Data Technology Group 1 Trendiness Project Midterm Report

Part A:

Current Milestone Progress:

Milestone 1- Milestone I is near completion. The code just needs to be cleaned up to get rid of testing logic, and Part D needs to be completed. In addition, we will have to write out how to run the code in our README.md file.

Milestone 2- Milestone II is in the middle of completion. We successfully wrote the tweets to the database, but now we just need to evaluate the trendiness score. Parts A, B, and D are finished. It needs clean-up, however.

Milestone 3- Milestone III is waiting on the success of Milestone 2 to move forward. We want to make sure the trendiness score is accurate so we can use the same schema and compute the trendiness score.

Team Member Contributions:

Logan Butson: Tested error or failure scenarios for Milestone 1 subparts D and created the Failure.md documentation. Also have begun working on storytelling approach and artistic slide deck for final presentation.

Chelsi Gondalia: Contributed to milestone I, specifically worked on initial thoughts for subparts A and B and then to code and draft final versions of subparts B and C.

Keith Hines: Helped brainstorm and design schema for Milestone II Part A. Currently working on taking a word as an input and calculating the frequency of that word for Part C of Milestone II, and then on calculating trendiness score for Part E.

Matt Plantz: Assisted Hao and Chelsi on milestone I parts A and B with initial research into the problem. Designed SQL database schema in milestone II part A and worked with Hao on milestone II part B. Assisted Keith on milestone II C. Created the python script for Milestone II part D.

Hao Tran: Played a role in each of the milestones, mainly contributed to figuring out how to parse the words from each tweet. For Milestone I, I created the script to write the tweets to a text file (Part A), and I worked with Chelsi in computing the word count for Part B. I am also working with Logan on documenting failures/recovery logic. For Milestone II, I helped parse the tweet and write it to the SQL database according to the schema written by Matt and Keith. For Milestone III, I worked with Xinyuan to write tweets to the Kafka queue. Currently waiting on computing trendiness score before we can continue writing it to the sql database for Milestone III.

Daniel Waller: Took up the role of project manager, assigning group members to different milestones, setting schedules, and facilitating communication between group members. Acted as a sounding board for teammates when discussing how to approach new steps of the project.

Xinyuan Zhang: Used tweepy to write Milestone I Part A, and wrote Milestone III Part A. Made changes to word_count.py for it to detect real phrases. Currently working on Milestone III Part B.

Project Challenges: Working with multiple people on the same task is challenging. It felt like two people's effort being coordinated to work on the same thing was difficult. Since we did not have official deadlines for each Milestone, we could not determine what kind of pace we were at to finish the project.

Parsing the word from each text was very difficult because of our knowledge with regex libraries. We never talked about it in class, but the regex library helped us parse the data out. If we understood how to clean text using this library in lecture, it would have helped everybody.

We also did not understand at what capacity our error handling should be. There was some confusion in determining how much recovery code we should put in because we can only do so much. Our conclusion is that we were going to stick with one error-handling code and discuss 5 issues to recover from in the FAILURE.md file.

Future Suggestions: We could use deadlines for each milestone. Right now, it seems like we are working on Milestones in parallel because everything is getting pushed off until one final due date. If we were also given specific libraries to work with, that could have helped our search in writing the code. Having teams review their peers in forms at the end of the project might serve as a fairer way to judge participation.

Part B:

Data Lake versus Data Warehouse:

For a company interested in a Twitter trendiness score, we suggest deploying a data lake rather than a data warehouse. A data lake is the superior option because a data lake would allow for the data to be transformed as it was received via an ETL process. In comparison, a data warehouse would require an ELT process where the data would need to be transformed for analytics after being loaded to the warehouse. A data lake would also allow for more timely data as the data would be processed and made available as it arrives. In contrast, a data warehouse would need to be loaded in batches which would introduce a time delay on the data thus delaying any real-time analytics on the data. A data lake would suit this purpose better because it allows for real-time analytics and does not require transformation after being loaded. If the goal of the analysis is determined, then we should use a data warehouse due to warehouses being better structured and easier to query. This is what makes them useful for our project, as we are using them to analyze specific words and phrases. However, if the purpose of the analysis is explorative, as we believe it to be, then we should use a data lake, which allows for more diversification in the format of data stored within them, including unstructured data.