Assumptions:

I am uncertain if the question is asking for a model monitoring pipeline in general or for this ASR model. I have chosen to answer this with the ASR model in mind.

Essay

Model drift occurs when the data the model encounters changes over time, causing a decrease in its performance. For an Automatic Speech Recognition (ASR) model, drift can happen due to variations in accents, background noise, or changes in audio quality. For example, if the ASR model is deployed in a new region with different accents or it processes audio from noisy environments, its performance could degrade, causing more errors in transcriptions. A model monitoring pipeline helps detect this drift and ensure that the model continues to perform well.

The first step in monitoring drift is to determine the metrics used to track its performance. For an ASR system, we can focus on Word Error Rate (WER) which measures how many words the model got wrong compared to the ground truth transcription and Character Error Rate (CER) which Measures errors at the character level such as mispelling or missing and extra characters. If these error rates increase significantly, it indicates that the model is not performing well and might be drifting from its original accuracy.

Every time the ASR model processes an audio file, audio file ID, the transcribed text output, and the ground truth transcription will be logged. This can be done by using a Python script that logs each of these elements. With the transcriptions and ground truths, the WER and CER can be calculated directly in Python.

To track drift over time, another python script can be set up to periodically check the WER and CER for new batches of transcriptions. For instance, every day or week, this script would be ran to calculate the average WER and CER for all transcriptions processed in the last period. If the WER or CER exceeds a predefined threshold (e.g., WER > 0.2), this indicates a significant drop in model accuracy.

This python script can then be set up to notify if the average WER or CER exceeds a certain threshold another python. This can be done by triggering a notification such as an email or a log entry to send an alert the model might be experiencing drift. Cron job on a server or a scheduled task on the local machine with Windows Task Scheduler can be used to automate the monitoring and scheduling scripts to run periodically. The script will continuously monitor and log the WER and CER, checking for any signs of drift.

The main challenge with this would be the need for the Ground truth transcription which can be labour intensive as it would likely require manual labelling to correctly determine the actual transcription of the new input audio files. Hence, Semi-supervised Evaluation is proposed so that instead of comapring every single transcription to human-generated ground truth, only a subset of the transcriptions will be manually verified. Periodically check the model’s performance on small but representative set of data to reduce the need for a complete ground truth for every audio input.