A model monitoring pipeline is essential in ensuring consistent performance of Automatic Speech Recognition (ASR) model. In this pipeline, it consists of multiple stages from data preprocessing where data is ingested and transformed, performance monitoring to detect on any potential model drift, continuous finetuning with the new data, to deployment of updated model to production.

In the data preprocessing stage, there is a need for us to compare new data with the training data, so we take some sample of the new data where the sample size depends on the size of the new data. We will then transform the data as needed and feed this data to the next stage.

In the model monitoring stage, we will look for any sign of model drift present. Model drift can split into 2 types, concept drift and data drift. Concept drift occurs when the relationship between independent and dependent variables changes over time, for instance, change in accent/slang. Data drift occurs when there is a change in input data distribution, for instance, presence of background noise in new data.

Regardless, we could detect both concept and data drift with a method. We could first extract embeddings of same sentence from both training data and new data by using the feature extraction layer of the model. We could then compare both embeddings using cosine similarity. If there is low similarity score, it could indicate potential model drift. Upon detecting potential model drift, the pipeline could either trigger an alert for human to review the model or automatically proceed to the next stage.

In continuous finetuning stage, we will finetune the ASR model with the previously sampled new data. For the mistranscribed data, we might have to involve human-in-the-loop feedback to incorporate human reviewer/user's corrections. Alongside with the new data and updated transcription, we will perform incremental training to finetune and adapt the model without retraining the entire model. We could then test on the new unsampled data with the updated model to evaluate the overall quantitative impact. We could use Word Error Rate, which indicates the percentage of incorrectly predicted words, to measure its accuracy. If the metric is deemed to be similar or better than the previous model, we will proceed to the next stage.

Lastly, in deployment stage, we could implement A/B testing for both updated and the previous model. We split the traffic 50/50 to both models and we will use qualitative metric as the key metric for the A/B testing. For instance, assuming that there is user satisfaction scoring system in place, we will use user satisfaction score to determine if we should fully deploy the updated model after a given time period. In the case If the pipeline needs to be fully automated, we could gradually increase the traffic to the updated model till 100% traffic.

This entire model monitoring pipeline can be run daily/weekly depending on the urgency of the use case or the budget that the team has.