Model Monitoring Pipeline

A model monitoring pipeline is crucial for monitoring and ensuring the model is performing reliably overtime. The pipeline should include checks for common reliability issues such as drift and anomalies.

Firstly, ingest raw data and store them in a data lake or warehouse. Then extract the features and process them to fit our requirements.

Secondly, check for data and concept drift. Monitor changes in input feature distributions using statistical tests and compare results against actual labels using metrics like Population Stability Index.

Thirdly, monitor the model's performance through key metrics such as accuracy, precision, recall or RMSE for regression models.

Next, ensure fairness in target variables using demographic parity or equalized odds and using bias mitigation techniques such as re-weighting if biasness exists.

Then, check for anomalies using Isolation Forests or autoencoders and make logs and send alerts for unsatisfactory performances

Finally, repeat the pipeline with new data learnt using MLOps pipelines such as MLflow to ensure minimal error over time.

Model Drift Tracking

Model drift is the degradation of a machine learning model's performance overtime due to the changes in data or input and output variables. There are a few types of model drift: Concept Drift, Data Drift, Feature Drift, Prediction Drift and Label Drift.

Concept Drift occurs when the patterns learned have become irrelevant. We can focus on the changes in relationships between input features and the target variable, comparing it to a baseline using performance metrics like accuracy, precision or recall.

Data Drift occurs when the input distribution changes, but the relationship between inputs and target variable remains the same. We can focus on the changes in the distribution of input features such as data volume and bias. This can be done manually or using algorithms such as Chi-square to track changes in feature distribution.

Feature Drift occurs when there is a change in statistical properties in the input features resulting in poor model performance. We can monitor the feature statistics manually or use algorithms as mentioned previously to test feature distribution.

Prediction Drift occurs when the model's predictions for new data differ significantly from earlier periods. We can focus on the changes in distribution of model predictions with statistical tests such as accuracy, precision, recall or F1-score and cross-validate with updated datasets to ensure adaptability.

Label drift occurs when the distribution of the target variable changes over time, even if input features remain the same. We can focus on the changes in distribution of the target variable by manually monitoring or using algorithms such as ADaptive WINdowing to detect changes in variable distribution.

The best way is not just to detect it but to know how to respond. Therefore, we need a drift-aware system that can detect changes. Has memory to adapt to changes but forget irrelevant details. Be able to continuously learn and update the predictive model from the evolving dataset. And finally, ability to measure how bad the model's prediction was in each batch to minimize errors in prediction over time.