Internship Report

Customised Article Recommendation Engine

Siddheswar Mukherjee
MS Applied Computer Science / EEIP
Siddheswar.Mukherjee@vub.be

Duration: July 2019 – August 2019

CONTENT

INTRODUCTION

This project focuses on creation of a Proof-Of-Concept (POC) for a customised Recommendation Engine that suggests reading material to end users of a content marketing website maintained and managed by EEIP. This is the first step towards a mobile-first website focused on user experience. This work was undertaken in close collaboration with other team members of EEIP, working on content categorisation, company taxonomy and master data management. However, the decision on the algorithm side, was left open by the organisation, and the final code has been developed using standard open source libraries in Python. The final API has been developed to run using the minimum possible infrastructure. The focus has been on making an economically viable Recommendation System. Hence there are multiple improvements that need to be made to achieve high speed results. Suggestions for the same have been made in the concluding sections.

BUSINESS REQUIREMENT :

Content Marketing Website that displays articles to end users.

- Website gets ~100,000 hits per month, 75% from Europe.
- There is no provision of registration or user login. IP addresses / Session IDs will be used to track user behaviour (interest in different topics, languages etc).
- Currently homepage content, readable articles, are presented to all website viewers in the same default order. In future "*user behaviour data*" will be used to show content in order of relevance for users.
- Suggested reading links are recommended at the end of each article, which, at the moment are manually set. In future usage of "*user behaviour data*" will be used to show dynamically generated relevant suggestions.

TECHNICAL REQUIREMENT SPECIFICATIONS:

The environment consists of a TYPO3 based website which will use a customised Recommendation Engine (RE).

The RE will consist of 3 parts:

1. Content Based Similarity
2. Collaborative Similarity
3. Additional Business Logic

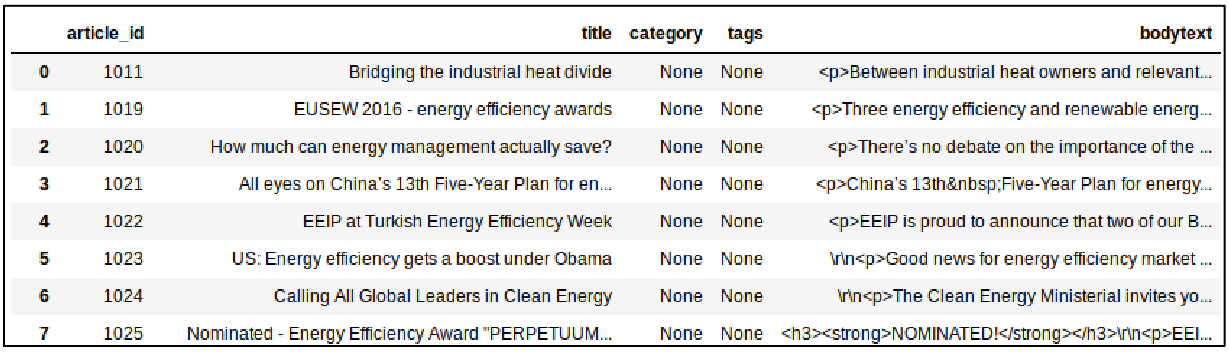
The Recommendation core will be accessible via various APIs. Apart from the standard recommendation methods, the final list is created by applying additional business rules. This logic and the learning models have been discussed in the following sections.

DATASET:

Content & Metadata

Information available for individual articles needs to be imported directly from the website database (MySql) and consists of the following attributes.

- Textual content for the articles (all translated into English) .
- Title
- Category
- Tags



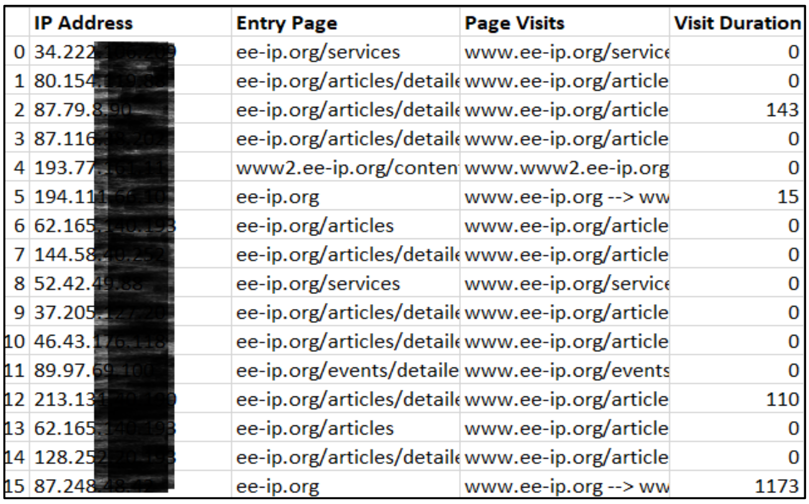
	article_id	title	category	tags	bodytext
0	1011	Bridging the industrial heat divide	None	None	<p>Between industrial heat owners and relevant...
1	1019	EUSEW 2016 - energy efficiency awards	None	None	<p>Three energy efficiency and renewable energ...
2	1020	How much can energy management actually save?	None	None	<p>There's no debate on the importance of the ...
3	1021	All eyes on China's 13th Five-Year Plan for en...	None	None	<p>China's 13th Five-Year Plan for energy...
4	1022	EEIP at Turkish Energy Efficiency Week	None	None	<p>EEIP is proud to announce that two of our B...
5	1023	US: Energy efficiency gets a boost under Obama	None	None	\r\n<p>Good news for energy efficiency market ...
6	1024	Calling All Global Leaders in Clean Energy	None	None	\r\n<p>The Clean Energy Ministerial invites yo...
7	1025	Nominated - Energy Efficiency Award "PERPETUUM...	None	None	<h3><strong>NOMINATED!</strong></h3>\r\n<p>EEI...

Figure 1 Metadata and Content

User Interaction

Interaction data is available via a third party IP enrichment service in ***.csv*** format. Relevant fields are as follows:

- IP Address
- URLs Visited
- Total session duration



	IP Address	Entry Page	Page Visits	Visit Duration
0	34.222[redacted]	ee-ip.org/services	www.ee-ip.org/service	0
1	80.154[redacted]	ee-ip.org/articles/detaile	www.ee-ip.org/article	0
2	87.79.[redacted]	ee-ip.org/articles/detaile	www.ee-ip.org/article	143
3	87.116[redacted]	ee-ip.org/articles/detaile	www.ee-ip.org/article	0
4	193.77[redacted]	www2.ee-ip.org/conten	www.www2.ee-ip.org	0
5	194.11[redacted]	ee-ip.org	www.ee-ip.org --> ww	15
6	62.165[redacted]	ee-ip.org/articles	www.ee-ip.org/article	0
7	144.58[redacted]	ee-ip.org/articles/detaile	www.ee-ip.org/article	0
8	52.42.[redacted]	ee-ip.org/services	www.ee-ip.org/service	0
9	37.205[redacted]	ee-ip.org/articles/detaile	www.ee-ip.org/article	0
10	46.43.[redacted]	ee-ip.org/articles/detaile	www.ee-ip.org/article	0
11	89.97.[redacted]	ee-ip.org/events/detaile	www.ee-ip.org/events	0
12	213.13[redacted]	ee-ip.org/articles/detaile	www.ee-ip.org/article	110
13	62.165[redacted]	ee-ip.org/articles	www.ee-ip.org/article	0
14	128.25[redacted]	ee-ip.org/articles/detaile	www.ee-ip.org/article	0
15	87.248[redacted]	ee-ip.org	www.ee-ip.org --> ww	1173

Figure 2 Implicit Feedback

Note: *In future, session information will be directly captured and stored in-house. In order to be compliant with GDPR, IP addresses cannot be stored within European jurisdiction. Cookies and Session IDs will be used to identify users in the feedback data.*

CONTENT BASED RECOMMENDATION:

This algorithm uses the text content and article metadata to calculate a similarity matrix amongst articles. Steps used are as follows:

Pre-processing

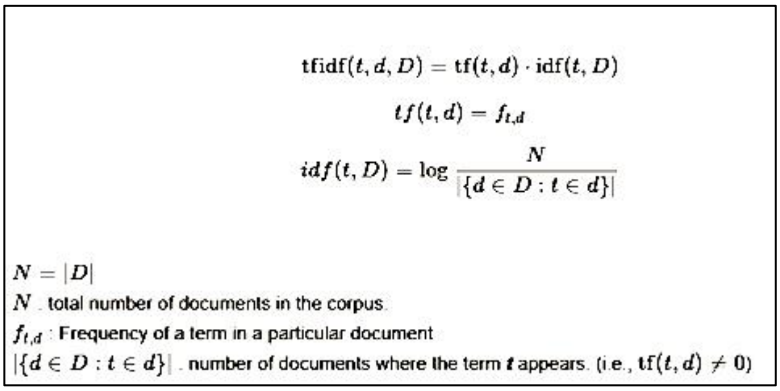
1. HTML Tag Removal : The text content is available in HTML format.
2. Lower case conversion : Applied to content and title.
3. Stemming: After due comparison, Snowball stemming is preferred over the Porter Stemming algorithm due to faster performance.
4. Category & Tag: The category and tags are imported from the database concatenated together in a comma separated format.

Model Creation

1. **TF-IDF & Term Frequency:**

   i. Term Frequency – Inverse Document Frequency (TF-IDF) measure of each individual word in the entire corpora is calculated and aims at downweighing words which are more common among articles. Consequently important keywords specific to individual articles are highlighted. This allows precise text based similarity calculation. TF-IDF count is used for **content** and **title** attributes.

   ii. Simple term frequency is calculated for the Tags and Categories. This metadata is comprised of keywords; so, unlike article content, the presence of common **tags** and **categories** do not need to be downweighed.



$$\mathrm{tfidf}(t, d, D) = \mathrm{tf}(t, d) \cdot \mathrm{idf}(t, D)$$

$$tf(t, d) = f_{t,d}$$

$$idf(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

$N = |D|$

N : total number of documents in the corpus

$f_{t,d}$: Frequency of a term in a particular document

$|\{d \in D : t \in d\}|$: number of documents where the term t appears. (i.e., $\mathrm{tf}(t, d) \neq 0$)

Figure 3 TF-IDF calculation

2. Similarity Matrices:

Two content recommendation APIs have been developed: Cosine Similarity Recommendation and TS-SS Recommendation.

Cosine Similarities:

Individual cosine similarities are generated for **Content, Title** and **Keywords**. This finally creates N * N matrix mapping the similarity between every article pair possible. These matrices can now be used to generate the top 10 articles similar to any article. At the time of recommendation, the 3 similarity values are combined using the weighted average method. The weights are manually set.

Triangle – Sector Similarity (TS-SS):

The TF-IDF vectors for **content** is also used to create a TS-SS similarity matrix. This can also be combined with the cosine similarity values for title and metatags. TS-SS similarity is designed to take more details into account, as compared to Cosine Similarity, when calculating text based similarities. Hence it is expected to provide more accurate results for big datasets.

Recommendation & Additional Business Logic

The content recommendation APIs accept a JSON object that contains article id and durations. These values will be obtained from cookies stored in the end user’s web browser. Top 10 suggestions for each article is calculated from the concerned similarity matrices. The time duration values are normalised and these weights are multiplied with the similarity scores of all suggestions. Finally the top 10 articles are shortlisted from the whole set and passed back in JSON format.

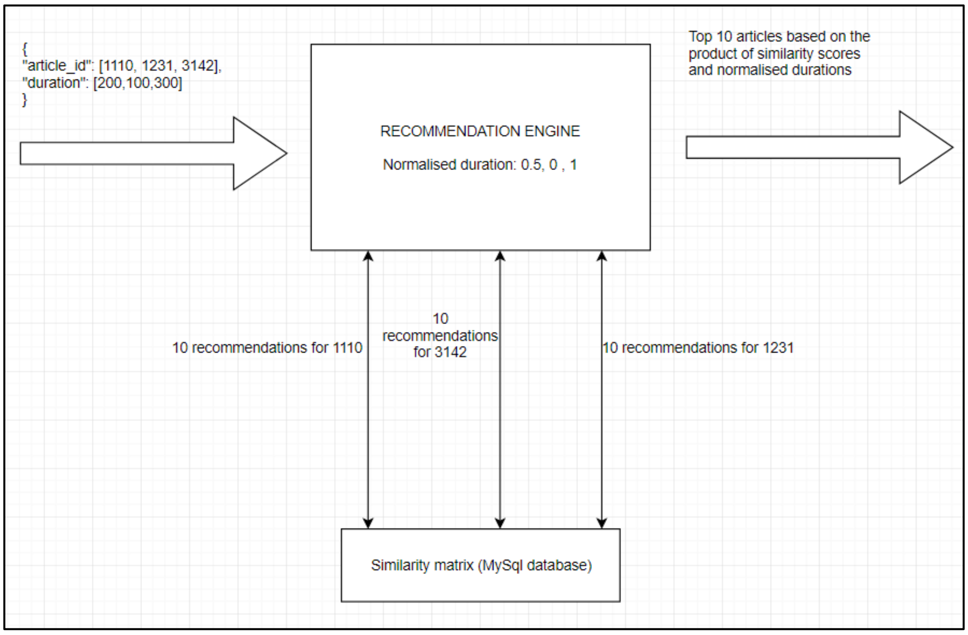
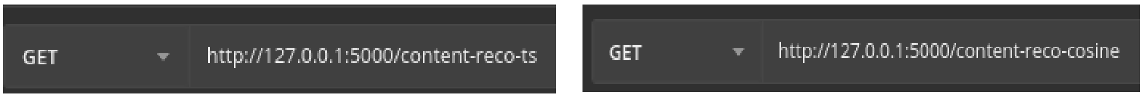

{
"article_id": [1110, 1231, 3142],
"duration": [200,100,300]
}
RECOMMENDATION ENGINE
Normalised duration: 0.5, 0 , 1
Top 10 articles based on the product of similarity scores and normalised durations
10 recommendations for 1110
10 recommendations for 3142
10 recommendations for 1231
Similarity matrix (MySql database)

Figure 4 Content Recommendation API Activity

As already mentioned, there is a provision for recommendations based on 2 different similarity measures – Cosine and TS-SS. As a result there are 2 content recommendation APIs as shown below.

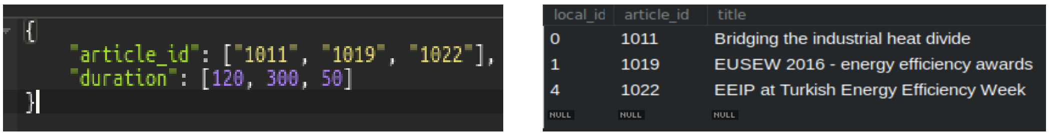


GET http://127.0.0.1:5000/content-reco-ts

GET http://127.0.0.1:5000/content-reco-cosine

Figure 5 Content Recommendation APIs

For testing purposes the following JSON file being used contains 3 article ids and the corresponding durations. The actual names of the articles are shown alongside for reference.



```
{
    "article_id": ["1011", "1019", "1022"],
    "duration": [120, 300, 50]
}
```

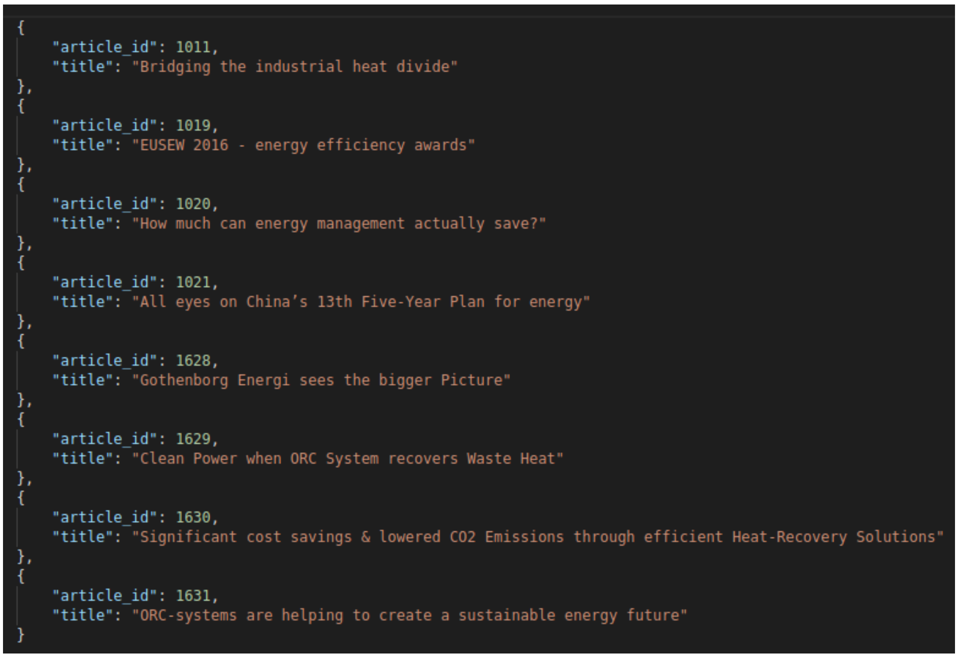
local_id	article_id	title
0	1011	Bridging the industrial heat divide
1	1019	EUSEW 2016 - energy efficiency awards
4	1022	EEIP at Turkish Energy Efficiency Week
NULL	NULL	NULL

Figure 6 JSON input

Observation:

It was observed that similar recommendations were obtained from both similarity measures. The explanation to this is the low number of articles present in the database at the moment. Apart from that, since EEIP is shifting to a new data management system, articles have not yet been categorised and tagged. After these issues have been resolved, some difference in results is expected.

Initial manual testing has yielded good quality suggestions as confirmed by EEIP content managers and Article writers.



```
{
    "article_id": 1011,
    "title": "Bridging the industrial heat divide"
},
{
    "article_id": 1019,
    "title": "EUSEW 2016 - energy efficiency awards"
},
{
    "article_id": 1020,
    "title": "How much can energy management actually save?"
},
{
    "article_id": 1021,
    "title": "All eyes on China’s 13th Five-Year Plan for energy"
},
{
    "article_id": 1628,
    "title": "Gothenborg Energi sees the bigger Picture"
},
{
    "article_id": 1629,
    "title": "Clean Power when ORC System recovers Waste Heat"
},
{
    "article_id": 1630,
    "title": "Significant cost savings & lowered CO2 Emissions through efficient Heat-Recovery Solutions"
},
{
    "article_id": 1631,
    "title": "ORC-systems are helping to create a sustainable energy future"
}
```

Figure 7 Content based recommendations

COLLABORATIVE RECOMMENDATION:

The collaborative model calculates similarity between articles based on the user interaction. This type of recommendation does not take into account the similarity based on the content and metadata. The idea behind this is that a particular user may be interested in varied unrelated topics. Collaborative Filtering aims to target similarities based on user feedback; which in this case is implicit. This will capture the aforementioned varied interests.

User interaction data is available via a third party IP enrichment service. This data has to be pre-processed in order to draw meaningful results.

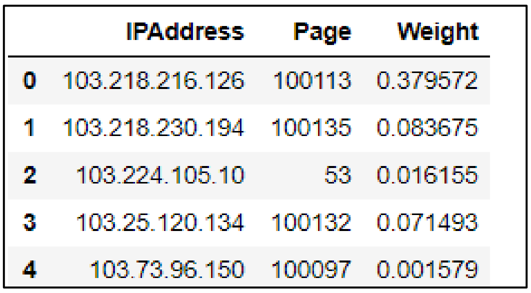
Pre-processing

1. **Feature Reduction:** Interaction data is available along with a lot of attributes that cannot be used as feedback. These are removed and finally the columns that remain are the page URLs visited by an user in a session and the total time.

2. **URL to Article ID:** The article ids corresponding to the URLs are matched. This requires rigorous pattern matching, since, numerous individual articles have multiple URLs. Thereafter sessions that consist entirely of non-article pages are removed.

3. **Article Page Visit Durations:** A peculiarity of the data is that the total session time is provided in an aggregated manner. Hence the visit time has been assumed to be equally divided among all pages. Although there is an obvious flaw in this assumption, in absence of any other source of historical data, this was the only way to forward with a collaborative model. The session times were equally divided into pages. Finally all other interactions other than article pages are removed. Total time spent by each user on each article page is aggregated.

4. **Final Data-frame:** The final data-frame consists rows IP addresses (which serves as user id), Page ID (which is effectively the article ID) and normalised visit durations.



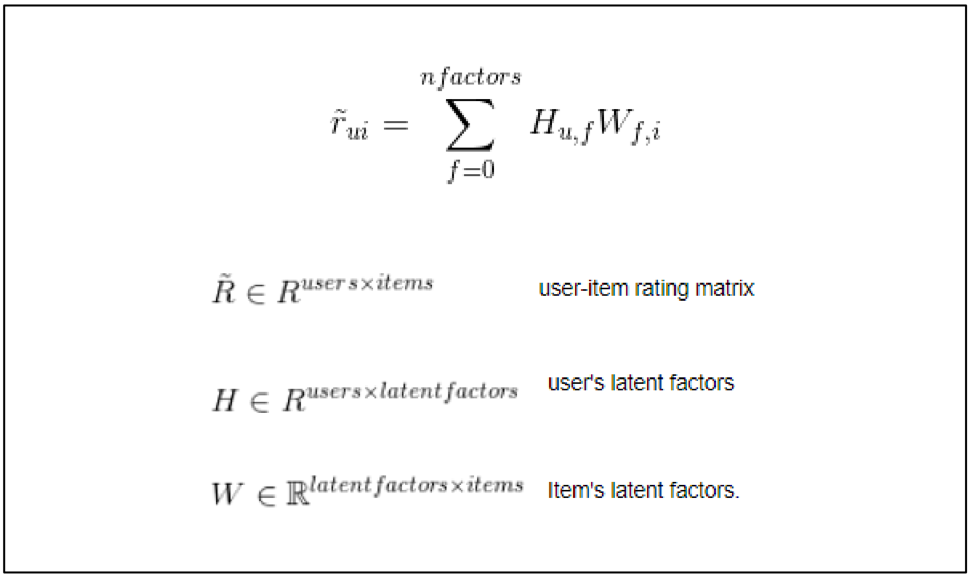
	IPAddress	Page	Weight
0	103.218.216.126	100113	0.379572
1	103.218.230.194	100135	0.083675
2	103.224.105.10	53	0.016155
3	103.25.120.134	100132	0.071493
4	103.73.96.150	100097	0.001579

Figure 8 Preprocessed Interaction Data

Model Fitting

The basic idea of collaborative model is to create a mapping of each user to each item. However due to enormous size of both the factors, this is unfeasible. Matrix Factorisation algorithms are required to find (***User - Latent_Factors)*** and (***Latent_Factors – Article)*** matrices instead of a (***User – Article***) matrix. This reduces the high dimensionality problem. User interactions in this case are implicit feedbacks, so, the dataset may not be sparse. Alternating Least Squares is the method of choice for matrix factorisation, since it is faster than other alternatives (E.g. Stochastic Gradient Descent).

These two matrices will finally be used to generate the predicted interaction, which, in this case is the likeliness of the reading a particular article.



$$\tilde{r}_{ui} = \sum_{f=0}^{nfactors} H_{u,f} W_{f,i}$$

$\tilde{R} \in R^{users \times items}$ user-item rating matrix

$H \in R^{users \times latentfactors}$ user's latent factors

$W \in \mathbb{R}^{latentfactors \times items}$ Item's latent factors.

Figure 9 Collaborative Predictions using Feedback

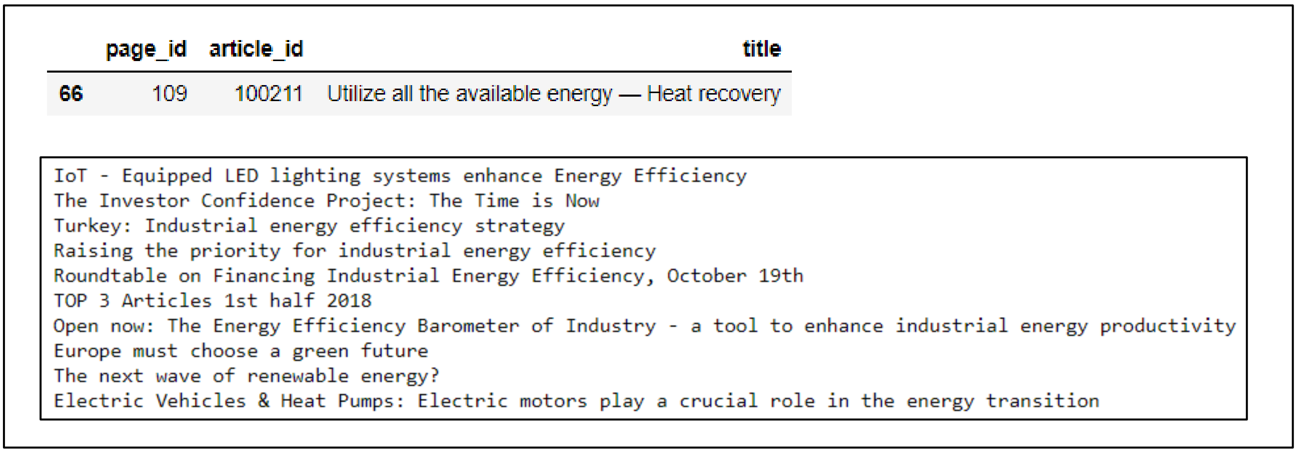
Recommendation

Unlike content similarity, collaborative models require storage of user interaction data in the database. This behavioural data is used to pre-calculate the models. Two types of recommendations are possible:

1. Top 10 articles similar to a particular article.
2. Top 10 articles that may be liked by a particular user.

The input for this recommendation would a ***user_id*** or an ***article_id*** and the output would consist of top 10 recommendations. Optionally the similarity scores can also be obtained alongside. These scores can be useful in applying additional business logic as described in the content recommendation section. Additionally the recommendations from both engines can be combined to provide ***Hybrid Recommendations***.

1. In order to test articles similar to a particular content, the following article is chosen. The top 10 articles as suggested by the collaborative model is shown below.

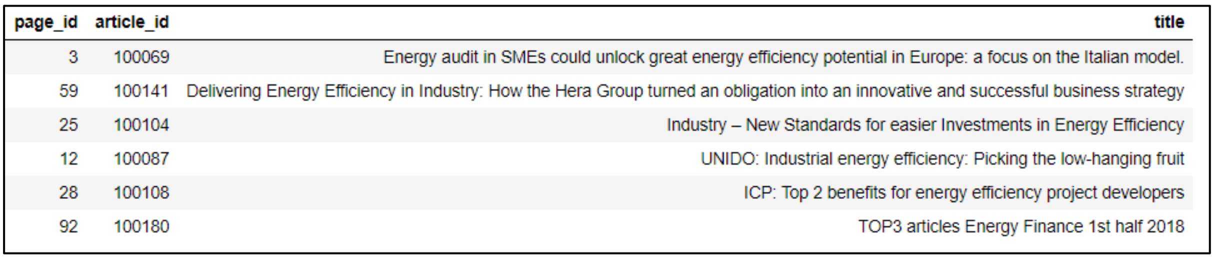


	page_id	article_id	title
66	109	100211	Utilize all the available energy — Heat recovery

```
IoT - Equipped LED lighting systems enhance Energy Efficiency
The Investor Confidence Project: The Time is Now
Turkey: Industrial energy efficiency strategy
Raising the priority for industrial energy efficiency
Roundtable on Financing Industrial Energy Efficiency, October 19th
TOP 3 Articles 1st half 2018
Open now: The Energy Efficiency Barometer of Industry - a tool to enhance industrial energy productivity
Europe must choose a green future
The next wave of renewable energy?
Electric Vehicles & Heat Pumps: Electric motors play a crucial role in the energy transition
```

Figure 10 Articles similar to an Article

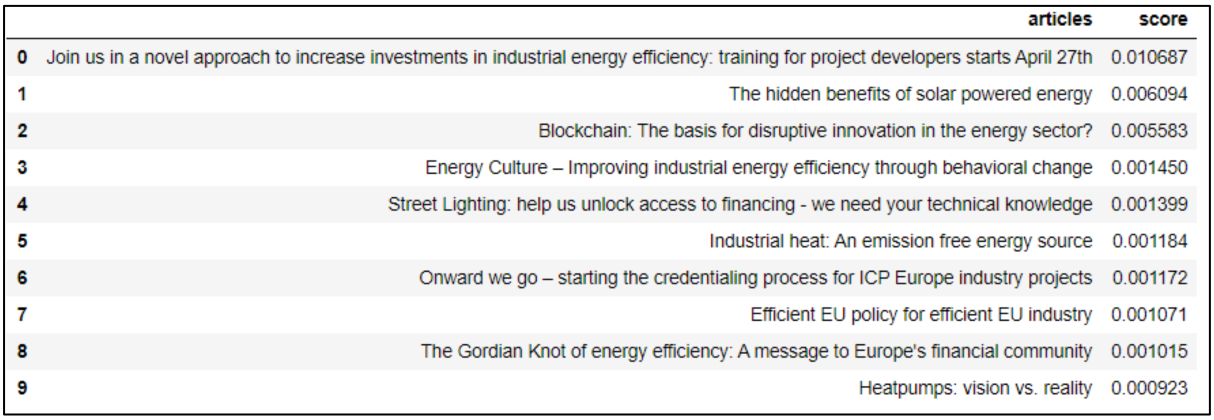
2. The next suggestion takes as input a particular user id. The user chosen for testing purpose has already read the following articles.



page_id	article_id	title
3	100069	Energy audit in SMEs could unlock great energy efficiency potential in Europe: a focus on the Italian model.
59	100141	Delivering Energy Efficiency in Industry: How the Hera Group turned an obligation into an innovative and successful business strategy
25	100104	Industry – New Standards for easier Investments in Energy Efficiency
12	100087	UNIDO: Industrial energy efficiency: Picking the low-hanging fruit
28	100108	ICP: Top 2 benefits for energy efficiency project developers
92	100180	TOP3 articles Energy Finance 1st half 2018

Figure 11 User history

The recommendations as suggested for the mentioned user are as follows. The list has been displayed in decreasing order of likelihood and the confidence scores for user can also be observed.



	articles	score
0	Join us in a novel approach to increase investments in industrial energy efficiency: training for project developers starts April 27th	0.010687
1	The hidden benefits of solar powered energy	0.006094
2	Blockchain: The basis for disruptive innovation in the energy sector?	0.005583
3	Energy Culture – Improving industrial energy efficiency through behavioral change	0.001450
4	Street Lighting: help us unlock access to financing - we need your technical knowledge	0.001399
5	Industrial heat: An emission free energy source	0.001184
6	Onward we go – starting the credentialing process for ICP Europe industry projects	0.001172
7	Efficient EU policy for efficient EU industry	0.001071
8	The Gordian Knot of energy efficiency: A message to Europe's financial community	0.001015
9	Heatpumps: vision vs. reality	0.000923

Figure 12 Collaborative Filtering Recommendations

Observation:

It has been observed that the performance of collaborative model has been quite inferior as compared to the Content recommendation results. This can be attributed to the lack of sufficient data pertaining to end-user interaction with article pages. Most importantly more accurate page visit information is required, especially the time spent in individual article pages.

FURTHER EXTERNAL WORK:

- **Evaluation** of Recommendations in production environment.
- **Integration into website**: The website at present uses the TYPO3 extension: *News System theme* to display content. Integration of API recommendations into the website interface needs creation of TYPO3 extension that alters the default display order of links and articles as set by the theme.

SUGGESTED IMPROVEMENTS:

- **Incremental model creation:** Every time new articles are added, the whole model needs to be recalculated. This will result in an enormous overhead in time. Incremental model creation can be implemented by additionally storing the Term Frequencies(**tf**), Document Frequencies(**idf**), Euclidean Distances and Theta matrices can be stored in the database along with the final similarities. This will eliminate recalculation of all similarity values with every new article.
- **Implicit Feedbacks:** Detailed session information needs to captured for collaborative recommendation.
- **Scaling Up:** ORDBs are more suitable than RDBMS, especially as website activity scales up.
- **Security:** At present the API might be vulnerable to against various types of injection attacks.
- **Code Stability:** The current Model Creation program will fail if no articles are present in the database.
- **Business Logic:** Similarity combination weights can be generated by parametric estimation algorithms.
- **Hybrid Recommendation:** Collaborative model results can be combined with content based suggestions to generate hybrid recommendations.

CONCLUSION

As already indicated, the purpose of this internship was to build a working POC of a recommendation system that would be best suited for EEIP's business model and current state of affairs. After application of various models it can be concluded that considering the absence of good quality historical interaction data, collaborative recommendations are not an option. However, content based similarity has shown promising results. Content RE APIs are ready to be put to use into the production environment. Successful creation of a Collaborative model can make way for further customised hybrid recommendation engines.

REFERENCE

http://yifanhu.net/PUB/cf.pdf

https://realpython.com/build-recommendation-engine-collaborative-filtering/

https://github.com/benfred/implicit

https://towardsdatascience.com/building-a-collaborative-filtering-recommender-system-with-clickstream-data-dffc86c8c65

https://implicit.readthedocs.io/en/latest/als.html?source=post_page-----dffc86c8c65---------------------

https://medium.com/radon-dev/als-implicit-collaborative-filtering-5ed653ba39fe

https://github.com/susanli2016/Machine-Learning-with-Python/blob/master/Articles%20Rec%20System%20Implicit.ipynb

https://github.com/taki0112/Vector_Similarity

https://en.wikipedia.org/wiki/Matrix_factorization_(recommender_systems)