here referring to *Fig 1 – User Preference of report,* we see that Document 1 has the highest rating of 5, which is most preferred by the user A. The model extracts the features of the report in order to find the similar reports as showing in *Fig 2-Report properties*. The features can be externally defined or can be embedded in the name of the document nomenclature.

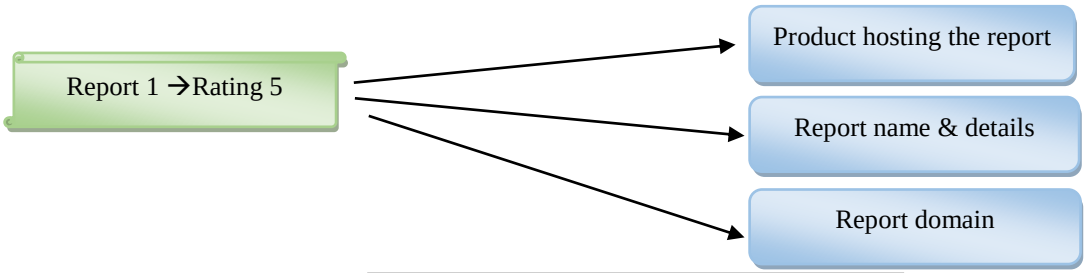


Fig 2 - Report properties

Using the highest rated report for the user, we find similarity with all other candidate reports that exists in the system.

The standard similarity score is cosine similarity. The cosine similarity scores of all the comparison is then sorted in descending order. There after the top 5 cosine score documents are the candidate set for the recommendation.

2.1.2. Mathematical model

The key component of content-based similarity is cosine similarity. There are 2 parts to it

1. **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.

   E.g. – Name of the report is (<product name>< report domain name>< Report functionality name>< Report name 1><report name 2>)

   Unique tokens are generated from each of these individual words for the model to work. English Stop words like THE, AND, AS, etc. are ignored or removed from the token set as they do no generate any meaningful information.

2. Cosine similarity

Cosine similarity is the COS of the angle between 2 vectors. In other words, it is projection of one vector over the other.

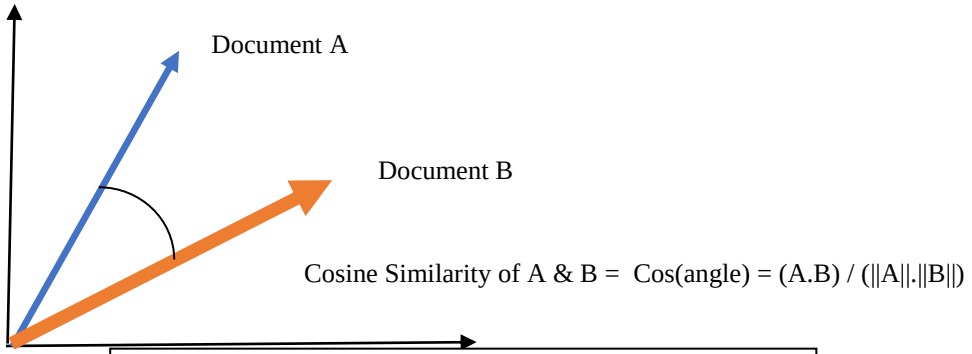


Fig 3 – Cosine similarity representation

The Cosine similarity measures how close or similar 2 vectors are as shown *in Fig 3 - Cosine similarity representation*

The COS value provides a value between 1 & 0.

COS (90) = 0, COS (0) = 1

So, if 2 documents are at 90-degree angle, meaning no similarity, then the similarity score is ZERO. If 2 vectors are same, meaning the angle so Zero, COS (0) = 1. Thus value 1 means they are 100% similar.

2.1.3. Drawbacks

COSINE similarity suffers from the "Cold Start Problem". Meaning if the user does not have historical record of preference or usage, then the algorithm will not work, as we will not have any data to compare or generate the recommendation data. This algorithm suffers from user cold start problem, as we may not have the relevant data for the user to recommend the relevant products.

To overcome the cold start problem whenever it may exist, we use a combination of the recommendation models like Collaborative recommendation & User profile-based recommendation.

2.2.User profiling-based collaborative recommendation

2.2.1. What is user profiling-based collaborative recommendation?

User profiling-based recommendation groups user in various clusters based on their predefined attributes. These clusters can be supervised when the attributes are pre-defined or can be unsupervised when we let the model cluster them as shown in *Fig 4 – User profiling based collaborative recommendation*. Each cluster is then processed to identify the usage pattern of the users within the cluster. Once the usage pattern is identified, the same can be recommended for any new user that belong to that cluster.

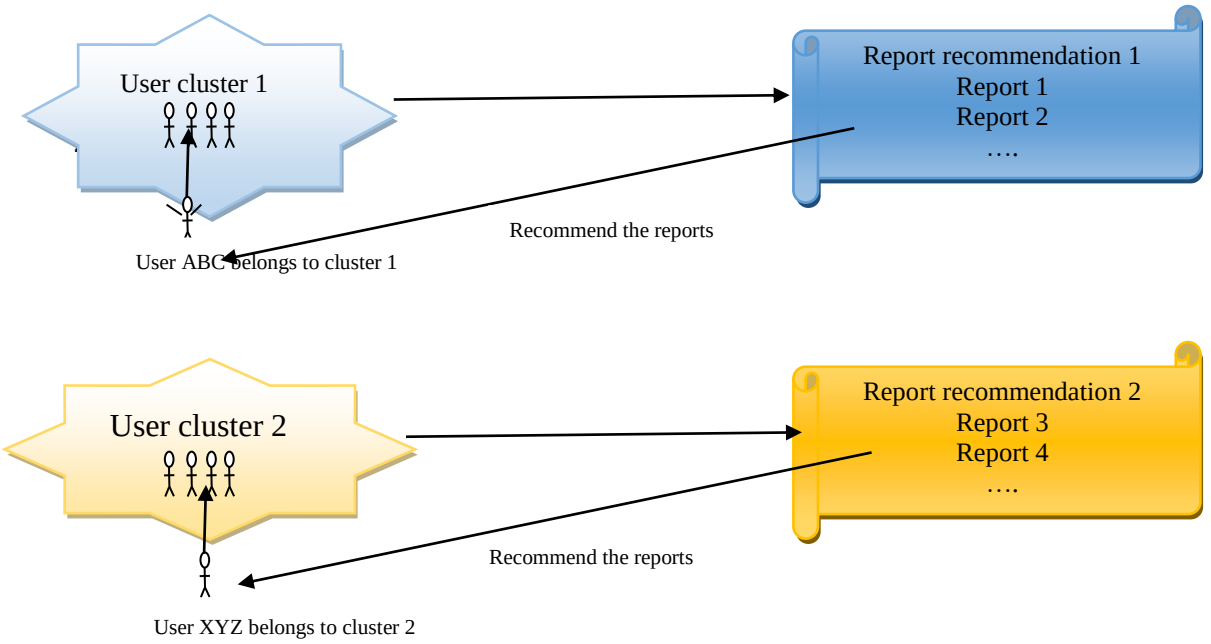


Fig 4 – User profiling based collaborative recommendation

2.2.2. Mathematical model

User profile primarily depends on the clustering of similar attributes for user profiles and mapping the relevant outcomes for recommendations.

f(recommendation) = clustering (attribute 1, attribute 2, attribute 3…attribute n) X recommendation (cluster)

2.3.Collaborative recommendation using matrix factorization (*Post midsem*)

The collaborative filtering was built in reference to - https://www.kaggle.com/gspmoreira/recommender-systems-in-python-101 which outlines the matrix factorization

2.3.1. What collaborative filtering using matrix factorization?

Let us considers the below user to movie ratings as shown in *Fig 5- Collaborative recommendation*. Let's focus on the User 1 & User 5. Both have them have liked movie 1, and movie 4. So, we can somewhat say that they have similar preference. Based on that we can recommend that User 5 will be neutral towards movie 2 & User 1 will love movie 3 as user 5 loved it.

So, here we took 2 users into account. However, for a large data set, we need to consider all possible combination to identify preference & compute the recommendation. This is core of collaborative recommendation.

Here matrix factorization comes to rescue as it tends to extract features and inference from the interaction matrix and reduces the dimensions to more manageable data set for recommendation.

	🕮 Movie 1	☎ Movie 2	☠ Movie 3	✈ Movie 4
User 1	☺ Loved	😐 Neutral	? Not rated	☺ Loved
User 2	😐 Neutral	☺ Loved	☹ Hated	? Not rated
User 3	😐 Neutral	☺ Loved	😐 Neutral	☺ Loved
User 4	? Not rated	? Not rated	☺ Loved	☹ Hated
User 5	☺ Loved	? Not rated	☺ Loved	☺ Loved

Fig 5 – Collaborative recommendation

So, the objective of matrix factorization is to represent the user movie matrix into 2 multipliable matrices A X B which have a lower dimension of rows & columns as shown in *Fig 6-Matrix factorization*. Now each of the 2 matrices will represent 2 different aspect of the user-movie matrix.

Matrix A will have rows as users & columns has Latent Factors, where as Matrix B will have rows as Latent Factors & columns as movie.

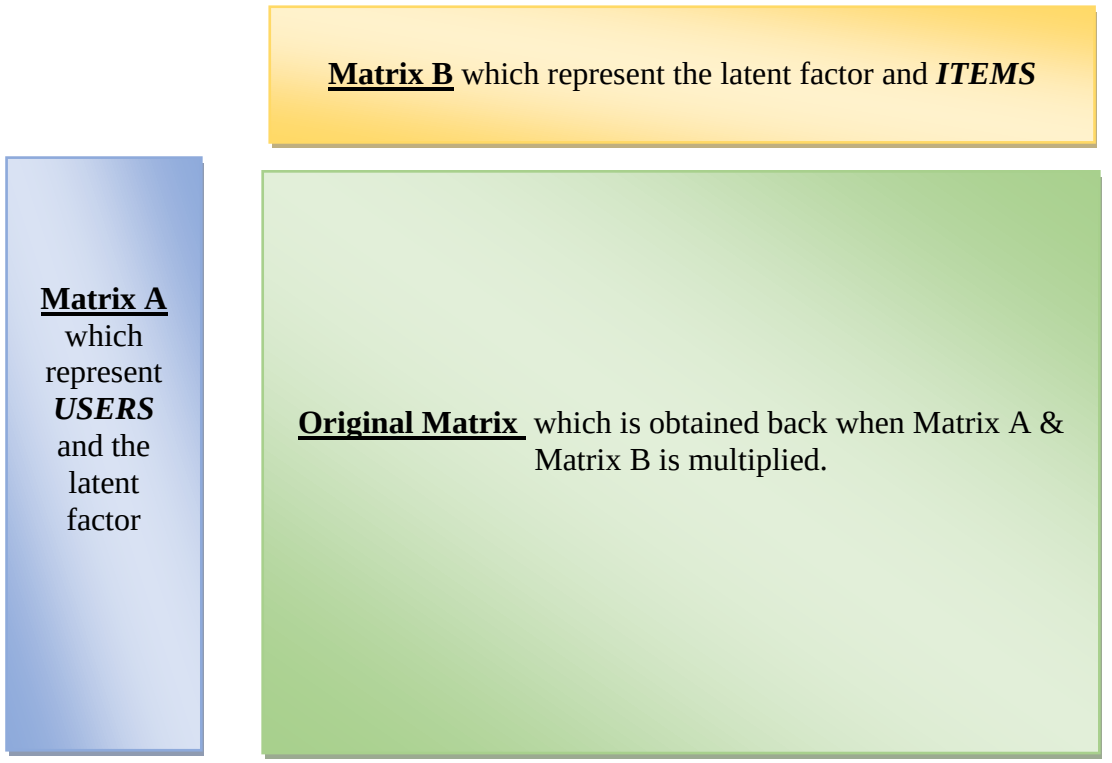


Fig 6 – Matrix factorization

Note – The original matrix is very sparse matrix, as not all user would have rated the movie, so doing the matrix factorization needs to be done in a proper way to find the right latent factors. There are multiple ways to do it, and the right way to measure the most effective solution is through Root Means Square Error (RMSE) comparison OR Mean Absolute Error (MAE).

2.3.2. Mathematical model for 3 types of matrix factorization

2.3.2.1. Singular Value decomposition (SVD)

In SVD we decompose the matrix into 3 parts. Let's us consider Original matrix (X rows & Y columns). We decompose this into 3 multipliable matrices as shown in *Fig 7 – SVD Matrix factorization*

1. **Matrix U -** Left Singular Values with size X rows and M columns. This is vertical matrix will less columns. (M < Y)

2. **Matrix Sigma –** Singular values. This matrix is a diagonal matrix of M rows & M columns
3. **Matrix V –** Right Singular values. This is of size Y rows & M columns

Properties

1. Matrix U, Sigma & V are unique
2. Matrix U & V are column orthogonal. This means that U X U transpose, OR V X V transpose results in an Identify matrix. Also Rows U X Columns of V will result in Zero.
3. Matrix Sigma have values in the diagonal, and they are all positive. They are also sorted in descending order.

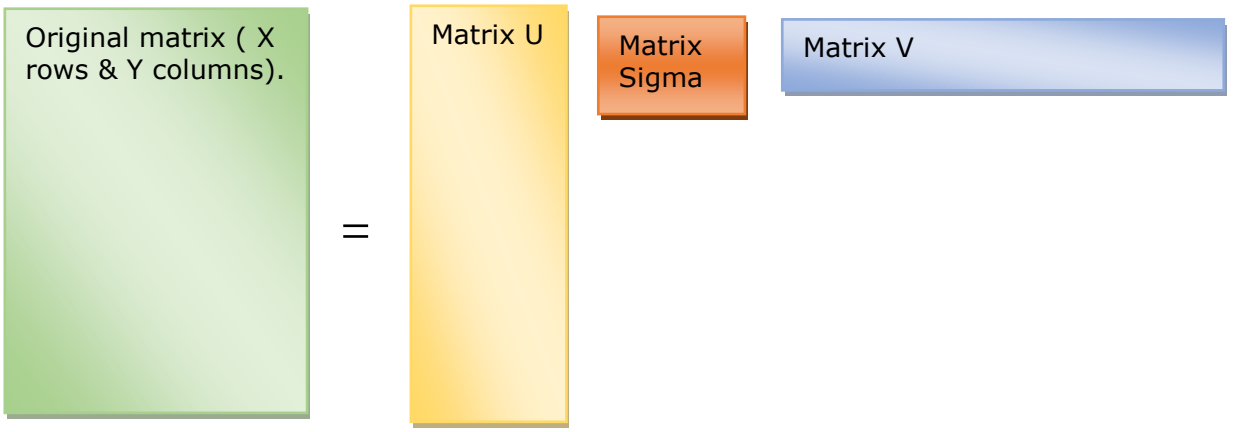


Fig 7 – SVD Matrix factorization

2.3.2.2. Singular Value decomposition with implicit feedback (SVD++)

SVD++ was popularized by the Netflix prize context to improve upon the SVD model.

SVD leverages latent factors to provide recommendation. However, the drawback of the SVD using latent factors are:

1. Association detection for a smaller set of items that maybe closely related to each other are not considered
2. Does not perform well when there is sparse data

SVD++ addressed the above drawbacks by considering implicit feedback of a user, where the user rates an item or uses a report without any consideration or influence of other's usage. So SVD++ leverages inter relationship that are local and closely related.

SVD++ has 4 parts as shown in *Fig 8 – SVD++ factorization*

1. Global mean + User Bias + Item Bias
2. Core SVD in addition to implicit feedback
3. Bias of the rating
4. Implicit feedback and its effect on local rating

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T \left(p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j \right) + |R^k(i;u)|^{-\frac{1}{2}} \sum_{j \in R^k(i;u)} (r_{uj} - b_{uj}) w_{ij} + |N^k(i;u)|^{-\frac{1}{2}} \sum_{j \in N^k(i;u)} c_{ij}$$

Source: http://dparra.sitios.ing.uc.cl/classes/recsys-2015-2/student_ppts/CRojas_SVDpp-PMF.pdf

Fig 8 – SVD++ factorization

2.3.2.3. Non-Negative Matrix factorization (NMF)

Non-Negative Factorization is a very useful in a data that is very sparse, or when the relative properties are distributed. The key criteria as the name suggests, that the original matrix should be non-negative. The original matrix is decomposed into two unique matrix A & B. A form the basis of each item & B represents the basis for data points for each element represented in A. This the Original matrix is decomposed into 2 lower rank matrices which are easy to interpret as shown in *Fig 9 – NMF matrix factorization*

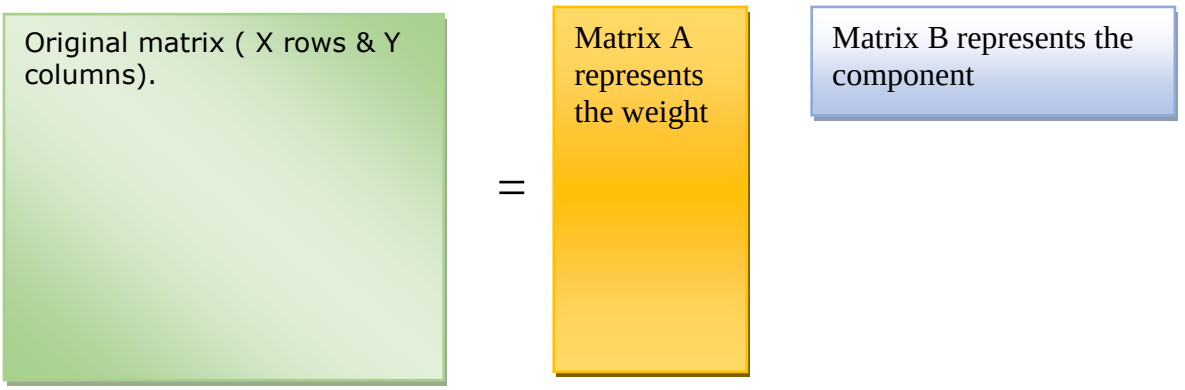


Fig 9 – NMF matrix factorization

As NMF is classified as NP HARD problem and iterative approach is used to formulate the two matrices A & B. Various heuristic methods are used to ensure convergence of NMF. This uses stochastic gradient function to come up with the Matrices A & B.

3. Model Evaluation *(post midsem)*

3.1. Model Evaluation – Root Means Square Error (RMSE) & Mean Absolute Error (MAE)

Model evaluation is done through calculating the difference between the predicted value VS Observed value. The 2 most popular method to do the same is RMSE & MAE. The reference used for model evaluation was - https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-recommendation-engine-python/

3.1.1. What Root Mean Square Error (RMSE)?

RMSE is calculated as Square root of the mean squared difference of observed & predicted value.

RMSE = Square root [{Summation of (Predicted – Observed) square)}/ number of elements]

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (P_i - O_i)^2}{n}}$$

Source - https://gisgeography.com/root-mean-square-error-rmse-gis/

3.1.2. What Mean absolute Error (MAE)?

Mean absolute error as the name suggests is the mean of the absolute difference of predicted & observed value.

MAE = [{Summation of absolute values| (Predicted – Observed) |)}/ number of elements]

$$MAE = \frac{1}{n}\sum_{j=1}^{n} |y_j - \hat{y}_j|$$

Source - https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d

3.2.Which is better RMSE or MAE?

The key difference between RMSE & MAE is that in RMSE we square the difference before doing square root, where as in MAE the difference is not squared. This leads to magnification of large errors in RMSE, where as in MAE there is equal weightage of all errors.

E.g. – Let's consider 2 cases
Case 1 - The (Predicted – Observed) values are 2, 2, 1 for 3 set data

RMSE = Square root of {(4 + 4+ 1) / 3)} = 1.732
MAE = (2+2+1)/3 = 1.67

Case 2 - The (Predicted – Observed) values are 10, 1, 1 for 3 set data

RMSE = Square root of {(100 + 1+ 1) / 3)} = 5.83
MAE = (10+1+1)/3 = 4

So, we observe that the difference between RMSE & MAE is significant when there is large difference in the error. RMSE successfully magnifies the difference.

So, both RMSE & MAE needs to be observed to understand the nature of the error and gap in Observer Vs predicted value. RMSE helps in highlighting large errors, whereas MAE helps in interpreting the error.

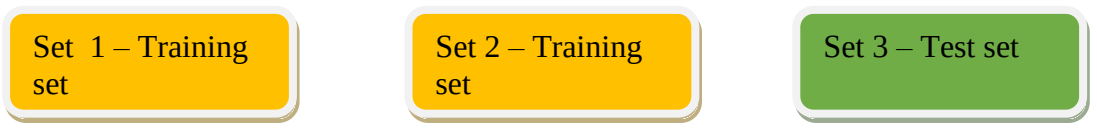
3.3.K fold cross validation for model evaluation

K-Fold cross validation helps in randomizing the data while building the model to eliminate any bias in selecting the training & test data. The following are steps of cross validation process

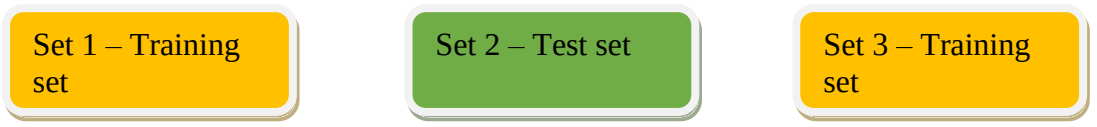
1. Randomize the order of the data
2. Make equal split of the data into k parts
3. For each K parts, make 1 part has Test, and rest k-1 as training set
4. Build the model & train it
5. Repeat step 3 till all parts had a chance to be test set, and part of the training set

Example there are 30 data elements, and we want to do 3-fold cross validation as shown in *Fig 10- 3 -fold cross validation*. Each set is made of 10 data elements.
We do the following:

Iteration 1 – Use Set 1 & 2 for training & set 3 for test



Iteration 2 – Use Set 1 & 3 for training & set 2 for test



Iteration 3 – Use Set 2 & 3 for training & set 1 for test

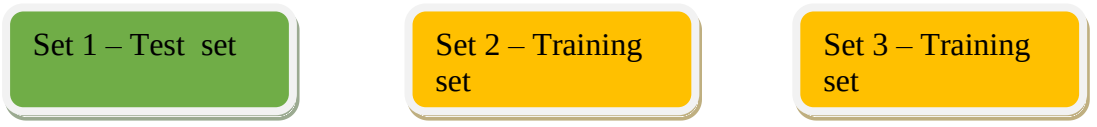


Fig 10 – 3-fold cross validation

4. Data gathering & generation for the project

4.1.Report usage from core products

3 core products were considered for recommendation project. The 3 products were chosen as they were critical for business users and had the top usage among the suite of 10+ products that existed for data consumption & reporting. These products were used globally and addressed all the various functional domain. Also, these products provided the logging of the report usage, thereby providing the feasibility to build the analytical model for their recommendation.

4.1.1. Product 1- Finance Data Store

Finance Data store is primarily an analytical reporting that addressed multiyear data extracts. This product fulfilled user requirement where analytical report was downloaded for offline analysis or downstream feeds consumptions

4.1.2. Product 2 - Finance Navigator -OBIEE data

Finance Navigator – OBIEE is an ORACLE based reporting product that contained the financial reporting that was used for month end closing & reporting the company numbers. This reporting platform had reports segregated by business and domain area.

4.1.3. Product 3 - Finance Navigator -Tableau data

Finance Navigator – Tableau is a visualization reporting tool for analytical reporting. This used for aggregated reporting & trend analysis.

4.1.4. Human resource system -User data

Human Resource system (hereby known as HR) contains the user data for each user. This data provides crucial information to cluster users based on the job function, location, reporting manager, reporting business & department to name a few as shown in *Table 1 – User domain information*

Table 1 – User domain information

Sequence	Metadata name	Data Type	Description
1	User name	Text	Name of the user using the report
2	User SSO	Numeric	Unique user identification number
3	User Org ID	Numeric	Parent Organization ID
4	Use location ID	Numeric	User location ID
5	User origin country	Text	Origin of the user country
6	User reporting manager ID	Numeric	SSO id of the manager
7	Job type	Flag	Employee or Contractor or part time
8	Lega Entity ID	Numeric	Legal business Org of the user

4.1.5. Report usage metadata extracted from the above products

The report usage metadata contains the following information as shown in *Table 2 – Report usage metadata*

Table 2- Report usage metadata

Sequence	Metadata name	Data Type	Description
1	User name	Text	Name of the user using the report (derived from HR table)
2	User SSO	Numeric	Unique user identification number
6	Product used	Text	Name of the product that the user has used
7	Report name	Text	Name of the report that the user has used
8	Count of usage	Text	Usage count of the report

4.2 Data generation process (flow diagram)

Each of 3 applications listed above has a logging mechanism to generate the usage logs for the users. Each session is logged where the User-Product-Report_name combination is unique. If a user visits the same report on the same product twice, then the usage count is incremented.

The data is ingested & stored on a common S3 bucket for aggregation

Employee information is extracted from HR systems through a Talend job daily. The HR table contains the details of the all the users as shown in *Table 1 – User Domain Information*

The employee information is extracted for the unique users that are present in the reports.

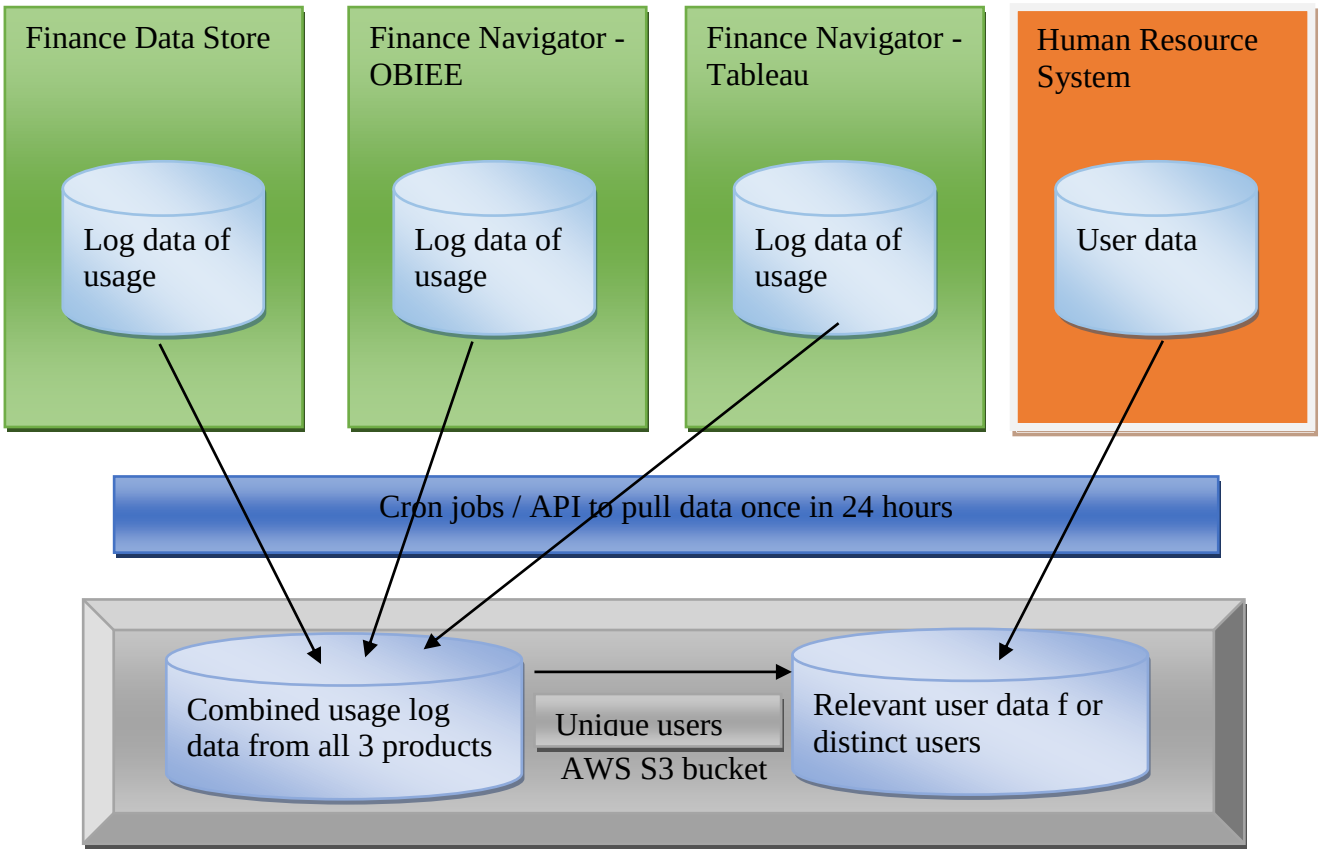


Fig 11 – Data logging for model

The *Fig 11 – Data logging for model* diagrams highlight the basic architecture to collate the data into S3 bucket from all the products for building the models. Each of the product had enabled to access their usage logs for user accessing the reports.

Finance Data Store – The logs were enabled by middle layer – DENODO

Finance Navigator OBIEE – Standard oracle BI tool had logs enabled through a custom view for ingestion

Finance Navigator Tableau – Standard Tableau logging was enabled through standard metadata capturing the logs.

5. Recommender model implementation

5.1. Data preprocessing

The report usage data as depicted in Table 2 was used for content-based implementation. 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.

Also, data elements with missing report name or product name was eliminated, as there could have been some session termination which resulted in partial capture of the data.

5.2. Content based similarity recommendation

5.2.1. Step 1 - Token generation

As a step 1 the tokens are generated for the combined Product Name & Qualified name to initiate the model.

Sample output of the tokenization is as follow shown in *Fig 12 – Token generation*

```
print (cv. vocabulary_)
{'fnobiee': 704, '001test001': 0, 'balances': 282, 'analysis': 212, 'ytd': 1672, 'bank': 285,
'balance': 268, 'basic': 292, 'net_pay': 1089, 'undis': 1557, 'vat': 1603, 'common': 413, 'cost':
452, 'copy': 437, 'default': 514, 'center': 361, 'feed': 674, 'correction': 450, 'corp': 442,
'notification': 1099, 'report': 1281, 'es': 622, 'cclgl': 359, 'es_journals_': 628, 'fx': 720,
'details': 526, 'home': 872, 'page_memsql': 1142, 'ibs': 888, 'accounts': 144, 'error': 621,
'journal': 949, 'activity': 158, 'mandatory': 1022, 'field': 677, 'menat': 1044, 'unusuals':
1563, 'ggo': 770, 'restructuring': 1316, 'temp': 1484, 'meta': 1050, 'qtd': 1225,
'my_je_extract': 1077, 'advanced': 182, 'ahmed': 191, 'othman': 1133, 'my_tb_extract': 1078,
'source': 1402, 'access': 137, 'ssa': 1430, 'td': 1479, 'condition': 425, 'condition1': 426,
'condition2': 427, 'test': 1492, 'tb': 1472, 'trial': 1539, 'turkey': 1545, 'und': 1556, 'acc':
135, 'fds': 673, 'drm': 570, 'account': 138, 'hierarchy': 868, 'company': 418, 'code': 395,
'function': 717, 'map': 1025, 'product': 1198, 'line': 1002, 'profit': 1201, 'headcount': 856,
'finance': 690, 'ptd': 1214, 'coco': 393, 'pc': 1154, 'mapping': 1026, 'sparkline': 1415,
'ddl_parquet': 510, 'general_ledger_global_query': 759, 'gl_dump_global': 787, 'ops': 1123, 'pw':
1220, 'biz': 309, 'specific': 1418, 'specific_ap': 1419, 'oc': 1109, 'hq': 876, '2019': 52,
'nam': 1080, 'income': 903, 'projects': 1207, 'ar': 234,……..Continue
```

Fig 12 – Token generation

5.2.2. Step 2 – Cosine similarity

As a step 2, the task was to generate the cosine similarity between the Product & Report name combination as shown in *Table 3 – Cosine Similarity of reports*

Table 3- Cosine similarity for reports

report_cosine_similar - NumPy array

	0	1	2	3	4	5	6	7	8	9	
0	1	0.353553	0.408248	0.5	0.5	0.408248	0.408248	0.408248	0.408248	0.5	0.4
1	0.353553	1	0.288675	0.353553	0.353553	0.288675	0.288675	0.288675	0.288675	0.353553	0.2
2	0.408248	0.288675	1	0.408248	0.408248	0.333333	0.333333	0.333333	0.333333	0.408248	0.3
3	0.5	0.353553	0.408248	1	1	0.816497	0.816497	0.816497	0.816497	0.5	0.4
4	0.5	0.353553	0.408248	1	1	0.816497	0.816497	0.816497	0.816497	0.5	0.4
5	0.408248	0.288675	0.333333	0.816497	0.816497	1	0.666667	0.666667	0.666667	0.408248	0.3
6	0.408248	0.288675	0.333333	0.816497	0.816497	0.666667	1	1	0.666667	0.816497	0.3
7	0.408248	0.288675	0.333333	0.816497	0.816497	0.666667	1	1	0.666667	0.816497	0.3
8	0.408248	0.288675	0.333333	0.816497	0.816497	0.666667	0.666667	0.666667	1	0.408248	0.3
9	0.5	0.353553	0.408248	0.5	0.5	0.408248	0.816497	0.816497	0.408248	1	0.4
10	0.408248	0.288675	0.333333	0.408248	0.408248	0.333333	0.333333	0.333333	0.333333	0.408248	
11	0.316228	0.223607	0.258199	0.316228	0.316228	0.258199	0.258199	0.258199	0.258199	0.316228	0.5
12	0.353553	0.25	0.288675	0.353553	0.353553	0.288675	0.288675	0.288675	0.288675	0.353553	0.2
13	0.353553	0.25	0.288675	0.353553	0.353553	0.288675	0.288675	0.288675	0.288675	0.353553	0.2
14	0.353553	0.25	0.288675	0.353553	0.353553	0.288675	0.288675	0.288675	0.288675	0.353553	0.5
15	0.316228	0.447214	0.516398	0.316228	0.316228	0.258199	0.258199	0.258199	0.258199	0.316228	0.2
16	0.408248	0.288675	0.333333	0.408248	0.408248	0.333333	0.333333	0.333333	0.333333	0.408248	0.3
17	0.408248	0.288675	0.333333	0.408248	0.408248	0.333333	0.333333	0.333333	0.333333	0.408248	0.3
18	0.5	0.353553	0.408248	0.5	0.5	0.408248	0.408248	0.408248	0.408248	0.5	0.4
19	0.408248	0.288675	0.333333	0.408248	0.408248	0.333333	0.333333	0.333333	0.333333	0.408248	0.3
20	0.408248	0.288675	0.333333	0.408248	0.408248	0.333333	0.333333	0.333333	0.333333	0.408248	0.3

Note – The diagonals are of same value, as it is comparing the report with itself, this getting a 100% match for the Cosine Similarity score which is 0-degree angle there by COS value as 1.

5.2.3. Step 3 – Sort the cosine similarity for candidate set

So, for each candidate report we have a row of similar report set. In this step we sort the data and pick the top 5 similarity score for the report.

5.2.4. Step 4 – Get top 5 matching report

In this step we use the index of the matching set to determine the name of the top 5 reports that are recommendation for the Product-Report name combination as shown in *Table 4 – Content based recommendation*

Table 4- Content based recommendation

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

Here in the above output get the following data

1. Report_ID → This is numeric representation of the Product-Report combination
2. Candidate → This is the index of the candidate set generated
3. Similarity → This is the similarity score of the matching reports in the cosine similarity
4. Report name → This is the combined name of the product & the report name. E.g. FNOBIEE is the Finance Navigator OBIEE product.
5. Candidate name → This is the name of the matching reports that can be considered as candidate set
6. Rank → This is rank of the candidate set. We have sorted the candidate set in descending order of similarity score and only top 5 scores are considered.

5.2.5. Future work

The following are the future considerations for the content-based similarity

1. Add more attributes for the report – Like report description, Report Functionality bucket, etc.
2. User preference – Consider the highest rated usage reports for a user and generate the candidate set for the same. Say user ABC has used 15 reports from 3 different products. The process should consider the top 5 reports based on the usage number,

and then generate the recommended candidate set for these top 5 reports. So, if we generate 5 candidates set for each report, we get 5 X 5 = 25 candidate set. The objective then to sort those candidates set by a combined score of Highest rated report by the user & the Highest similarity score.

5.3.User-profiling based collaborative recommendation

5.3.1. Step 1 – User profiling

The HR table is used to cluster users in 2 dimensions.

1. **Cluster 1 based on BAND – Business – Sub Business**

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

2. **Cluster 2 based on BAND – Job Function – Job Family**

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

5.3.2. Step 2 – Recommendation for the profile

For each of the cluster, the usage of reports is aggregated to identify the top usage pattern. In this process we generate recommendation for each unique combination of the cluster 1 attributes & cluster 2 attributes. Therefore, we generate 2 set of recommendation as shown in *Table 5 – Profile based recommendation – Cluster 1*, and *Table 6 – Profile based recommendation – Cluster 2.*

Set 1 – Recommendation for each unique entry of Cluster 1 → Band-Business-Sub Business

Table 5- Profile based recommendation – Cluster 1

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

5.3.2 Step 2 – Recommendation for each unique entry of Cluster 2 → Band-Job Function-Job Family

Table 6- Profile based recommendation – Cluster 2

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

5.3.3. Step 3 – User recommendation

In this step for any given user we determine the profile of the user. The profile can be as per Cluster 1 or Cluster 2. As the nature of the user is unknown, recommendation from both the cluster is presented to the user in form of the following grouping

1. Recommendation set 1 – "Similar reports used ***within*** your business"
2. Recommendation set 2 – "Similar reports used ***across*** the business"

5.4.Collaborative recommendation *(post midsem)*

In this collaborative filtering we prioritize user preference and similarity of the usage with other users. Matrix factorization technique is being used to generate the recommendation. We use 3 different matrix factorization & evaluate their performance to determine the optimal model for deployment. Following are the steps followed:

5.4.1.Step 1 – Data normalization using min-max method

The below screen shot is the data set we start with. The data elements contain User ID, Report ID, product type, total access, and the other report nomenclature details as shown in *Table 7 – User report usage score*

The user ID or User SSO is personal data and is categorized as protected information so this has been masked hereafter.

Data Set – The below is sample data set with which we start our journey of collaborative filtering.

Table 7- User report usage score

df - DataFrame

Index	user_sso	report_id	product_type_x	total_access	report_name	report_url	ep_domain_name
14660	[redacted]	545989877	2	5	Mandatory PO Filter	/shared/ WorkBench/Ref…	Report
14661	[redacted]	545989877	2	1	Mandatory PO Filter	/shared/ WorkBench/Ref…	Report
14662	[redacted]	545989877	2	41	Mandatory PO Filter	/shared/ WorkBench/Ref…	Report
14663	[redacted]	545989877	2	2	Mandatory PO filter	/shared/ WorkBench/Ref…	Report
14664	[redacted]	581898547	2	1	Indirect Spend - Vendor Summ…	/shared/ WorkBench/Ref…	Report
14665	[redacted]	823153395	2	1	AP06.06.01	/shared/ WorkBench/Ref…	Report
14666	[redacted]	823302349	2	3	AP06.01.02	/shared/ WorkBench/Ref…	Report
14667	[redacted]	823302349	2	1	AP06.01.02	/shared/ WorkBench/Ref…	Report
14668	[redacted]	823302349	2	37	AP06.01.02	/shared/ WorkBench/Ref…	Report
14669	[redacted]	823302349	2	1	AP06.01.02	/shared/ WorkBench/Ref…	Report
14670	[redacted]	823302350	2	6	AP06.01.01	/shared/ WorkBench/Ref…	Report
14671	[redacted]	823302350	2	2	AP06.01.01	/shared/ WorkBench/Ref…	Report
14672	[redacted]	823302350	2	74	AP06.01.01	/shared/ WorkBench/Ref…	Report
14673	[redacted]	823302350	2	2	AP06.01.01	/shared/ WorkBench/Ref…	Report
14674	[redacted]	1602502046	2	1	Indirect Spend - Vendor Summ…	/shared/05 - Misc. Testing…	Report
14675	[redacted]	24557354	2	4	Source System Access	/shared/06 - Other/Access/…	Report
14676	[redacted]	24557354	2	1	Source System Access	/shared/06 - Other/Access/…	Report
14677	[redacted]	24557354	2	1	Source System Access	/shared/06 - Other/Access/…	Report

The data set can be described as follows:

```
Raw dataframe ==> (26881, 7)
DataFrame after sso filter ==> (26871, 7)
DataFrame after report filter ==> (10707, 7)
```

The **key column in the data set is the Total access**. Like movie ratings, we consider this as the rating of the report based on the usage numbers.

```
Out[43]:
count    10707.000000
mean         6.194732
std         17.493825
min          1.000000
25%          1.000000
50%          3.000000
75%          7.000000
max        754.000000
Name: total_access, dtype: float64
```

As we see from the statistics of the data that the Min is 1, and the Max is 754. This is a huge variance, as result, this needs to be scaled down to a rating from 1 to 5.

Min Max scaling

Once Min max scaling has been done on the total access, the new values are stored in a separate column named Score. As you see from the below statistics, score now is scaled from 1 to 5, which is much more usable.

```
score_df['score'].describe()
Out[50]:
count    10707.000000
mean         1.447785
std          0.713293
min          1.000000
25%          1.000000
50%          1.195122
75%          1.585366
max          5.000000
Name: score, dtype: float64
```

The resulting data frame has the **total access** count corresponding scaled down value **score**. As we see the scores values are within the desired range to run our algorithm as shown in *Table 8 – Min-Max scaling of access count for model generation*

Table 8- Min-Max scaling of access count for model generation

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
14671	[redacted]	823302350	2	1.09756
14672	[redacted]	823302350	74	5
14673	[redacted]	823302350	2	1.09756
14674	[redacted]	1602502046	1	1
14675	[redacted]	24557354	4	1.29268
14676	[redacted]	24557354	1	1
14677	[redacted]	24557354	1	1
14678	[redacted]	24557354	4	1.29268
14679	[redacted]	24557354	2	1.09756
14680	[redacted]	24557354	1	1

5.4.2 Python Package used - Surprise

The Surprise package has been used for generating the below models & evaluating them.
https://surprise.readthedocs.io/en/stable/matrix_factorization.html
Surprise package offers recommended algorithms using the various matrix-factorization techniques.

5.4.3 Step 1 – SVD method

As a first step, we run the SVD model. The SVD model help decompose the usage matrix into latent vectors which can be used for recommendation. We run 5 cross validation to determine the performance of SVD as shown below.

```
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

The recommended data frame generated is as below. We take only the top 5 recommendation to be displayed to the end user as shown in *Table 9 – SVD recommendation generation*

Table 9- SVD recommendation generation

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

The latent vecotrs generated for SVD is as follows. We represnet the Item latent vecotrs & its bias, User latent vecotrs & it's bias. These numbers do not mean much to visual inspection, however these data elements are usefull for the interna SVD model as shown in *Table 10 – Latent vectors for SVD*

Table 10- Latent vectors for SVD

5.4.4 Step 2 - SVD++ method

As a 2nd step, we run the SVD++ model. The SVD++ model help decompose the usage matrix into latent vectors which can be used for recommendation. SVD++ also considers the user local preference and works well on sparse matrix. We run 5 cross validation to determine the performance of SVD++.

```
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
```

The recommended data frame generated is as below. We take only the top 5 recommendation to be displayed to the end user as shown in *Table 11 – Recommendation for SVD++*

Table 11- Recommendation for SVD++

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

The latent vecotrs generated for SVD++ is as follows. We represnet the Item latent vecotrs & its bias, User latent vecotrs & it's bias. These numbers do not mean much to visual inspection, however these data elements are usefull for the interna SVD++ model as shown in *Table 12 – Latent vectors for SVD++*

Table 12- Latent vectors for SVD++

5.4.5 Step 3 – NMF method

As a 3rd step, we run the NMF model. The NMF model help decompose the usage matrix weight of preference & component latent vectors which can be used for recommendation. NMF uses gradient descent function & error minimizing to get the optimal result. We run 5 cross validation to determine the performance of NMF

```
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
```

The recommended data frame generated is as below. We take only the top 5 recommendation to be displayed to the end user as shown in *Table 13 – Recommendation for NMF*

Table 13- Recommendation for NMF

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

The latent vecotrs generated for NMF is as follows. We represnet the Item latent vecotrs & its bias, User latent vecotrs & it's bias. These numbers do not mean much to visual inspection, however these data elements are usefull for the interna NMF model as shown in *Table 14 – Latent vectors for NMF*

Table 14- Latent vectors for NMF

5.4.6 Model performance evaluation

Once we have run through the model, we need to evaluate their performance to ensure that we select the right model to move forward. We compare the RMSE & MAE for each of the algorithms over mean of the 5 fold cross validation.

Comparison

Algorithm	RMSE	MAE
SVD	0.6055	0.3881
SVDpp	0.5735	0.358
NMF	0.5534	0.2935

We observe that the NMF (Non negative Matrix Factorization) method has the better predictions in both RMSE (Root Mean Square Error) & MAE (Mean Absolute Error) as shown in *Fig 13 – Root Mean Square Error comparison & Fig 14 – Mean Absolute Error comparison*

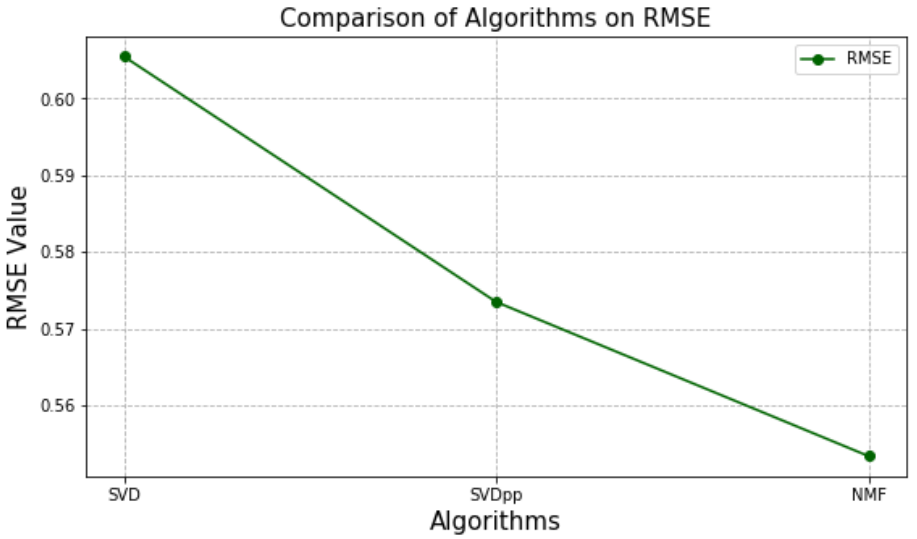


Fig 13 – Root Mean Square Error comparison

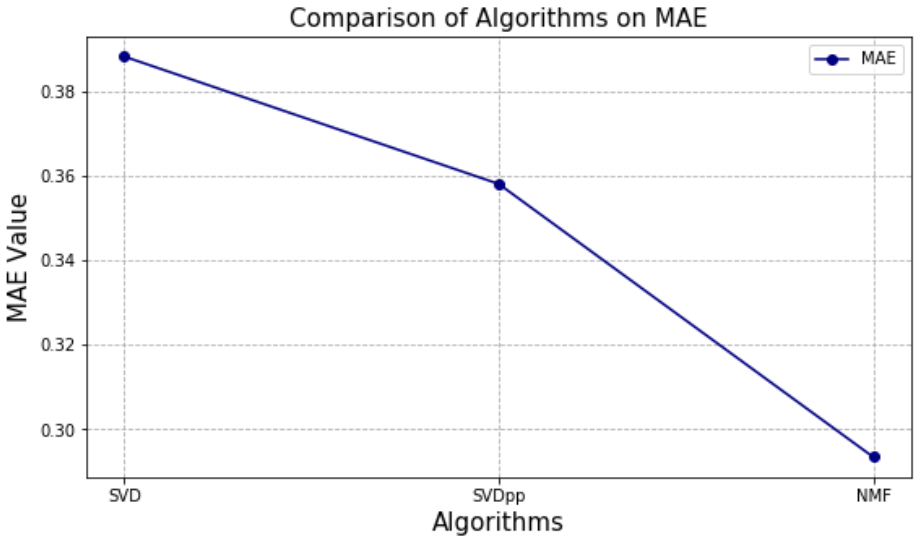


Fig 14 – Mean Absolute Error comparison

Therefore, as a conclusion we move forward with NMF (Non Negative Matrix Factorization) recommendation model for implementation.

One of the reasons that the Non-Negative Matrix factorization performs well over the others, is due to the usage & rating matrix is a very sparse matrix and the preference is localized to usage pattern in a cohesive manner. NMF performs better as it uses stochastic method to find the optimal solution.

5.4.7 NMF model recommendation report names

The NMF model had the prediction of just the Report ID based on the lowest RMSE & MAE. We translate the report IDs to report name for relatable outcome as shown in *Table 15 – NMF predictions* which is corresponding to Table 13

Table 15- NMF predictions report name

recomendation_reportname_df - DataFrame

Index	user_sso	report_id	score_estimate	rank	report_url	report_name	no_of_views	author	ast_modified_date	max_views_id	short_description	report_type	business_process	url
0	[redacted]	257315482	2.49657	1	/shared/Reference Rep…	Base Cost - Forecast_v1	0	Rudrakshala,S…	28:58.0	257315482	nan	8	nan	/shared/Reference Rep…
1	[redacted]	1449349248	2.26921	2	/shared/Reference Rep…	Base Cost - Forecast	0	Yesumithra,Ji…	28:58.0	1449349248	nan	8	nan	/shared/Reference Rep…
2	[redacted]	468633286	2.2593	3	/shared/CO - Corporate/Con…	Accrued Estimated Cos…	0	Rudrakshala,S…	57:33.0	468633286	nan	8	nan	/shared/CO - Corporate/Con…
3	[redacted]	597286461	2.25919	4	/shared/Usage Reports/Mano/…	Long Running Queries Report	0	Rudrakshala,S…	28:59.0	597286461	nan	8	nan	/shared/Usage Reports/Mano
4	[redacted]	957029259	2.03637	5	/shared/WorkBench/GPA…	GPASDetailRep…	0	Rudrakshala,S…	29:00.0	957029259	nan	8	nan	/shared/WorkBench
5	[redacted]	468633286	3.45253	1	/shared/CO - Corporate/Con…	Accrued Estimated Cos…	0	Rudrakshala,S…	57:33.0	468633286	nan	8	nan	/shared/CO - Corporate/Con…
6	[redacted]	257315482	3.12865	2	/shared/Reference Rep…	Base Cost - Forecast_v1	0	Rudrakshala,S…	28:58.0	257315482	nan	8	nan	/shared/Reference Rep…
7	[redacted]	597286461	3.09678	3	/shared/Usage Reports/Mano/…	Long Running Queries Report	0	Rudrakshala,S…	28:59.0	597286461	nan	8	nan	/shared/Usage Reports/Mano
8	[redacted]	1449349248	2.81755	4	/shared/Reference Rep…	Base Cost - Forecast	0	Yesumithra,Ji…	28:58.0	1449349248	nan	8	nan	/shared/Reference Rep…
9	[redacted]	957029259	2.63825	5	/shared/WorkBench/GPA…	GPASDetailRep…	0	Rudrakshala,S…	29:00.0	957029259	nan	8	nan	/shared/WorkBench
10	[redacted]	468633286	3.0142	1	/shared/CO - Corporate/Con…	Accrued Estimated Cos…	0	Rudrakshala,S…	57:33.0	468633286	nan	8	nan	/shared/CO - Corporate/Con…
11	[redacted]	597286461	3.00128	2	/shared/Usage Reports/Mano/…	Long Running Queries Report	0	Rudrakshala,S…	28:59.0	597286461	nan	8	nan	/shared/Usage Reports/Mano
12	[redacted]	257315482	2.90694	3	/shared/Reference Rep…	Base Cost - Forecast_v1	0	Rudrakshala,S…	28:58.0	257315482	nan	8	nan	/shared/Reference Rep…
13	[redacted]	957029259	2.54127	4	/shared/WorkBench/GPA…	GPASDetailRep…	0	Rudrakshala,S…	29:00.0	957029259	nan	8	nan	/shared/WorkBench
14	[redacted]	189755771	2.35848	5	/shared/04 - Training/Tool…	Navigator - Step 1	0	Manne,Neelima	52:05.0	189755771	nan	8	nan	/shared/04 - Training/Tool…

5.4.8 Recommendation display to users (Work in progress)

The integration team is working for final integration of the recommendation. Below is the wireframe design of the recommendation report presentation. The collaborative recommendation is 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 as shown in the wireframe design *Fig 15 – Collaborative recommendation proposed display to user*

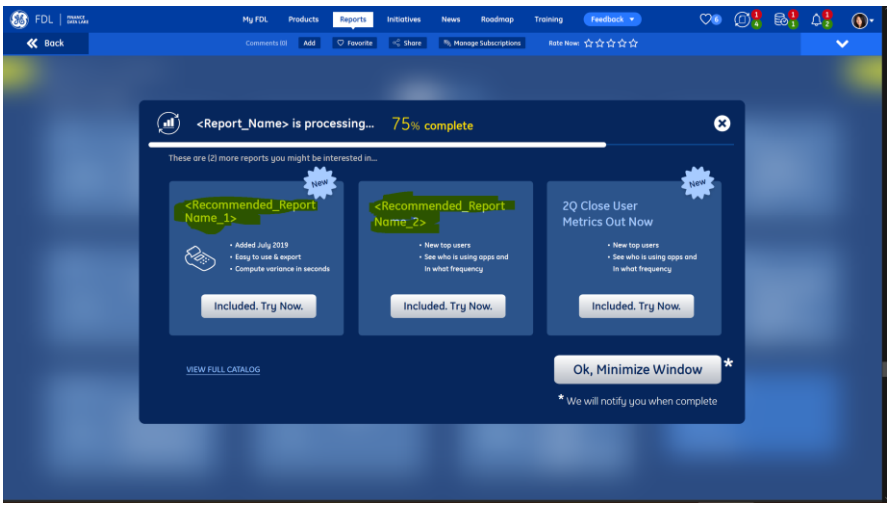


Fig 15 – Collaborative recommendation proposed display to user

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 as shown in wire frame design *Fig 16 – Content based recommendation proposed display to users*

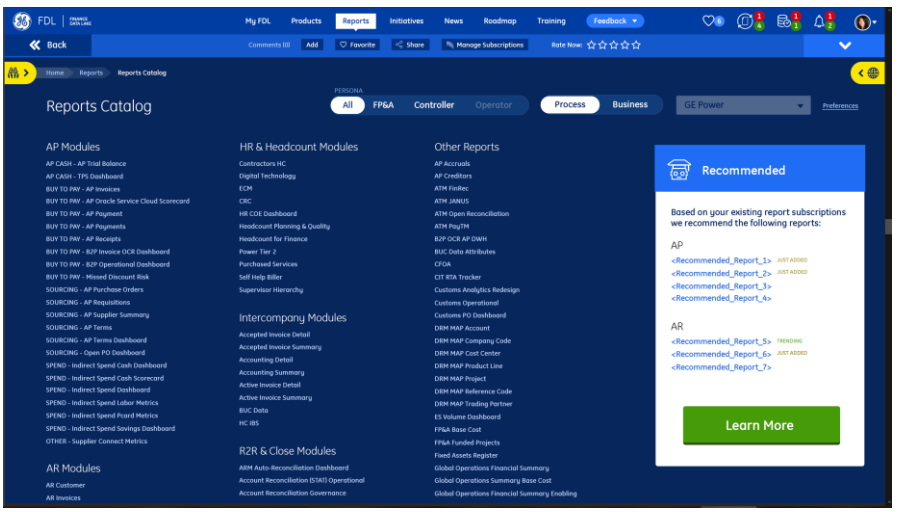


Fig 16 – Content based recommendation proposed display to user

6. Conclusion & Summary

The recommendation system development allows to identify the underlying traits & usage pattern of the current products & their report base. 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. The recommendation will be presented to the user at various aspects of the process in order to drive a change in user behaviors by ensuing adoption of the recommended reports or most popular or similar reports.
2. Product usage analysis– The underlying data of the recommendation engine helps the product managers in the following aspect
    a. Identify patterns of reports usage across various user profiles thereby cross training & socializing relevant reports
    b. Identify reports that are sparsely used and retrospect the cause of the same – Is the report not relevant or does not have the adequate functionality?
    c. Dynamic resource allocation for the infrastructure team – Reports & underlying data that are highly used needs more compute resources during peak usage period. The analysis would be used by the infrastructure team to explore ways for dynamic scaling

The recommendation engine is the first step towards understanding user traits and how it changes across geographical regions or various GE business.

7. 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.

Immediate project fulfillment

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

Pipeline

1. Optimize the recommendation model using neural networks to provide customized recommendation for each user
2. Identify trends & change in usage patterns of users to predict the most suitable report at that point of time or at that region.
3. Incorporate other attributes in the recommendation algorithm to make more effective recommendation with higher hit ratio.

8. References

[1] Science Direct - Recommendation systems: Principles, methods and evaluation
https://www.sciencedirect.com/science/article/pii/S1110866515000341
[2] Towards data science – Youtube video recommendation
https://towardsdatascience.com/how-youtube-recommends-videos-b6e003a5ab2f
[4] Kaggle – Recommender system in Python
https://www.kaggle.com/gspmoreira/recommender-systems-in-python-101
[5] Analytics Vidhya – Recommender system in Python
https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-recommendation-engine-python/
[6] Python Surprise package
https://surprise.readthedocs.io/en/stable/
[7] SVD++
http://dparra.sitios.ing.uc.cl/classes/recsys-2015-2/student_ppts/CRojas_SVDpp-PMF.pdf
[8] RMSE
https://gisgeography.com/root-mean-square-error-rmse-gis/
[9] MAE
https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d

9. Acronyms

Num	Acronyms	Full form / meaning
1	RMSE	Root Mean Square Error
2	MAE	Mean Absolute Error
3	SVD	Singular Value Decomposition
4	NMF	Non-Negative Matrix Factorization
5	OBIEE	Oracle Business Intelligence Enterprise Edition
6	FDS	Finance Data Store
7	SSO	Single Sign On

Checklist of items for the Final Dissertation Report

This checklist is to be attached as the last page of the report.

This checklist is to be duly completed, verified and signed by the student.

1.	**Is the final report neatly formatted with all the elements required for a technical Report?**	Yes
2.	Is the Cover page in proper format as given in Annexure A?	Yes
3.	Is the Title page (Inner cover page) in proper format?	Yes
4.	(a) Is the Certificate from the Supervisor in proper format? (b) Has it been signed by the Supervisor?	Yes Yes
5.	Is the Abstract included in the report properly written within one page? Have the technical keywords been specified properly?	Yes Yes
6.	Is the title of your report appropriate? **The title should be adequately descriptive, precise and must reflect scope of the actual work done.** Uncommon abbreviations / Acronyms should not be used in the title	Yes
7.	Have you included the List of abbreviations / Acronyms?	Yes
8.	Does the Report contain a summary of the literature survey?	NA
9.	Does the Table of Contents include page numbers? (i). Are the Pages numbered properly? (Ch. 1 should start on Page # 1) (ii). Are the Figures numbered properly? (Figure Numbers and Figure Titles should be at the bottom of the figures) (iii). Are the Tables numbered properly? (Table Numbers and Table Titles should be at the top of the tables) (iv). Are the Captions for the Figures and Tables proper? (v). Are the Appendices numbered properly? Are their titles appropriate	Yes Yes Yes Yes Yes Yes
10.	Is the conclusion of the Report based on discussion of the work?	Yes
11.	Are References or Bibliography given at the end of the Report? Have the References been cited properly inside the text of the Report? Are all the references cited in the body of the report	Yes Yes Yes
12.	Is the report format and content according to the guidelines? The report should not be a mere printout of a Power Point Presentation, or a user manual. Source code of software need not be included in the report.	Yes

Declaration by Student:

I certify that I have properly verified all the items in this checklist and ensure that the report is in proper format as specified in the course handout.

Bangalore

Place

4-Nov-2019

Date

Siddhartha Bg

Signature of the Student

SIDDHARTHA BANERJEE

Name

2017HT13125

ID No:

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
WORK-INTEGRATED LEARNING PROGRAMMES DIVISION
First Semester 2019-2020

BITS ZG628T : Dissertation EC-3 Pre-Final Evaluation Sheet

Upload Softcopy of Final Dissertation Report, Pre-Final Evaluation Sheet, and Final Presentation by October 31, 2019

ID No.	:	**2017HT13125**
NAME OF THE STUDENT	:	**SIDDHARTHA BANERJEE**
EMAIL ADDRESS	:	**Banerjee.siddhartha.sb@gmail.com**
SUPERVISOR'S NAME	:	**Ramji Sarangarajan**
DISSERTATION TITLE:	:	**Recommender Systems for suggesting the financial reports to users based on the usage pattern for Finance Data Lake Products**
Details of work done till date (with reference to Outline)	:	Design & implemented usage data logging across 3 products to build the recommendation models. Implemented Content based recommendation for the reports and predicted similar reports using cosine similarity. Implemented user profile-based recommendation using 2 different clustering - inter & intra business. ***Post mid-sem*** -Designed collaborative filtering using 3 matrix factorization techniques – SVD (Singular Value Decomposition), SVD++ (SVD + Implicit ratings) & NMF (Non Negative matrix factorization). Conducted 5-fold cross validation on these algorithms to determine the best fit for final implementation. Implemented DB connectivity to pull real time data for ongoing implementation & write back the recommendation output onto DB for display to users.

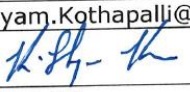
Dissertation Final Evaluation *(Please put a tick (✔) mark in the appropriate box)*

EC No.	Component	Excellent	Good	Fair	Poor
1.	Final Dissertation Report	✓			
2	Final Seminar and Viva-Voce	✓			

EC No.	Component	Excellent	Good	Fair	Poor
1.	Technical/Professional Competence	✓			
2	Work Progress and Achievements	✓			
3	Documentation and expression	✓			
4	Initiative and Originality	✓			
5	Research & Innovation	✓			
6	Relevance to the work environment	✓			

Please ENCIRCLE the Recommended Final Grade: Excellent / Good / Fair / Poor

Remarks of the Supervisor:
The algorithms determined by Siddhartha Banerjee are a key success factor for the recommendation engine within the FDL Wrapper tool. The ability to recommend reports to users based on text recognition will benefit the end user greatly as they can now access previously built reports without having to create them individually. The other recommendations based on a user's role in the organization and based on the organization that a user belongs to, will not only provide users access to a larger repository of reports but also expose them to newer forms of reporting analysis. This initiative by Siddhartha will be crucial to the FDL's reporting journey as it aims to improve the overall user experience within the FDL ecosystem.

	Supervisor	**Additional Examiner**
Name	Ramji Sarangarajan	Shyam Kumar Kothapalli
Qualification	BE, MS	BTech (20 yrs.+ exp)
Designation	Director – Data & Analytics	Director – Data & Analytics
Employing Orgn and Location	GE, Bangalore	GE, Bangalore
Phone No. (with STD Code)	9740155133	7259106689
Email Address	ramji.sarangarajan@ge.com	Shyam.Kothapalli@ge.com
Signature		
Date	4-Nov-19	4-Nov-19