Electronic Commerce  
Assignment 1 documentation:

The code:

The code is divided into three main modules:

In our code we used the following open source libraries:

Numpy, matplotlib, Networkx

To install this libraries:

pip install numpy

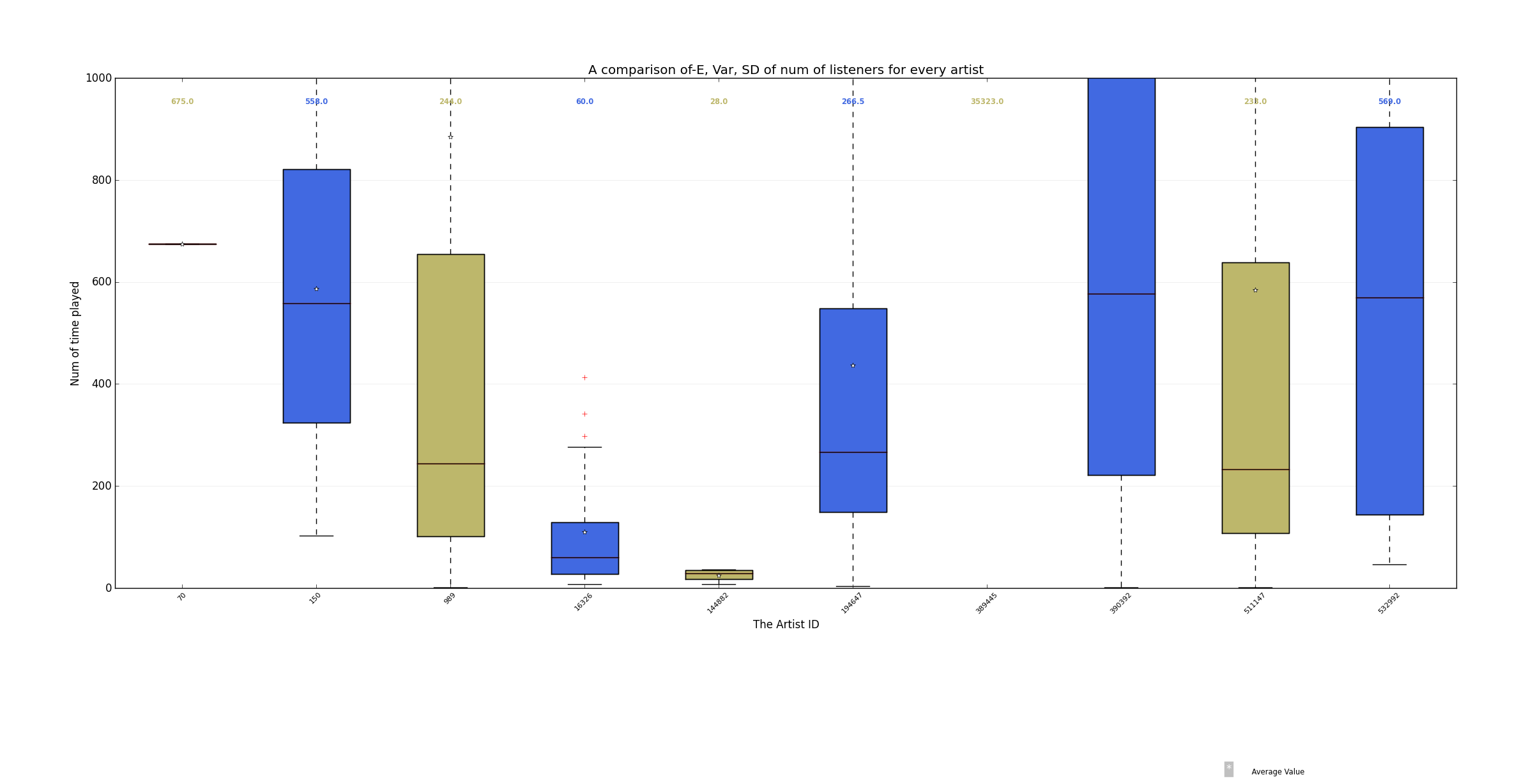
pip install matplotlib

pip install networks (for python 2.7)

Analytics.py:

Step 1:

This module is used to collect the given data and visualize it in a way that allow us to choose the artists with highest sales potential. We did that by looking at properties of the distribution of the listeners. We considered not only the amount of time the song was played but also by how many users.



We also generated the following information:

70:

num of friends:1

min is:675

max is:675

std is:0.0

average is:675.0

median is:675.0

150:

num of friends:4

min is:103

max is:1128

std is:380.729152417

average is:586.75

median is:558.0

989:

num of friends:66

min is:1

max is:28851

std is:3505.868002

average is:884.378787879

median is:244.0

16326:

num of friends:21

min is:7

max is:413

std is:118.314197935

average is:109.476190476

median is:60.0

144882:

num of friends:4

min is:7

max is:36

std is:11.8400802362

average is:24.75

median is:28.0

194647:

num of friends:78

min is:4

max is:1901

std is:448.125232163

average is:437.141025641

median is:266.5

389445:

num of friends:1

min is:35323

max is:35323

std is:0.0

average is:35323.0

median is:35323.0

390392:

num of friends:555

min is:1

max is:114672

std is:6815.19321104

average is:2171.75855856

median is:577.0

511147:

num of friends:229

min is:1

max is:14299

std is:1322.32685942

average is:585.0

median is:233.0

532992:

num of friends:25

min is:46

max is:324663

std is:63435.3445234

average is:13991.8

median is:569.0

Using the data shown above we decided to proceed with the following artists:

['150', '989', '194647', '390392', '511147', '532992']

Step 2:

After choosing the artists we had to find a way to maximize the sale by giving away CDs to the most influential nodes in the graph. To do that we had to learn about methods used to maximize information diffusion in graphs. We came up with the idea to find the nodes with maximum centrality but we quickly realize that it is less efficient in our exercise since we only focus in the first six epochs and this approach will minimize the time to cover the whole graph but it is not grantee to get the optimal result in the first steps.

In the end our approach was to look at the users that has the most friends and out of them to choose the lucky users that will get the free CD. To do that we ran get\_stats**(**usr**,** nett**,** n**=**1**)**

This function calls a recursive function “recursive\_stat” that does kind of DFS (down to depth n) and collect data from all the users in distance n from the root (=usr). We focused on counting the number of users around the root and the waited sum of the amount of listening to every artist. The maximum depth we could get is 4 due to a lack computational power (Intel core i5 1.7 GHz 8GB RAM).

We saved the results in dump file and ran the simulation from that database.

User.py:

This module holds the useful information about a specific user and enable efficient access to it.

Networkr.py:

This module holds the network data structure and contains all the methods to create the graph and maintain the data. Through this module we expend the network using update\_edges and move\_time methods.

Run\_sim.py:

This is the main module after collecting and analyzing the data we can decide how to give away the CDs to do that we chose the 25 users with the biggest circle of friends in distance 4 and from them we chose the users that are also in the group of 15 users with the largest number of close friends (neighbors). We end up with f\_best, the list of potential users:

f\_best, the list of potential users: ['138760', '263144', '375845', '649628', '160451', '909244', '386404', '939238', '468276', '209564', '837172', '965152', '756469'] .

Then for every artist from the list

Leading artists: ['150', '989', '194647', '390392', '511147', '532992']

we decide which user from f\_best will get free CD. We chose the users based on the weighted average of the amount of time the cd was played by the users in the circle of distance 4. For every artist we chose the top 5 users. We generate a dictionary {‘artist name’ : [ list of user names ] }and send it to full\_run to complete the simulation. Full\_run, deliver the CDs to the users we chose and run 6 epochs (include adding connections and propagating the CDs).

Ex1.py:

Finally just a script to create the artist files according to the example provided with the assignment. The dictionary selection is the {‘artist name’ : [ list of user names ] } calculated in run\_sim.py