Team 3 Project 2

Sprint Retrospective:

1. **What went well:**

   Group collaboration and communication went well for this project as we made sure to confirm our ideas with each other to ensure that we were all on the same page. We were able to follow the requirements to the best of our ability and present a finished and polished ETL pipeline that worked!

2. **What did not go well (things you would avoid in a redo):**

   We spent more time than necessary trying to go too in depth in our data cleaning and handling, especially for the payload flattening. At times, the cluster was not powerful enough to handle everyone's notebooks running in parallel, and it would take us a long time to check if our code worked. In a redo, from the start, we would spend the first week designing out our medallion layers structured around our queries so we could have a clear plan for what our tables would look like before starting cleaning. In addition, we didn't know about the github docs page for days on end and things only cleared up after discovering them.

   In the data, there were multiple places where fields such as user_id and repo_id existed in multiple places in different nested levels and cross checking these nulls to fill them in were done to an extent. However, given more time, we could have gone further in depth to fill in these nulls appropriately.

3. **What could improve for next time:**

   Regarding partitioning, we didn't implement caching in a few places in which there were multiple actions followed, so our runtime was not as efficient as it could have been. Having the opportunity to have all the layers tested off a PoC by the end of week one would allow for deeper learning. We also had a databricks exam that split our attention for the first week. Our team felt that if we did the project before the databricks exam, it would have been more helpful. We memorized the syntax for the exam, but this did not help us understand how to use the functions for the project. The Monday exam from Skillstorm was useful, but taking the cert exam after the project could have been useful.

   In our queries, we conducted operations such as groupBy and aggregation. However, within the Spark framework, there are more efficient alternatives, such as employing reduceByKey() and aggregateByKey(). Another optimization step that we could have taken was adjusting the autoBroadcastJoinThreshold to enable broadcast joins (event_table join actor_df, but actor was 7 times larger than the default 10 MB threshold). If we were to revisit this project, we would prioritize employing these more efficient functions to streamline our processes. Although we did employ persist() to cache data frames in several instances within our notebooks, we recognize that further enhancements could have been made. For instance, we could have optimized

caching by strategically placing it after joins, especially when followed by extensive aggregations, or by performing repartitioning prior to groupBy operations.

4. **Other comments on the project**:

   Having a poll for the project topic could be helpful as people had different levels of understanding coming into the project and understanding the data set would be more difficult for some than others.

5. **What hindered your progress in the project:**

   Understanding the dataset took us a few days of time to feel fully comfortable. In addition, the cluster speed made cleaning and manipulations slow, and our PoC only had two days worth of data. This made us uncomfortable with our decisions to handle nulls and duplicates.

6. **Things you learned from the project:**

   For our gold layer, we followed the star schema and decided to create it in regard to our aggregations and analysis, but upon creating the analysis, we realized that we overlooked a few useful columns and kept some irrelevant columns. When we started looking at our queries, we didn't have enough resources to go back and change our gold layer. We learned that communication between three people is also challenging as everyone needs to share their ideas and come to an understanding with each other. We did feel that everyone was satisfied with the end product, and we worked well as a team!