014487800, 014997604

The case here is that the customer maintains an info web site that has multiple different data sources that update with varying frequencies. So how to keep track of the new available data and how often update your own site with that data?

This is a task that takes place in the data ingestion and preparation part of one's machine learning pipeline and in this case as an info site it is important to have as recent data as possible. One way this could be achieved would be to implement alerts in one's monitoring system so that it alerts you either by a pre-defined time interval or whenever certain amount of new data would be available. This alert would the trigger one's data update procedure, whether it is labelling, normalization, transformation or so on depending on the data and source at hands.

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. One should also monitor model performance and data drifts to be aware of possible issues in one's ML pipeline as soon as possible.

If you interpret analysis texts as something like "...1st quarter was like that and the 2nd like this..." these texts could be automated to be updated while new data is introduced to this info web site. Certain phrases could be conditioned to certain data ranges or similar and when updating data these texts could be generated again based on new data.

What it comes to predictive models on this site, these models could be something like "…next quarter will most likely be nice…". These models could automatically be updated whenever new data regarding each model would be introduced to the system. How and how often these models would be updated depends heavily on the model and with that kind of data it is using; is it real time, is it batch like data, how often is the information relevant that the model produces etc.

Evidently would be suitable for setting the alarms and to monitor the data sources and data downloaded. Kafka and/or Kinesis could be used to handle acquired data and prepare it for downstream use. MLFlow could act as a space for downstream use and include model and experiment tracking. This all, including data requests and download procedures, could be orchestrated and organised in different modules using Kubernetes.