Stage 14 – Deployment and Monitoring Reflection

In moving from productization to deployment, the biggest risk is that a model which worked well in development may behave poorly in production. The causes can range from data drift (new inputs look different from training data), schema changes, or higher rates of missing values, to gradual degradation of accuracy as the business environment evolves. Technical issues such as API downtime, slow response times, or unhandled errors also pose risks. The consequence of these failures is not only lower model performance but also business harm if incorrect predictions are acted upon.

To mitigate these risks, clear monitoring is needed at four levels. At the **data level**, I would track freshness of inputs, the percentage of nulls, and schema changes, raising an alert if nulls exceed 5% or if critical features go missing. At the **model level**, I would monitor accuracy, error rates, or other metrics (e.g., MAE, AUC), with alerts triggered if a two-week moving average falls below an agreed threshold. At the **system level**, latency (p95 under 500ms), uptime, and error rates are essential. Finally, at the **business level**, I would monitor KPIs such as approval rates or conversions, since these reflect real impact.

Responsibilities should be clearly divided: data/ML engineers own the dashboards, platform/on-call engineers own system alerts, data scientists handle retraining, and analysts/business owners monitor KPIs. Issues are logged and triaged via Jira, with rollbacks approved by an engineering lead. This layered monitoring ensures stability and trust in production ML.