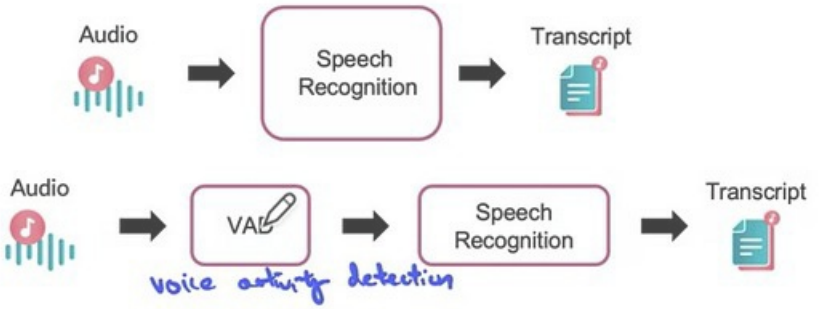
ML Pipeline

Many AI systems are not just a single ML model running a prediction service, but instead involves a pipeline of multiple steps.

What are ML pipelines? How do build monitoring systems for that?

Speech recognition example

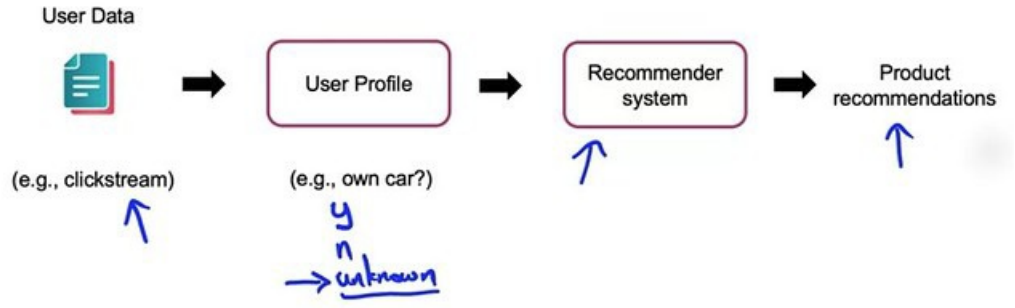
- In this system, we put a VAD (Voice Activity Detection) module before the speech recognition engine in order to reduce the load of server from unnecessary voice. We only want the engine to receive input when the user actually speaks to the mic.
- In this example, the VAD itself might be a learning algorithm.
- When you have two learning modules working together, changes to the first module may affect the performance of the second module.
    - e.g. the new cellphones VAD modules clips the audio differently (from what you trained your speech recognition model on).



Some cellphones might have VAD clip audio differently, leading to degraded performance

User profile example

- Assume you have user data (e.g. clickstream) and this can be used to build a user profile that tries to capture key characteristics of a user (e.g. whether user owns a car? YES/NO/UNKNOWN).
- Now, imagine this “predicted” user profile now is being fed to a recommender system to generate product recommendations.
- If (for some reason) the user data distribution changes and as a result the user profile model (input to the recommender system) now outputs more "UNKNOWN"s, then this may affect the product recommendation results.



Sum up: In ML pipelines, the cascading effects of differet ML components can be complex to keep track on.

It's useful to brainstorm metrics to monitor that can detect changes, including concept drift or data drift (or both) at multiple stages of the pipeline.

Metrics to monitor

Monitor

- Software metrics
- Input metrics
- Output metrics

Note: the principle from last section to brainstorm whatever that could go wrong and use them as metrics still applies here but to multiple components of the pipeline.

How quickly does data change?

- It's very problem-dependent.
  - In face recognition system, the rate at which people's appearances change is not that fast.
  - On the contrary, in a cellphone factory, they receive all new materials to make new phones.
- The range of change (depending on the application) is vast; goes from minutes to months/years.
- On average,
  - User data generally has slower drift.
  - Enterprise data (B2B applications) can shift fast.