TIETS43 Recommender Systems Fall 2017

Assignment 2

Lucas Machado
Student Number 413875

Instructions to run the code

Install Python in your computer if you don't already have it. The code was developed and tested with Python 3.6.3. The packages being used are `pandas` and `networkx`. A default installation of anaconda should already contain these packages.

First, we need to create a Graph object from the txt file. For that, execute the following command *(Assuming that* `facebook-links.txt` *is in the same folder as* `friends.py`*. Otherwise, pass the path as argument)*:

```
python friends.py facebook-links.txt
```

Then, we can get recommendations of friends for a determined user, represented by *<user_id>*, with the following command:

```
python friends.py graph.pickle <user_id>
```

Example

```
$ python friends.py graph.pickle 42

The recommended friendships, in order, are:
85, 1469, 1457, 1458

Out of curiosity, their scores were respectively:
27, 25, 23, 18
```

Assumptions and explanations

A recommendation system like this probably have a noSQL graph database for storing and computing the relations of friendship between users efficiently. To emulate that, I used `networkx` Python package to create a Graph object after reading all of the text file. That Graph object is saved in binary format in the `graph.pickle` file to avoid reprocessing the txt file every time we need to calculate a recommendation. Because of that, the first instruction is to generate that Graph object.

After that, it is fairly easy and fast to use the package functions to find neighbours and neighbours of neighbours from a particular node. Then, we can fetch the common neighbours between two nodes to get the desired score (amount of friends in common).

The information is organised and sorted in a `pandas` DataFrame since it is easy to do data manipulations within it.

Top four recommendations to particular users:

User	Recommendation	Respective scores
20341	14235, 14253, 27408, 22373	74, 73, 70, 63
33722	6958, 29070, 29955, 13819	51, 50, 48, 47
35571	24197, 7959, 19602, 25528	8, 7, 7, 7
25017	11755, 17387, 14358, 27089	33, 26, 22, 22