**Case 3**

In the case of many data sources, you need to have implemented good monitoring systems in the data ingestion section in your machine learning pipeline. In this case, you always want to have the most recent data possible. This is a version of weighted sampling, where the most recent data is most valuable to you. This could mean that you have implemented alerts in your monitoring system, that give an alarm periodically when new data is available. The alert might then trigger collecting the data from the source and further process it and output it in to update your models for example. Kafka and Kinesis for example allow you to transport the data from applications to data warehouses.

This is a task that happens in the data preparation part of your machine learning pipeline. Once the system knows that new data is available, the system need to trigger data update procedure, whether it is labelling, normalization, transformation and so on. This requires solid monitoring solution in the data ingestion segment and solid pipeline to continually evaluate what kind of data is ingested and from which source, and what updates the data needs.

Preparing for things braking up is a task for all the parts of the MLops pipeline. This requires good observability from the pipeline, which means setting up the system so that it gives visibility through the whole system to help investigate what went wrong. This includes having good monitoring practices, whether it is for tracking and counting missing values in the data, adding timers to the functions, or tracking and logging how inputs are transformed through the system. In this case it could mean monitoring the prediction models for distribution shifts for example by summary statistics, two sample tests and visualizations of the prediction distribution. Which in turn could use alarming systems and a troubleshooting dashboard. This system could also benefit from monitoring the changes in features that the models use as inputs and monitoring the transformations from raw inputs into final features. It would also be useful to monitor the features in a way that they stay inside certain limits and follow certain distributions. We could assume, that in this case our models would not be the most complex, so that model performance degrading would not be our main concern, but the quality of the data and the features.

Models and information on the info board would need updating for sure. To do so we need the new data. This could be achieved collecting, logging, and updating the data automatically periodically, then in certain time intervals retraining the models and updating analysis texts. This of course requires again good monitoring practices and solid ML pipeline.