If Quiz 6: Data Streams [100 pts]

Problems in this quiz are based on code in my github repo. To begin,

```
git clone https://github.com/singhj/big-data-repo
```

1. Querying Users [30 pts]

This problem is designed to be solved on a Linux machine (GCP is not required for it). The code for this question lives in `big-data-repo/spark-examples/first_streaming`. README.md contains the directions for producing and consuming queries entered by 6 different users[1]. After sampling the stream for some time[2], determine the average number of duplicate queries issued by any user. How well do your findings compare with expectations?

2. Bloom Filter [40 pts]

This problem is designed to be solved on a GCP cluster with 0 workers running Pyspark. The file `big-data-repo/spark-examples/drunk-speech.py` emulates the speech of a person speaking gibberish laced with obscenities that rate -4 or -5 in the AFINN word list. Your task is to

- [10 pts] Create a Bloom Filter[3], approximately 1000-2000 bits in size, for detecting bad words[4] (i.e., AFINN of -4 or -5). It should be designed to run in Spark.

---

[1] The query stream looks like this, where sndr0000, etc. are user IDs and qry4752 etc are query IDs.

```
At 2024-10-22 00:29:57.832525, sender sndr0005  said: qry2717
At 2024-10-22 00:30:01.958333, sender sndr0001  said: qry5919
At 2024-10-22 00:29:57.916552, sender sndr0001  said: qry4752
At 2024-10-22 00:30:02.587708, sender sndr0000  said: qry1106
At 2024-10-22 00:30:02.692793, sender sndr0004  said: qry1806
At 2024-10-22 00:30:02.697882, sender sndr0003  said: qry4941
At 2024-10-22 00:30:03.155397, sender sndr0002  said: qry5082
At 2024-10-22 00:30:04.395581, sender sndr0003  said: qry3255
```

[2] The phrase “some time” is deliberately vague. You’ll need to read the code for `click-feeder.py` to decide what’s appropriate.

[3] There are many implementations of Bloom filters on the internet. You can consult them if you wish but *the purpose of this exercise is to write your own.*

[4] There are about 65 such words. If we wanted the bit-vector to be no more than 10% populated with 1’s, how large would the bit-vector need to be?

- [10 pts] The bit vector should be placed in HDFS as a Base64-encoded[5] text file and loaded into Spark from HDFS.
- [15 pts] Integrate the Bloom Filter into Spark such that every arriving sentence is examined and passed along if none of the words in the sentence are bad words. Sentences that do contain bad words should be suppressed.

- What to submit:
    a. Your Spark (or pySpark) code,
    b. A printout of the Base64-encoded bit-string, and
    c. A recorded video session showing the streaming filter in action. The session should be no more than 120 seconds in length. [To do this, create a zoom meeting, set it to *record to the cloud*, plan what you are going to say, then start the meeting with just yourself, share the screen, and go through the demo. Soon after you end the meeting, zoom will send you a recording URL, which you may watch – and submit the URL.]

3. Counting Unique Users [30 pts]

This problem is designed to be solved on a Linux machine (GCP is not required for it). The file `big-data-repo/spark-examples/first_streaming/click-feeder.py` can be reconfigured to emulate any number of users typing queries. Modify read_stdin.py to implement the HyperLogLog algorithm. Increase the number of senders and decrease the (μ,σ) of the delay between queries until the receiver can no longer keep up! Draw a graph of the estimated number of users as a function of elapsed time[6].

[5] Why Base64-encoded? See this.
[6] You may use any graphing technique, going from graphing on paper to using any graphing package.