

CMPT120 - Program 2 (**Prog2YourLastName.py**)

Due: Tuesday, Sept 18, before 1:30 pm. (Submitted via iLearn and printed)

On my desk (in the classroom) beginning of class

The main purpose of this assignment is to get comfortable when writing your own program and computing with numbers.

Social media sites such as Twitter use mathematical algorithms to help sort through the millions of tweets to decide which tweets to show to a particular user. For example, Twitter provides a service called promoted tweets in which an advertiser would pay to have its tweets about a particular topic to appear at the top of the list. For example, the *Perky Beans Coffee Company* might pay to have its tweets appear at top of the list when users search for tweets about the topic coffee. Sounds simple – pay money to have your tweets at or near the top of the Twitter search results. However, from Twitter’s point of view, they would not want promoted tweets that turned out to be lame or duds to stay at the top of its search results for very long. This is where a mathematical algorithm comes to the rescue.

What factors might Twitter use to determine a score or rank for a promoted tweet which would determine how close to the top of the search results list it should appear? Here are some possible factors:

- Number of re-tweets (**numReTweets**) – higher value is better
- Number of @ replies (**numAtReplies**) – higher value is better
- Number of days (**numDays**) since this tweet first appeared – lower value is better

The number of days portion of the tweet’s score might be computed using a time decay function, meaning that the older a tweet gets, the lower its score becomes. Here is one possible time decay function:

$$\text{dayScore} = 2 / (\text{number of days})^3$$

Twitter may decide that one of these three factors should carry more weight or influence in the computed rank so they would assign a weight to each of the factors. The sum of the three weights must be 100%. So Twitter might use this equation to compute the score for a particular promoted tweet:

$$\text{score} = \text{reWeight} * \text{numReTweets} + \text{atWeight} * \text{numAtReplies} + \text{dayWeight} * 2 / (\text{numDays})^3$$

For our program, we will simplify the formula and assume that each of the three factors receives an equal weight of 33.33 percent.

$$\text{score} = 0.3333 * \text{numReTweets} + 0.3333 * \text{numAtReplies} + 0.3333 * (2 / (\text{numDays})^3)$$

For example, a run of your program might look like this, with typed user inputs in **bold**.

This program will compute the score for a promoted tweet.

Please enter the number of re-tweets: **50**

Please enter the number of @ replies: **10**

Please enter the number of days since tweet appeared: **2**

Your tweet’s score is 20.081324999999996 (higher is better)

In this case your program would evaluate the score equation as shown below.

$$\text{score} = (0.3333 * 50) + (0.3333 * 10) + (0.3333 / (2)^3)$$

$$\text{score} = 16.665 + 3.333 + 0.083325 = 20.081324999999996$$

If we run it again with different data, we might get:

This program will compute the score for a promoted tweet.

Enter the number of re-tweets: **100**

Enter the number of @ replies: **200**

Enter the number of days since tweeted: **10**

Your tweet's score is 99.9906666

(You should do the math on this one to check to make sure the output is correct.)

You can perform the score computation in a series of steps:

(1) Compute re-tweet portion of the score, for example,  $0.3333 * \text{number of re-tweets}$ .

(2) Compute @ replies portion, for example,  $0.3333 * \text{number of at replies}$

(3) Compute time decay portion, for example,  $0.3333 / (\text{number of days})^3$

(4) Find the sum of parts (1), (2), and (3).

Before writing your code, consider the software development process. Answer the following questions within your code as comments:

1. What is the problem trying to solve? (Make this answer the Purpose Statement in your header comments.)
2. What are the input/output specifications? (Make this answer a part of the header comments.)
3. Design a solution. (A lot of this has been done for you in the description of the problem above. Use this design to comment your code.)

Save your program as **Prog2YourLastName.py**

**File to be submitted:**

**Prog2YourLastName.py**

All identifiers should be meaningful. Include your design (pseudocode) as comments in your program. Be sure to include the FULL documentation header (as described in Prog1)!!!!!!