1. The existing pipeline was designed to be able to handle online transaction processing. One example is that they calculated processing\_time within the pipeline. This allows for additional features and calculations such as the fraud score calculation to be implemented relatively quickly. However, it is also a very low level ETL tool compared to other popular platforms (Informatica and other UI-based ETL software) thus making it somewhat difficult for just anyone to learn and to catch up to what the pipeline does.

2. A Transaction ID should be included in the transaction table to be able to keep track of and evaluate individual transactions. It does not also seem to be able to handle multiple kinds of items in 1 row. If the intention was for a single transaction with multiple items to be pushed as individual rows per item, then we definitely need a proper transaction ID to group together these individual rows. Fraudulent purchases should also be audited / verified to be actually fraudulent in order to help the ML algorithm learn and to prevent future false positives.

3. Business users should be aware of the general flow of the pipeline. The main component to stress here would be the calculation of process\_time as it is done within the pipeline itself. If transaction process\_time is an important metric for them, then they should understand what could influence the calculation of it. For example, if the order was created some time and there were network issues influencing the database connection with the pipeline, would this negatively influence process\_time? How would this affect their business metrics?

4. I think the simplest way to backfill fraud scores would be to implement the updated streaming pipeline first. Then once the first entry with fraud score comes in, we can note the time it was created and create a new job to fill in all entries older than the time of the first new entry.

5. They will probably want to be able to audit some of the less confident fraud scores and to keep track of all this new information in some way. Accuracy of the newly implemented ML fraud detection should be evaluated and appropriate measures to deal with false positives should be implemented as well.

6. Firstly I believe the client would require an auditor for fraud detection. Whatever changes that needs to be made to the algorithm should be something that the client is responsible for. Anything that is miscalculated or the like should be taken up with the Data Science team as soon as it is spotted and with proper data as well for us to refer to. Fraud alerts will also likely be changed around some amount of time as people leave the company or new people join the team, those who need to be alerted to fraudulent activities will also change. If the alert is in slack then minimal maintenance is required as people can be added and removed from channels as needed. Maintenance is required for Emails. I imagine there will be a CSV file where they can update the recipients of the alert. Any plans to add new features will need to be evaluated first by the client tech team then passed onto us if needed.