Friendship Paradox
Web Science
10 points

The goal of this assignment is to write a Python script which will determine if the friendship paradox is true for you: that most of us have fewer friends than our friends do on average. The friendship paradox was first observed by the sociologist Scott L. Feld in 1991. It is a form of sampling bias in which people with greater numbers of friends have an increased likelihood of being observed among one's own friends. A 2012 study by Pew Internet[1] concluded the average Facebook user had 245 Facebook friends, but the average friend had 359 friends.

In order to perform this experiment, you will use data from your Facebook social network. You can download your network in an XML file by using the NameGenWeb Facebook app: https://apps.facebook.com/namegenweb/

You will need to give this app permission to access your Facebook data. Make sure you select "Friend Count" as an Extended Attribute. When you download the data, download it in the GraphML format. This is a format that many network/graph applications support.

If you do not have a Facebook account, you may use my data set (see below).

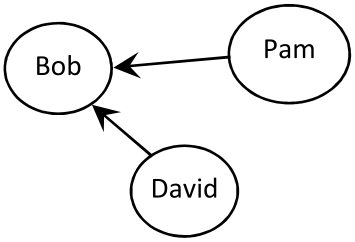
After downloading the XML file, view it in a text editor. Note that not all your friends were downloaded due to some security settings, but most of them have been. Each friend has an entry that looks like this:

```
<node id="Bob_White_5250237">
       <data key="Label">Bob White</data>
       <data key="uid"><![CDATA[5250237]]></data>
       <data key="name"><![CDATA[Joana Carlson]]></data>
       <data key="friend_count"><![CDATA[1261]]></data>
</node>
```

The friend_count data key contains the person's number of friends. Your friends that are friends of each other are represented as edges at the bottom of the file:

```
<edge id="3" source="Pam_Smith_54604435" target="Bob_White_5250237"></edge>
<edge id="4" source="David_Black_71001751" target="Bob_White_5250237"></edge>
```

The "source" is the person who initiated the friendship with the "target". So the social network for the above edges would look like this:



[1] Why most Facebook users get more than they give http://www.pewinternet.org/Reports/2012/Facebook-users/Summary/Friends-of-Friends.aspx

While we could use an XML parser to find the friend count information, it's much easier to just read the file line by line and extract the friend count with the following regular expression:

```
friend_count"><!\[CDATA\[(\d+)
```

which would put the matching number in the first group (because of the parenthesis). Use this regex to count how many friends you have that have more friends than yourself.

The last thing your script needs to find is the largest number of mutual friends. Although the NameGenWeb app count have provided this information in our GraphML file, you will compute this information yourself. To do so, we will use an *adjacency list* data structure to represent the social network. An adjacency list is a list of the nodes in the graph where each item points to the list of nodes that are connected to it. We will use the `defaultdict`, a dict-like Python class that allows you to access keys that do not have values without getting an error.

To create an adjacency list for the edges above, we'd do the following:

```
from collections import defaultdict
graph = defaultdict(list)

# Edge between Pam and Bob
graph['Pam_Smith_54604435'].append('Bob_White_5250237')
graph['Bob_White_5250237'].append('Pam_Smith_54604435')

# Edge between Bob and David
graph['David_Black_71001751'].append('Bob_White_5250237')
graph['Bob_White_5250237'].append('David_Black_71001751')

for node in graph:
    print(node, "=", graph[node])
```

Displays:

```
Bob_White_5250237 = ['Pam_Smith_54604435', 'David_Black_71001751']
Pam_Smith_54604435 = ['Bob_White_5250237']
David_Black_71001751 = ['Bob_White_5250237']
```

Note that we are creating an undirected graph (edges point both ways) instead of a directed graph (edges point one way). Also note that if a friend of yours is not friends with any of your other friends, they will not appear in any of the edges. These friends do not need to be added to the adjacency list, since we are only interested in those friends who have other mutual friends.

Once you have built the adjacency list, you can determine which friend has the most number of friends in common by examining how many friends are in each friend's list. In the example, Bob would have the most friends since there are two friends in Bob's list and only one in Pam's and David's.

To parse out the edge information from the GraphML file, use the following regular expression:

```
source="(.+?)" target="(.+?)"
```

Which will save the source's name in the first group and the target's name in the second group.

Your script needs to output the following information:

```
Total friends: 684
More friends: 193 (28.2%)
Average friend count: 638.1
Most friends in common: Becky_'Pratt'_McCown_781849993 (234)
```

Please use one decimal place in your answers, just like the example above. The first line is how many friends were in the file. The second line is the number of your friends (and percent) that have more friends than you have (you can hard-code the number of friends you have in your script). The third line is the average number of friends your friends have. The last line is the friend that contains the largest number of mutual friends (and the actual number in parenthesis).

The example output above was produced from my Facebook graph file. You can obtain this file here:

\\cs1\classes\Comp475-Web Science\Facebook

I used the number 742 (obtained directly from Facebook) for the number of friends I had when determining how many of my friends had more friends than I did. Since my friends had on average 638.1 friends, I am not experiencing the friendship paradox.

Submit your friendcount.py script to Easel before it is due. Please put in comments at the top of your file the resulting output from running the script on your Facebook data so I can compile a count of how many of us experience the friendship paradox.