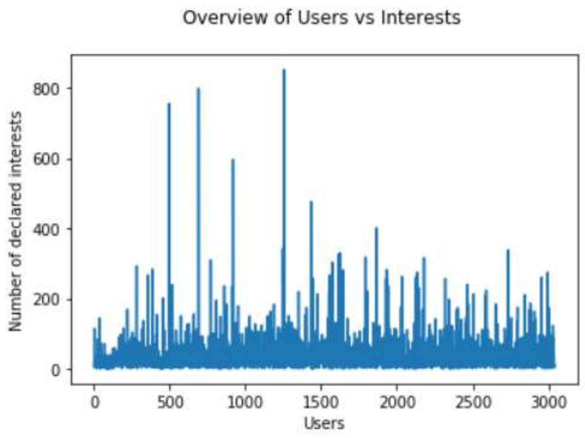
Table of Contents

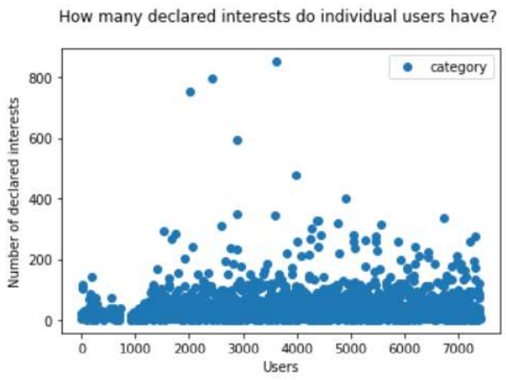
Basic Understanding of Data (Basic Understanding.ipynb)

How many users are there?
How many interest categories were created by users?

```
How many users are there?  3042
How many interest categories were created by users?  883
```

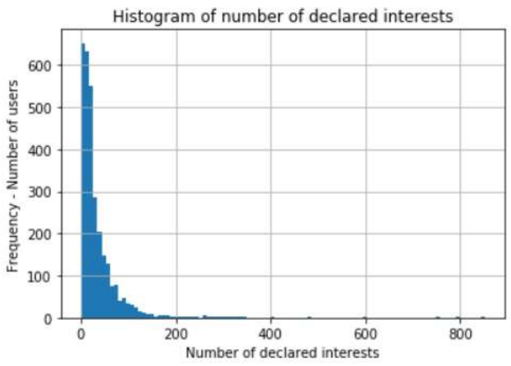
What is the distribution of interests in the entire population of individuals?

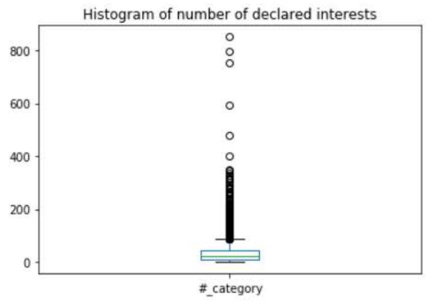




How many declared interest s do individual users have (histogram)?

The box plot shows that on average, a user has about 34 interests declared on his social media page.



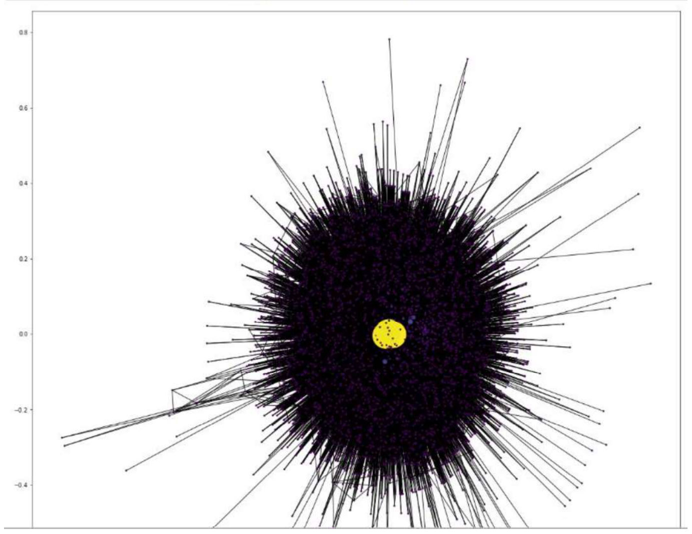


```
count    3042.000000
mean       34.888560
std        49.196073
min         1.000000
25%        11.000000
50%        21.000000
75%        42.000000
max       851.000000
Name: #_category, dtype: float64
```

Determine and plot one or more (social) network metrics such as connection degree distribution <u>(Network.ipynb)</u>

```
Name:
Type: Graph
Number of nodes: 3599
Number of edges: 37809
Average degree:  21.0108
```

```
pos = nx.spring_layout(G)
between = nx.betweenness_centrality(G, normalized=True, endpoints=True)
node_color = [20000.0 * G.degree(v) for v in G]
node_size =  [v * 10000 for v in between.values()]

plt.figure(figsize=(20,20))
nx.draw_networkx(G, pos=pos, with_labels=False, node_color=node_color, node_size=node_size )
```



Some more information about the network

```
Top 10 IDs with most connected friends:
 [ 196.   12.  596. 2418.   36.   26.   42. 6142. 1044. 1110.]

Top 10 betweenness centralities:
 [0.20668776 0.19811286 0.18449293 0.18088544 0.13089571 0.03384159
 0.02511453 0.00826002 0.00613209 0.00537806]
```

Comparison with Twitter, Facebook

Our client's social media channel is more like the Twitter than the Facebook social media. We have the concept of follower-followee whereas Facebook depicts more of a friend sort of relation.

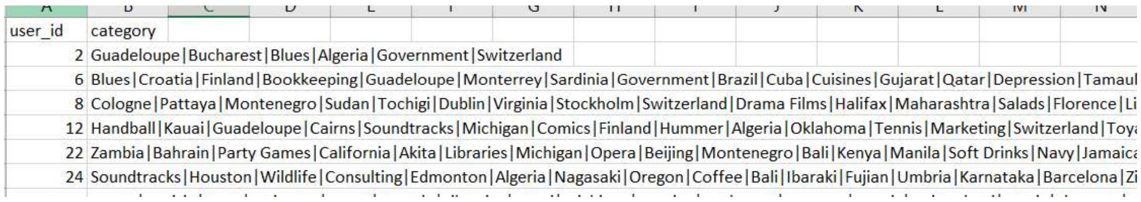
Funnily enough, Twitter called followers friends too in the very early days. Yet, by nature the two things are fundamentally different. Friending someone on Facebook carries a far deeper connection then following on Twitter.

Of course, the action itself doesn't mean anything and you can build meaningful relationships on Twitter in the same way as on Facebook. It feels that *in terms of personal spheres* following someone on Twitter is more detached than a Facebook friend though.

A Twitter following can be strictly interest based on topics the person is tweeting about. Friending someone might show a deeper social interest in the person that comes into play with Facebook. Trying to friend everyone you follow, might not be the optimal solutio**n.**

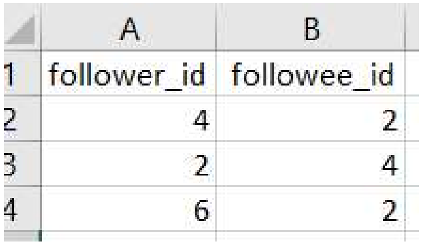
Recommender System
(Reccommender_A_B.py , RecommenderLightFM.ipynb)

I have used excel to manipulate the data (https://www.extendoffice.com/documents/excel/5018-excel-group-and-concatenate.html). Particularly the interest.csv file so that the interests are present in a column in the following manner. Also, the file that we will use for modelling the recommender system will be interest_edit.csv. It has been included in the zip file. I have used the excel formulas to make the data look like following-



user_id	category
2	Guadeloupe|Bucharest|Blues|Algeria|Government|Switzerland
6	Blues|Croatia|Finland|Bookkeeping|Guadeloupe|Monterrey|Sardinia|Government|Brazil|Cuba|Cuisines|Gujarat|Qatar|Depression|Tamaul
8	Cologne|Pattaya|Montenegro|Sudan|Tochigi|Dublin|Virginia|Stockholm|Switzerland|Drama Films|Halifax|Maharashtra|Salads|Florence|Li
12	Handball|Kauai|Guadeloupe|Cairns|Soundtracks|Michigan|Comics|Finland|Hummer|Algeria|Oklahoma|Tennis|Marketing|Switzerland|Toya
22	Zambia|Bahrain|Party Games|California|Akita|Libraries|Michigan|Opera|Beijing|Montenegro|Bali|Kenya|Manila|Soft Drinks|Navy|Jamaica
24	Soundtracks|Houston|Wildlife|Consulting|Edmonton|Algeria|Nagasaki|Oregon|Coffee|Bali|Ibaraki|Fujian|Umbria|Karnataka|Barcelona|Zi

The other file would be follows.csv, which represents the data like following -



follower_id	followee_id
4	2
2	4
6	2

*We have use .py files for the recommender system, as compared to a Jupyter notebook in the previous case. There are two recommender systems I propose

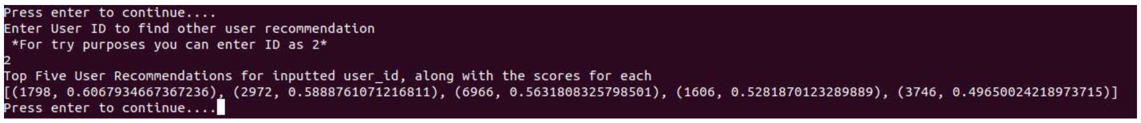
A. Content (Categories/Interests) based recommender system using Cosine Matrix

The Content-Based Recommender relies on the similarity of the users being recommended. The basic idea is that if you like an item, then you will also like a “similar” item. It generally works well when it’s easy to determine the context/properties of each item, which will be categories in our case.

```
Understanding the interests via a wordcloud.........
```

Rotterdam
Sardinia
Croatia
Algeria
Netherlands
Coffee
Tokyo
Guadeloupe
Boating
Mac OS
Yorkshire
Oklahoma
Finland
Michigan
Tamaulipas
Blues
Monterrey
Government

This will give out similar users to a particular user based on the interests. Along with the recommended users, it will also give out a score for each of the user, depicting how similar it is.



```
Press enter to continue....
Enter User ID to find other user recommendation
 *For try purposes you can enter ID as 2*
2
Top Five User Recommendations for inputted user_id, along with the scores for each
[(1798, 0.6067934667367236), (2972, 0.5888761071216811), (6966, 0.5631808325798501), (1606, 0.5281870123289889), (3746, 0.49650024218973715)]
Press enter to continue....
```

The data from this list can be matched from the data from follows.csv to identify the other similar users that a user is not yet following.

B. Collaborative Filtering Recommender using Pairwise Distance

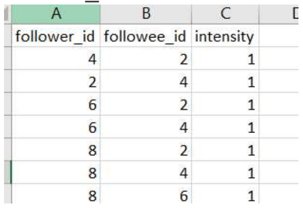
For instance, if user A follows users 1, 2, 3 and user B follows users 2,3,4, then they have similar interests and A should follow user 4 and B should follow user 1. This makes it one of the most commonly used algorithms as it is not dependent on any additional information.

We modify the follows.csv into follows_edit.csv

We create another column which indicates the intensity with with a user follows a followee. In our case, we'll consider it all to be 1, whereas it is not normally the case in real world.

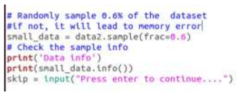
For example, if I like Trump more than Obama, I would give an intensity of 0.9 to Trump over 0.4 to Obama.

Follows_edit.csv data looks like the following-



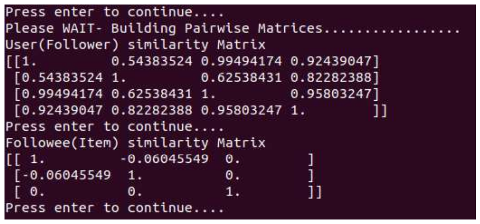
A	B	C
follower_id	followee_id	intensity
4	2	1
2	4	1
6	2	1
6	4	1
8	2	1
8	4	1
8	6	1

Also, we cannot use the whole dataset as it leads to memory error on my computer. So I am sampling 0.6% of the dataset before splitting it into test and train.

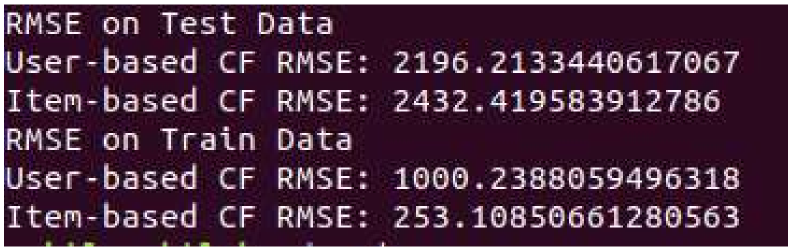


```
# Randomly sample 0.6% of the  dataset
#if not, it will lead to memory error
small_data = data2.sample(frac=0.6)
# Check the sample info
print('Data info')
print(small_data.info())
skip = input("Press enter to continue....")
```

We have split the dataset into test-train and train the model. Instead of depicting the recommended users like how we did in the previous case, we'll try to understand the RMSEs of the different models under collaborative filtering.



```
Press enter to continue....
Please WAIT- Building Pairwise Matrices.................
User(Follower) similarity Matrix
[[1.         0.54383524 0.99494174 0.92439047]
 [0.54383524 1.         0.62538431 0.82282388]
 [0.99494174 0.62538431 1.         0.95803247]
 [0.92439047 0.82282388 0.95803247 1.        ]]
Press enter to continue....
Followee(Item) similarity Matrix
[[ 1.         -0.06045549  0.        ]
 [-0.06045549  1.          0.        ]
 [ 0.          0.          1.        ]]
Press enter to continue....
```



```
RMSE on Test Data
User-based CF RMSE: 2196.2133440617067
Item-based CF RMSE: 2432.419583912786
RMSE on Train Data
User-based CF RMSE: 1000.2388059496318
Item-based CF RMSE: 253.10850661280563
```

C. Collaborative Filtering using Latent Features help of LightFM python module

 I have delved into another model with Follows.csv and latent features in interests.csv
Over here, we form sparse matrices with the main file having followers and followees. Then we feed in the item_feautres which are the categories in the interests.csv

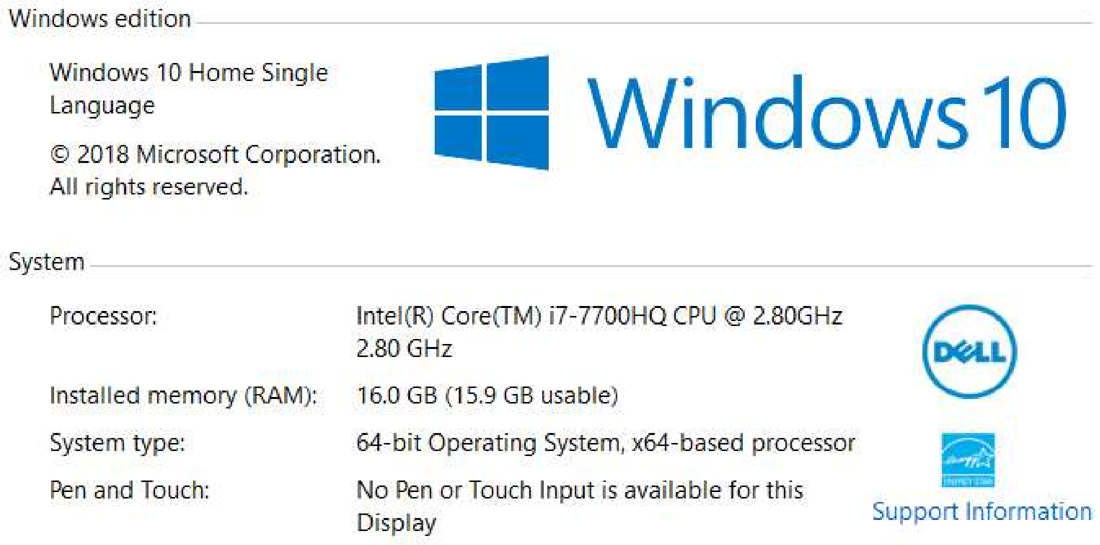
I have build the model using LightFM, but due to lack of resources (time and money), we will not be able to deliver the results of the same to the client. But the model can be found in the jupyter notebook RecommenderLightFM.ipynb

At a high level, describe a strategy the client could use (i.e. next steps) to evaluate the classifier's operation in the context of the client's business

The client should use the recommender system to recommend new users on the social media page by displaying suggestions. Then, using A/B Testing and calculating the click ratio, we can understand how the model is actually performing in real life.

Appendix A

What kind of hardware did you run the analysis on?



Windows edition

Windows 10 Home Single Language

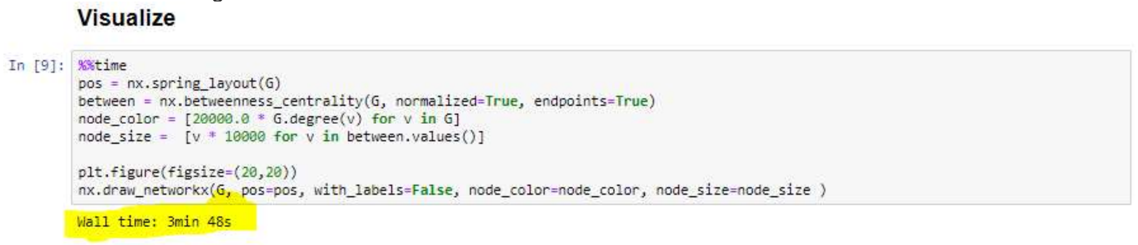
© 2018 Microsoft Corporation. All rights reserved.

System

Processor:	Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz 2.80 GHz
Installed memory (RAM):	16.0 GB (15.9 GB usable)
System type:	64-bit Operating System, x64-based processor
Pen and Touch:	No Pen or Touch Input is available for this Display

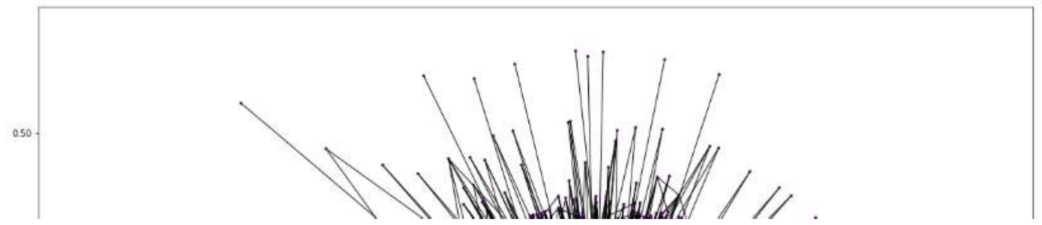
How long did it take? What are your recommendations regarding the volume of data you could manage for further analysis?

The network drawing took about ~4mins.



Visualize

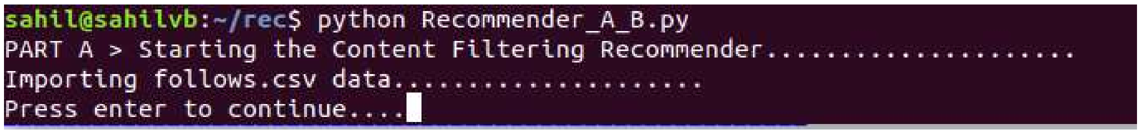
```
In [9]: %%time
        pos = nx.spring_layout(G)
        between = nx.betweenness_centrality(G, normalized=True, endpoints=True)
        node_color = [20000.0 * G.degree(v) for v in G]
        node_size =  [v * 10000 for v in between.values()]

        plt.figure(figsize=(20,20))
        nx.draw_networkx(G, pos=pos, with_labels=False, node_color=node_color, node_size=node_size )
Wall time: 3min 48s
```



We could do feature reduction in the further analysis if we'd want that to increase the computational speed. However, the whole volume is considered in my project for the analysis.

Appendix B

To run the recommender system



```
sahil@sahilvb:~/rec$ python Recommender_A_B.py
PART A > Starting the Content Filtering Recommender....................
Importing follows.csv data....................
Press enter to continue....
```