CSC – 501

Algorithm And DataModels

REPORT

TERM PROJECT: HETEROGENEOUS DATA

SUBMITTED BY: GROUP C

DIVYANSH BHARDWAJ(V0949736)

ARSHIYA GULATI (V00949938)

SHAIMA PATEL (V00949940)

VENISH PATEL (V00949300)

SUBMITTED TO:

Prof. Sean Chester (schester@uvic.ca)

INTRODUCTION

Heterogeneous data, as the name itself defines that we are provided with the variety of data, which contains data of various data types such as Textual data, Spatial Data, and Graph based. To deal with this type of data and linking it concurrently in multiple data models is one of the most challenging thing about this project.

The data was read from the files in the “datascience.stackexchange.com.7z”.

The database we used for storing the tables is SQLite 3. It is better and much more efficient than other SQL engines. SQLite 3 does not have any servers for reading and writing purposes whereas, other SQL engines do have specific servers for these purposes. It directly reads and writes to ordinary disk files. So it does not have the overhead of managing a separate server process. Other than that we used Neo4j which is a graph database platform to load the data from the postlinks table. Out of all the databases that were available for loading the graphs, Neo4j seemed to be quite efficient and faster than the rest and also some great visualisations could be drawn from it.

The language we used for mining and visualizing the data is python, one of the best languages for data scientists to work with. We used SQL Alchemy, a perfect data base toolkit for managing the data in python. It is known for its efficiency, flexibility and full power it gives to the developers. The code for visualisation is implemented using these developer-friendly libraries of python such as Matplotlib, Geopandas, Datetime, Seaborn, Pandas, Numpy, NLTK, Gensim, shapely, pycountry, Geopandas, Geometry, Py2neo and Word2Vec.

The major challenges we faced in this projects were:

- When we inserted the data into relational database we wrote a query which executed only 1 line at a time. But as we had multiple lines we had to use very complex **for** loops.
- We needed to clean the data to bring content into usable form but as we stored all of our tables in database, which was a 2D structure, we had to convert it into a flat list to proceed with our work.
- To get any single visualizations we had to merge 2-3 columns of different tables to get a complete set of information.
- We started working with Tumbleweed Badge to extract user is who were given that badge. For the same purpose we searched the user table to cross — check the ids we found with Tumbleweed badge. Surprisingly, not all ids were present as some of the ids were deleted and hence we had to deal with this inconsistent data.
- When we started plotting location of users it was quite a challenge to

construct a .shape file.

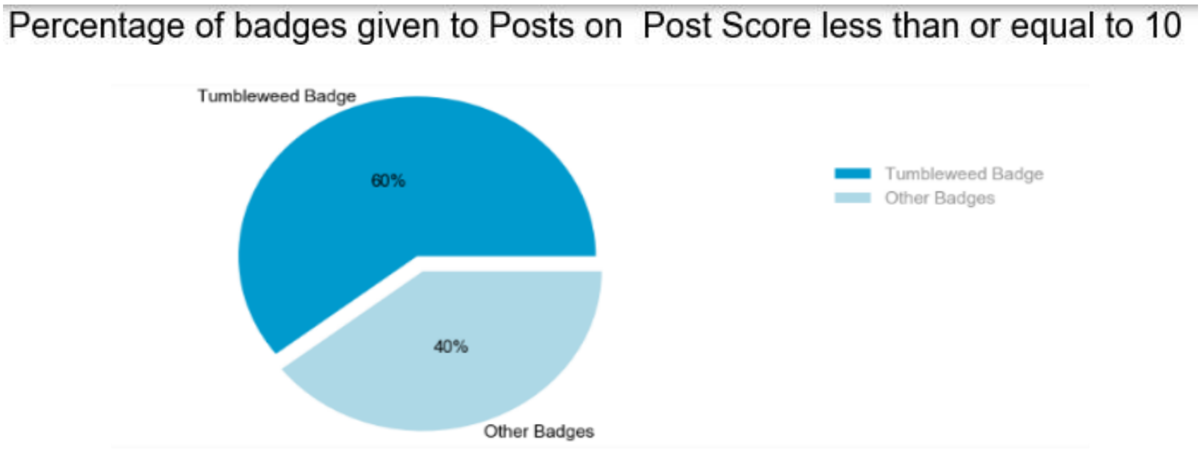
- Connecting the research papers we read to different modelling sections and utilising the concepts of research paper by linking it our course module.

RUBRIC —1: Raw Data To Insights

We started studying the data of xml files to get a better idea of what the columns display and the link amongst them. We got interested in seeing that many posts had a very less score. We wanted to see what badges were associated with these posts.

So that's how we went for our 1st insight.

1. Posts with less Post Score (i.e. <= 10)



So, we plotted a pie-chart to see what type of badges had less than or equal to 10 score. We displayed the Tumbleweed Badge separately because it interested us the most and also it was one of the top 10 used badge. But the output was really interesting to see as 60% users with less or equal to 10 score had a Tumbleweed Badge. This visualization got us interested in seeing the impact of Tumbleweed Badge on users who scored a bit lesser than others.

2. Defining Tumbleweed

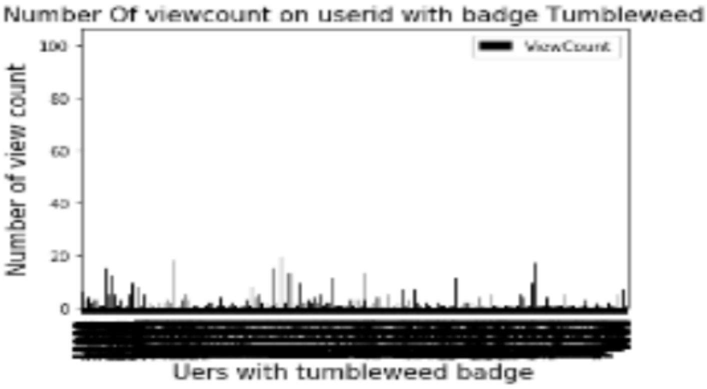
After knowing that the users with the badge had no up votes, down votes and view counts. So we saw the sentiments of those users, but from that we saw that only 23.3% of them had negative sentiment.

First we saw the number of posts along with their User ids with score less than equal to 10.

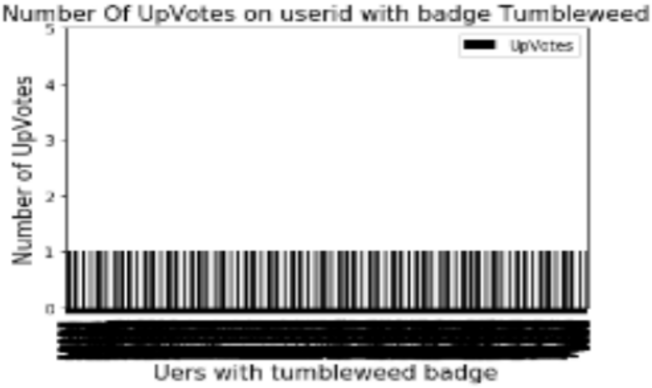
Then we saw the badges given to those users. Visualising this we saw that 60% of them were given the badge Tumbleweed.

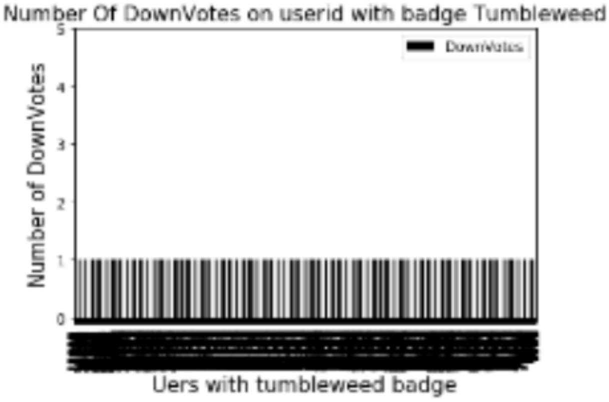
From the pie-chart we see that Tumbleweed is the most given badge to the users with score <=10 on their posts.

Going more in details of this badge we saw that the maximum users with this badge had 0 or one view count.

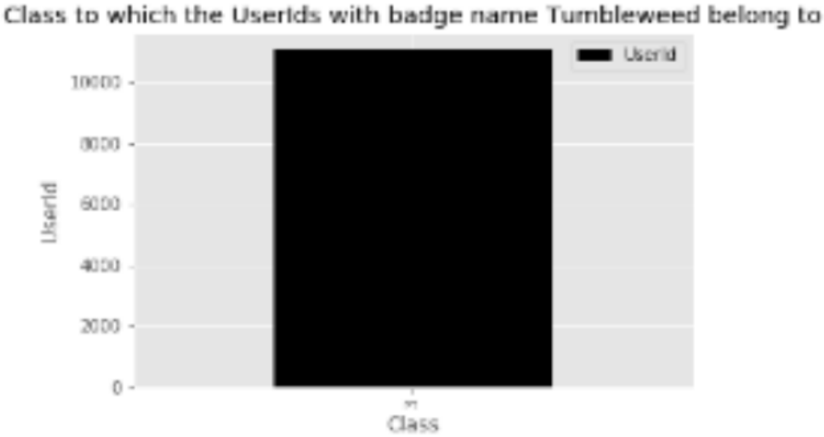


Not only the view count but also the up votes and the down votes on maximum of the posts with the tumbleweed badge were 0 or 1.





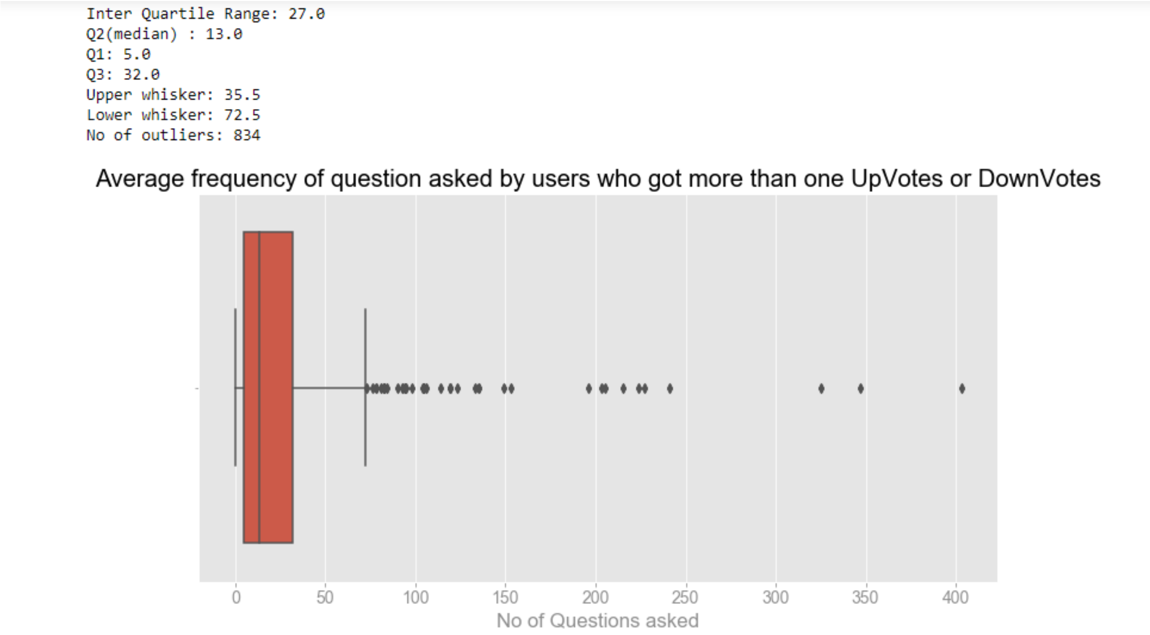
We also noticed that all the user ids with this badge belonged to the class 3 i.e. the bronze badge class, which is given to the users to help teach users how to use the system.



From the above insights we come to know the reason why user ids with score less than equal to 0 have the badge tumbleweed.

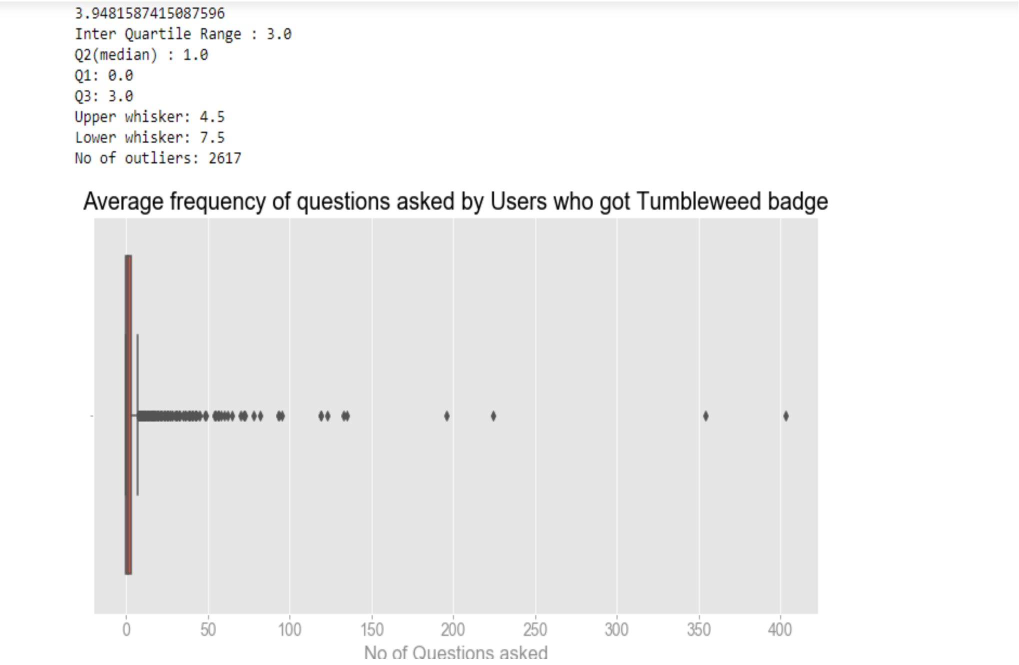
3. Impact of Tumbleweed Badge on Users

As the pie-chart concerned us, we decided to see the impact of Tumbleweed Badge by plotting a Box Plot which displayed average number of questions Tumbleweed Badge users asked compared to those users who had more than 1 upvotes or downvotes. Not surprisingly, the frequency of question asked by users with 1 upvotes or downvotes were 27 on an average, at the same time those asked by Tumbleweed Badge users were only 3 on an average. This shows that Tumbleweed Badge users asked 1/9 less question than other users. This insight raised many question behind the psychology of Tumbleweed Badge users and so we decided to see the reputation of these users.



```
Inter Quartile Range: 27.0
Q2(median) : 13.0
Q1: 5.0
Q3: 32.0
Upper whisker: 35.5
Lower whisker: 72.5
No of outliers: 834
```

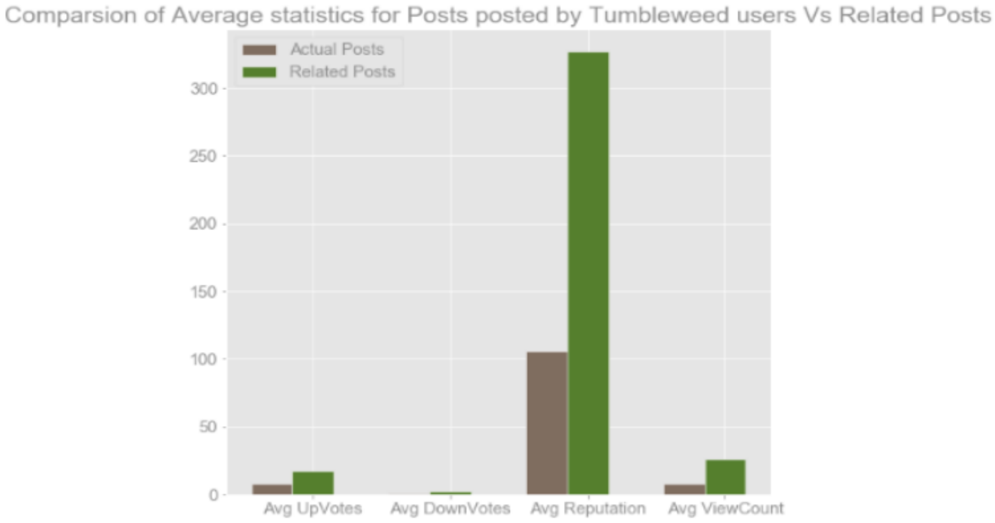
Average frequency of question asked by users who got more than one UpVotes or DownVotes
0
50
100
150
200
250
300
350
400
No of Questions asked



```
3.9481587415087596
Inter Quartile Range : 3.0
Q2(median) : 1.0
Q1: 0.0
Q3: 3.0
Upper whisker: 4.5
Lower whisker: 7.5
No of outliers: 2617
```

Average frequency of questions asked by Users who got Tumbleweed badge
0
50
100
150
200
250
300
350
400
No of Questions asked

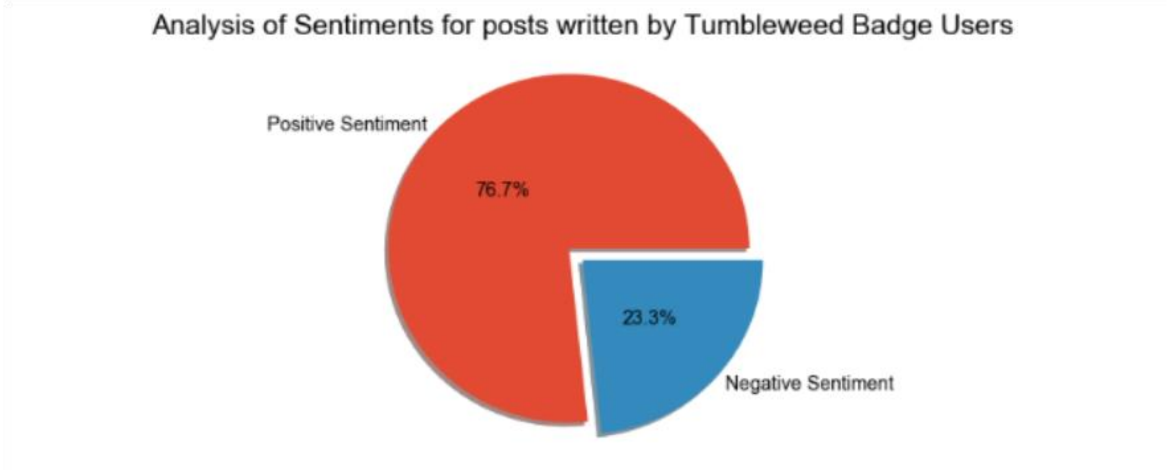
4.Average Statistics (upvotes, downvotes, reputation, view count) of Tumbleweed and Related users



Now we decided to use the PostLinks table to find the related posts. We found the related posts of the posts posted by Tumbleweed Badge users. Once we found that, we compared the users of those related posts with the users with Tumbleweed Badge. Although average upvotes, downvotes and viewcount shows no significant difference, average reputation really amazed us. The average reputation of Tumbleweed Badge users were statistically 3 time less compared to other related post users. The major difference between average reputation amongst these users interested us in knowing about sentiment of their posts.

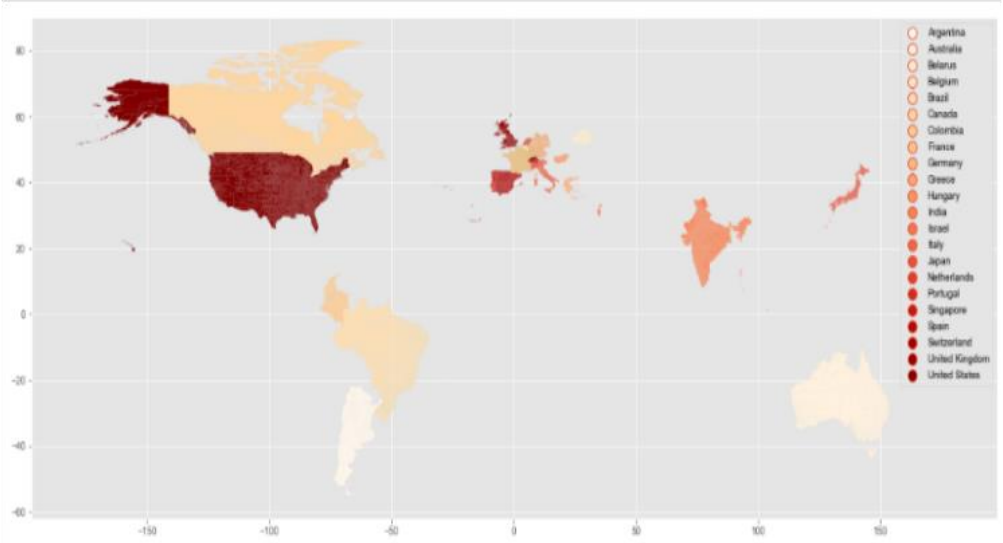
5.Sentiment Analysis on Posts Written by Tumbleweed Badge Users

We were interested in seeing the impact of content posted by Tumbleweed badge users which reduced their reputation amongst other less scored related posts users. But surprisingly, their content didn't have much negative sentiment which confirms that they didn't use any improper or foul language.



6. Location of the users with the tumbleweed badge

From the geograph above we see that the users belonged to the above 22 locations.



RUBRIC — 2: SYNTHESIS

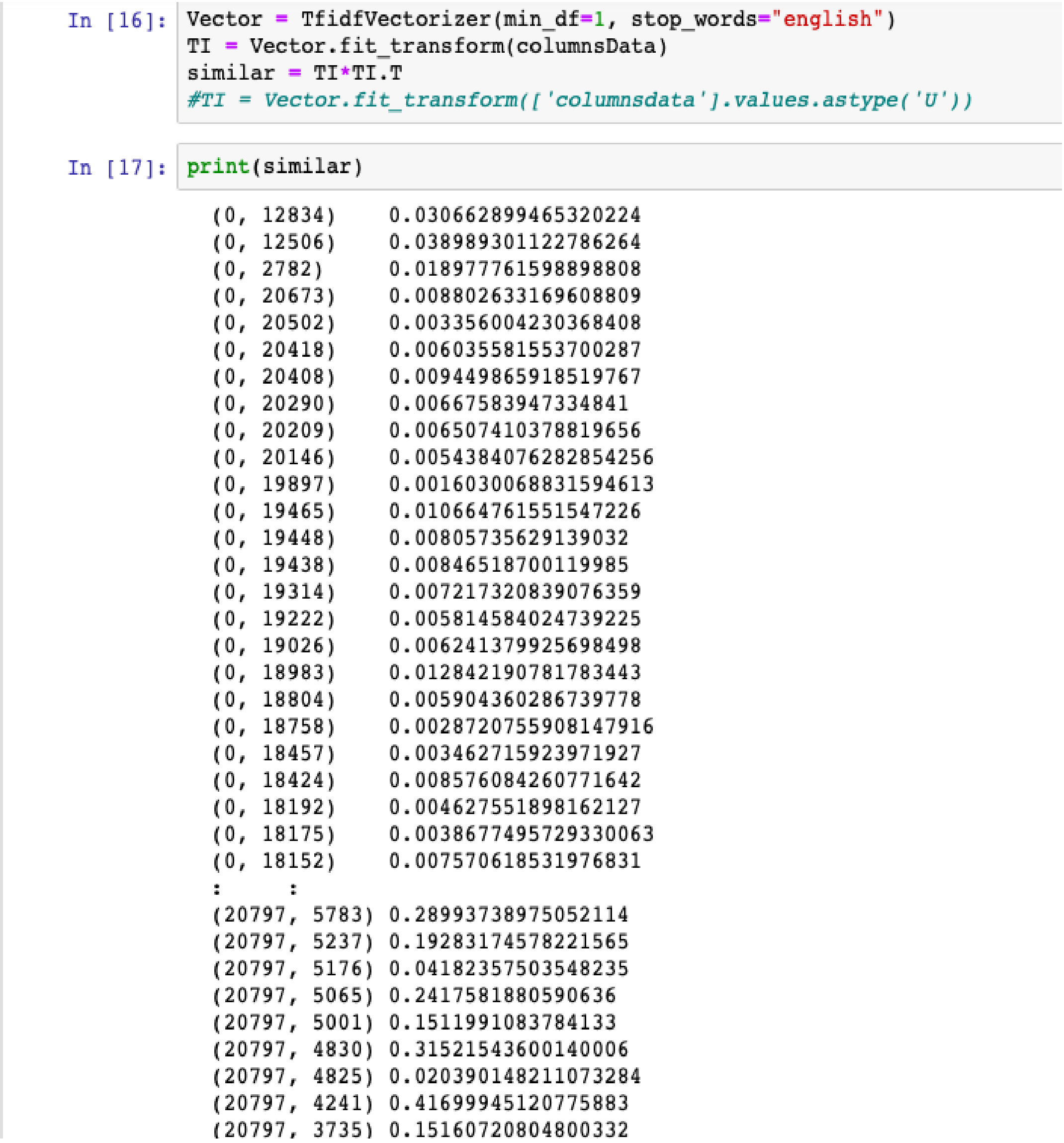
As this rubric dealt with combining ideas with course modules, our main focus was to plot graphs that covered our main data model topics, i.e. Graph, Spatial — Temporal and Text. Relational modelling was done in parallel as data was already normalised, we decided to create a database for all these 8 .xml files using SQLite3 in which we stored the files in the form of tables, after parsing the xml files. Parsing of an xml file is used to read the complete xml document by using library xml.etree.ElementTree.

After obtaining all the xml file data in our db_proj.db we started studying the content to utilize concepts studied in class. We decided to start with Text data as many columns of certain xml tables had text data. Some of them were : Posts, PostHistory, Users, Tags etc. As the tables needed cleaning, we used Beautiful Soup to remove all xml tags like <p>, <li> etc. and stored each line in form of list_words. The set of all list_words was then appended to list_list_words.

As we wanted to clean data we had to convert 2D list into a flat list. After doing that we imported NLTK and for downloading stopwords library. We removed stopwords, hashtags, url, words with length less than 3 and other useless stuff. After getting filtered_sent we created Vector model by using genism Word2Vec and formed Word Embeddings. We preferred word2vec over bag of words as in this context information is not lost and that the low dimension vectors are formed.

1. For utilising text concept we plotted TF-IDF to know similarity amongst users.

We wanted to know which users have similar content. For that we used AboutMe text column from users file. Using the tf-idf library we found the similar users.

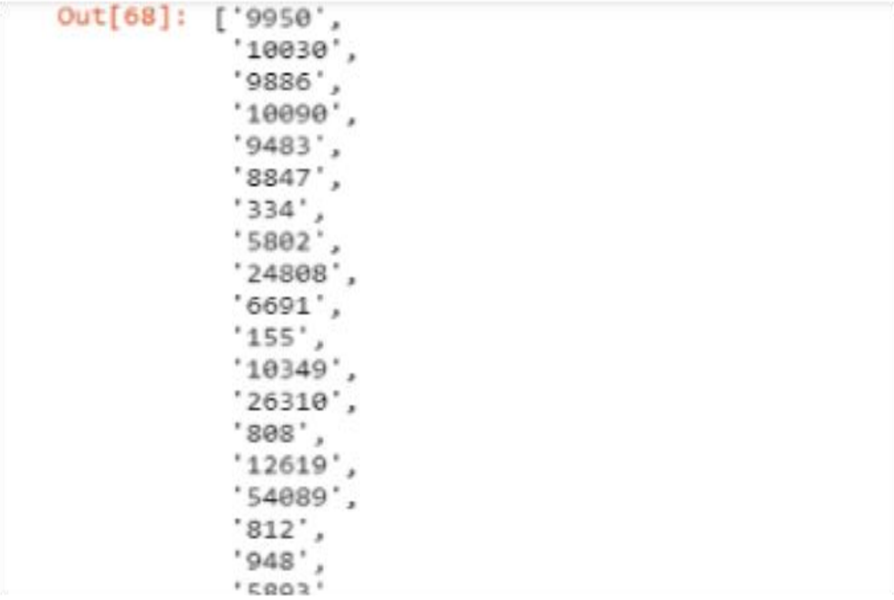


```
In [16]: Vector = TfidfVectorizer(min_df=1, stop_words="english")
         TI = Vector.fit_transform(columnsData)
         similar = TI*TI.T
         #TI = Vector.fit_transform(['columnsdata'].values.astype('U'))
```

```
In [17]: print(similar)
```

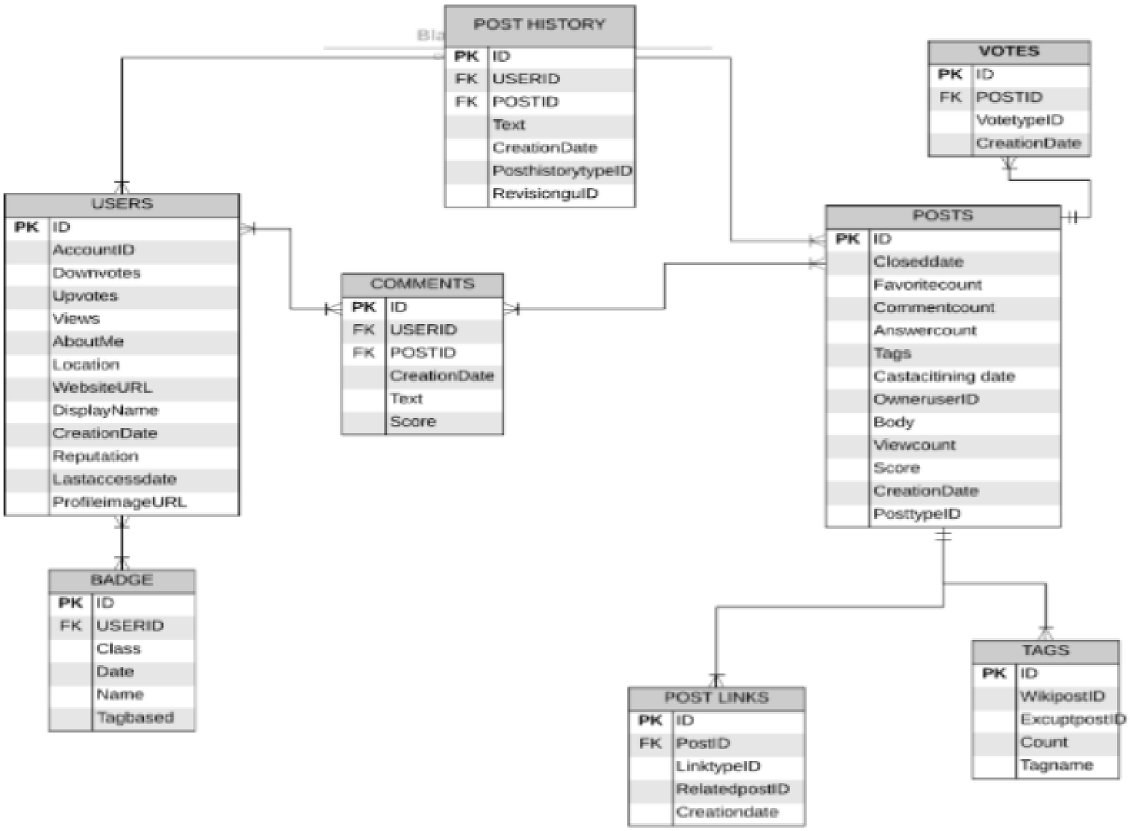
```
  (0, 12834)	0.030662899465320224
  (0, 12506)	0.038989301122786264
  (0, 2782)	0.018977761598898808
  (0, 20673)	0.008802633169608809
  (0, 20502)	0.003356004230368408
  (0, 20418)	0.006035581553700287
  (0, 20408)	0.009449865918519767
  (0, 20290)	0.00667583947334841
  (0, 20209)	0.006507410378819656
  (0, 20146)	0.0054384076282854256
  (0, 19897)	0.0016030068831594613
  (0, 19465)	0.010664761551547226
  (0, 19448)	0.00805735629139032
  (0, 19438)	0.00846518700119985
  (0, 19314)	0.007217320839076359
  (0, 19222)	0.005814584024739225
  (0, 19026)	0.006241379925698498
  (0, 18983)	0.012842190781783443
  (0, 18804)	0.005904360286739778
  (0, 18758)	0.0028720755908147916
  (0, 18457)	0.003462715923971927
  (0, 18424)	0.008576084260771642
  (0, 18192)	0.004627551898162127
  (0, 18175)	0.0038677495729330063
  (0, 18152)	0.007570618531976831
  :	:
  (20797, 5783)	0.28993738975052114
  (20797, 5237)	0.19283174578221565
  (20797, 5176)	0.04182357503548235
  (20797, 5065)	0.2417581880590636
  (20797, 5001)	0.1511991083784133
  (20797, 4830)	0.31521543600140006
  (20797, 4825)	0.020390148211073284
  (20797, 4241)	0.41699945120775883
  (20797, 3735)	0.15160720804800332
```

2. We used to concept of Spatial to know the location of users. We used the users file to know the location and used the .shape file to plot it. We were really interested in knowing the location with maximum users. We used this concept in plotting our 5th visualization.
3. We further stored the postlink data into the neo4j database created to deal with the links in them, in the form of a graph. It was done to make the data more understandable and for more clear visualisations (insights). We used this concept in plotting our 3rd visualization. We plotted an adjacency matrix by taking post in our source node and finding the posts related to it.



```
Out[68]: ['9950',
          '10030',
          '9886',
          '10090',
          '9483',
          '8847',
          '334',
          '5802',
          '24808',
          '6691',
          '155',
          '10349',
          '26310',
          '808',
          '12619',
          '54089',
          '812',
          '948',
          '5803'
```

4. Finally, we plotted database design model entity relationship diagram to know the relation amongst all 8 .xml tables.



RUBRIC —3: PROBLEM SELECTION

The first thing that struck our minds was the posts with lower scores. Side by side we were exploring the Badges table as it seemed quite interesting and insightful to us. We analysed all the posts with Score < 10. After that, we worked on fetching the Users who posted these posts according to their UserId’s given and we computed the badges that has been assigned to these users. Surprisingly, 60 % of the users were getting Tumbleweed Badge. There was something off about this badge that we wanted to analyse in the dataset, also given its name, it seems to be more of an insult.

Immediately, the next question that came into our minds was, what exactly the Tumbleweed badge is. So we made line graphs of users with tumbleweed badge on upvotes, downvotes, views and reputation. One thing that defined it was the count of upvotes and downvotes. For all tumbleweed badge users, upvotes and downvotes count is one. Also we have very less views and the reputation of the users is also relatively less which is quite understandable and obvious.

Now, the next question that we wanted to ask this dataset was the impact of this badge on users. That by any chance, they were getting affected psychologically when given this badge. So, we started comparing two scenarios, one where users were getting one or more than one upvotes or downvotes on their posts, on the other hand were the tumbleweed badge users. We compared the average frequency of questions asked by these two types of users. And voila, just what we thought. The users with more than one upvote or downvote were asking 27 questions on an average, whereas the tumbleweed badge users were asking 3 questions, which is 1/9th of the former category. So, definitely, this badge is affecting the users psychologically. They avoid asking questions.

Now, the next question that followed was, what about the posts that are related to the posts posted by the tumbleweed badge users. We did this by forming an adjacency list where the source nodes were the posts posted by tumbleweed badge users and the related posts were the other nodes. Once we got the related posts, we compared the statistics in a bar graph, average upvotes, average downvotes, average viewcount and average reputation. The related posts were eventually better than the former posts but still, the difference was not much. The significant difference was spoted in the reputation of users who posted the related posts. The average reputation was 3 times better than the reputation of tumbleweed badge users.

Next, we wanted to check, if the low scores of the tumbleweed badge users was because of the negative or offensive language of the users. So we did the sentiment analysis on the body of posts posted by tumbleweed badge users. But, the results suggested that negative sentiment constituted only 1/4th of the whole text. So, the questions posted by them were not in offensive language or had greater negative sentiment.

At last we wanted to see, where these users belong to. So, we took out the locations of the Tumbleweed badge users and plotted the countries they belong to on a world map. Out of all the countries, these tumbleweed badge users belonged to the 22 countries as shown in the legend of the map.

RUBRIC —4 : EXTENSION OF COURSE CONCEPTS

We studied quite many research papers to go through our project.

- We used “Visual analysis of Large Graphs” to model our Graph data by applying general concepts like filtering, aggregation, improving scalability. We also used this paper for graph visualization techniques suitable for visual graph exploration.
- We also used the paper called “Role of Adjacency Matrix and List in Graph” — Harmanjit Singh and Richa Sharma which discusses problems modeled by travelling along edges of certain graph. [1] We used this concept in forming our adjacency

matrix to get related post for a particular post.

- We were quite familiar with text modelling but just to ensure, we read the research paper called [2]“Using Word2Vec to process Big Text Data” written by Long Ma and Yanqing Zhang to form our Word2Vec vector for About Me column in Users table.
- For Spatial — Temporal data we revised our previous given paper called “Microsoft Power BI QuickInsights and Huawei EV Charging Station” which helped us in plotting location of users. Even though we didn’t use the concept of Charging stations and integrating temporal datasets, we covered the

  Spatial aspect of this paper in our project.

References

[1]https://www.researchgate.net/publication/274362141_Role_of_Adjacency_Matrix_Adjacency_List_in_Graph_Theory
[2]https://www.researchgate.net/publication/291153115_Using_Word2Vec_to_process_big_text_data