Chanhee Shin
Kulprawee Prayoonsuk
12/07/20

# Basic Subreddit Recommendation System

## Introduction

Ranking as one of the most popular websites in the United States, Reddit.com operates as a link aggregation site where people come together under specific topics of interest called subreddits. While subreddits are the central basis of the entire website, Reddit.com lacks any sort of subreddit recommendation system aside from its seldom used r/all function which only shows what is globally popular and not at all related to user experiences. This is why our project aims to solve that problem by introducing a basic subreddit recommendation system catered towards users. One of the core functions of Reddit is that the subreddit communities are user derived which fosters a strong sense of community and grassroots movement but comes at a cost of visibility. Many subreddits have obscure names which do not help with searchability and users do not really have a way of finding other subreddits without someone who's already a part of it letting them know. Reddit also does not divulge users the subscription information making this task a far harder goal which we found interesting. This means we need to find a way to recommend to users possible subreddits they would enjoy without knowing what subreddits they are subscribed to.

## Program High LevelOverview

Since Reddit does not store what subreddit each user is following for privacy reasons, the biggest challenge is how can one obtain information regarding the user's interest in order to derive a recommendation from. Since the Reddit API does allow for use to obtain user's comment activity we decided that we would store the the user's comment activity and what subreddit the comment was made on under the assumption that if user comment on a specific subreddit that mean that they have interest on that subreddit.

We use the PRAW api to crawl through subreddits, grabbing user comment information to build the initial network of users and their subreddit vector. After we have collected the information we input a user whom we want to find a recommendation for and once that is imputed the program will find the nearest neighbor node which is the user that has the closest euclidean distance from the imputed user base on interest vector. The recommender will then provide the user with their top recommendation.

**Repository Components**

The repository contain the following files:

- RedditScraper.py
- RecommenderEngine.py
- MongoHandler.py
- Main.py

**RedditScraper.py**

This section involves, using PRAW library to obtain user comment information along with the corresponding subreddit name. Here we specify our parameters such as how many users to store and from which subreddit(s) the base data are to come from. The data is then passed to MongoHandler to be stored on MongoDB database.

**RecommenderEngine.py**

The RecommenderEngine is where the program calculates and processes the users actual recommendations. It first gets a list of all the users in the database and begins to calculate the Euclidean distance between the given user and all the other users in the database. It first creates a vector representation of the user pairs' subscriptions (1 for subscribed and 0 for not) and then performs the appropriate algorithm to find the distance. From there it searches for the K-Nearest neighbor and the Subreddits which are most often shared between which the given user is not already subscribed to. Then we return a list of the ten most recommended Subreddits.

However since the program sources most of its data from very specific Subreddits that means many of the users may actually recommend the source Subreddits to the given user. For example in a situation where the User aligns with several people who are sourced from the "amitheasshole" subreddit would naturally all recommend said Subreddit highly to the user. This is an intended feature but does mean the data is heavily skewed towards recommending at least a few of the Subreddit's the data is sourced from no matter what.

| | A | B | C | D | E |
|---|---|---|---|---|---|
| User1 | x | | x | | x |
| User2 | x | | | x | x |
| User3 | x | | x | x | |
| User4 | x | x | x | | |
| User5 | | x | | x | x |

A basic example is the vector process is as follows. As shown above, Suppose the imputed user is User1, the likely recommendation that they will receive is “D” this is because user1’s nearest neighbor is User2 and user3 as they have 2 “subreddit” in common. User1 shows interest in A, C and E while User 2 shows interest in A, D and E and User 3 shows interest in A, C and D. As you can see User1 have A and E in common with user2 and have A and C in common with user3. Thus UserA would likely be recommended D as that is what their nearest neighbor has in common that they are no longer “following”. In this recommendation if not enough overlap exists then the user will be recommended the most popular subreddit across all users in the database. Which is an approach used in many real world recommenders. This is essentially what is being performed, just on a far larger scale.

**MongoHandler.py**

The section handles connection to mongoDB databases and specifies how data is to be stored.

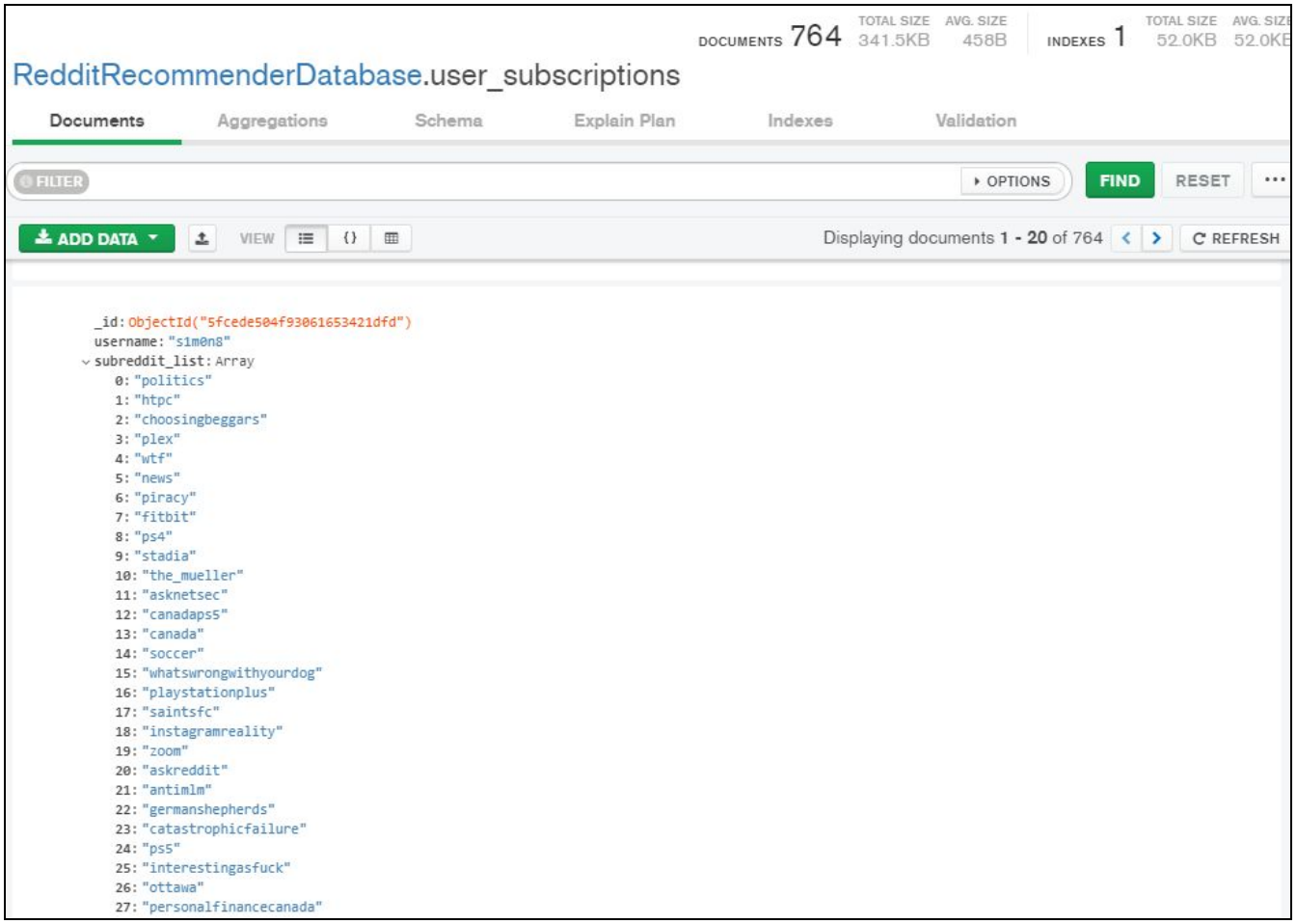
**Main.py**

The section is the “user interface” or the menu of the program where we can run the program and get the result.

**Sample Output**

MongoDBDatabase:

This is an example of database entry for one of the user



User Recommendations:

After selecting the option you will received a list of recommendation

```
C:\Users\ChanH\Anaconda3\envs\RedditRecommender1000\python.exe C:/Users/ChanH/Documents/GitHub/RedditRecommender1000
Options Menu:
1: Reload Data Into MongoDB
2: Get a recommendation for a user
Enter your choice: 2
Enter a username: InnovAsians
['amitheasshole', 'relationships', 'pics', 'gaming', 'formula1', 'guns', 'politics', 'foodporn', 'denver', 'news']
Options Menu:
1: Reload Data Into MongoDB
2: Get a recommendation for a user
Enter your choice:
```

```
Options Menu:
1: Reload Data Into MongoDB
2: Get a recommendation for a user
Enter your choice: 2
Enter a username: 13lood8
['relationships', 'gaming', 'guns', 'formula1', 'pics', 'foodporn', 'askreddit', 'denver', 'relationship_advice', 'ftm']
```

**Observation**

We observe the percentage of users that create new comments on each post is surprisingly low, most users will opt to comment on a pre-existing comment thread instead which may skew the result as we did not include the comment under the original thread due to computing limitations as that would increase the number of data drastically. The flaw of this system is that it takes commenting activity as interest, while in reality the user could be expressing their dissatisfaction with a particular subreddit. As shown in the figure above you will notice similarity in the recommendation, this is because the subreddit that we use as the base for our data will heavily out weight other subreddit as the majority of those people will have the base subreddit in their vector. In an ideal implementation we would get user comment data from many subreddit or possibly every subreddit, but that scale is out of our scope so this project will serve as proof of concept, but as you can see that it is creating a unique recommendation for each user.

Although a more sophisticated algorithm operates under similar principle, user behavior is an important factor to consider when it comes to social networks as an algorithm needs those types of behavioral data in order to be effective. Data science and social networks are truly heavily reliant on the understating of human behavior as well. In addition to this factor another important factor is scale. The more data you can gather and utilize the better your network will be able to make prediction.