**Model Monitoring Pipeline and Model Drift Tracking**

In the a machine learning model's lifecycle, monitoring and maintenance is required from the when the model reaches its deployment phase. Once deployed, models may be subjected to changes in the data environment, potentially resulting in model drift. Hence, a model monitoring pipeline is required to detect and address such drift. The following depicts a model monitoring pipeline that can be used. Since this pipeline is not created for any specific ML model, description of each part of the pipeline will be kept slightly generic.

**1. Logging and Data Capture**

The first part of the pipeline involves comprehensive logging. For each prediction generated by the deployed model, the system should log the following: raw input features (after preprocessing), prediction output, model version, timestamp, request latency, and relevant metadata. These logs can be asynchronously streamed to a message queue such as Apache Kafka to decouple logging from inference latency. Subsequently, the data can be stored in both a high-performance analytics database for real-time querying, and a long-term object storage for archive.

**2. Ground Truth Integration**

Next, To measure post-deployment performance, it is important to link predictions with actual outcomes once ground truth labels become available. This can be achieved via batch jobs that join previously logged predictions with newly available labels. The frequency of this process depends on the latency between prediction and label availability.

**3. Metric Computation**

The third part of the pipeline involves measuring and monitoring different metrics, which can be done in the following three ways.

1) Data Quality Monitoring: Evaluates the integrity of incoming features such as null value rate, range violations, feature type mismatches.
2) Measures shifts in feature distribution relative to a baseline training distribution using statistical tools such as the Population Stability Index or Kullback-Leibler divergence.
3) Model Performance Monitoring: Tracks traditional performance metrics such as accuracy, precision, recall, F1-score, and AUC. This layer may also include latency and throughput monitoring.

All metrics should be computed across relevant population segments (e.g., geography, customer type) to enable granular insights.

**4. Visualization and Alerting**

Upon measurement of the above metrics, dashboards are employed to visualize metrics in real-time. Alerting mechanisms will also implemented via Amazon CloudWatch with predefined thresholds.

**Conclusion**

A well-defined model monitoring pipeline is critical for sustaining the performance and reliability of ML systems in production. By integrating data quality checks, statistical drift

analysis, real-time dashboards, alerting mechanisms, and automated retraining workflows, organizations can proactively manage model drift. This ensures that ML models remain robust and accurate amidst evolving data landscapes and usage patterns.