Model monitoring pipeline and how I would track model drift

Model drift refers to the degradation of model performance. This can negatively impact model performance, resulting in poor predictions and decision making.

Two forms are:

1. Covariate drift, when the distribution of the input data P(X) changes over time
2. Concept drift, when the task that the model was designed to perform P(Y|X) changes over time.

In order to detect shifts and take action, we need to have a model monitoring pipeline.

Metrics

Evaluation metrics relevant to the model such as F1 or recall should be monitored. Ideally this should be compared against ground truth labels.

However, if ground truth labels are unavailable within a reasonable timeframe to be useful, we could instead monitor the changes in input features distribution using statistical methods such as the Kolmogorov-Smirnov (K-S) test. The K-S test is a nonparametric statistical test to determine whether two sets of data come from the same distribution.

These features are however harder to monitor. There could be high-dimensionality of features and many stages of data-processing which makes it difficult to drill down to a single cause of degradation.

The raw inputs could also be monitored, usually by the data engineering team. This includes checking for missing values, duplicates or outliers in the input data.

Performance of the end product/solution, or user feedback are useful in aligning the evaluation metrics used.

Ideally we have a combination of metrics or features to monitor, to draw insights of causes of model degradation, such as at which stage of the pipeline, or if it is due to external factors.

Time scale

We could monitor these statistics over a sliding window of a suitable time scale. For example, if the model or performance is sensitive to hourly changes, we might need to monitor granularity level of hourly or less.

A more granular level of monitoring (e.g. hourly) allows detection of more sensitive changes and faster alert of abnormalities. However, it may show more noise and is more operationally or resource heavy. A hybrid approach is when we have a less granular level of monitoring (e.g. daily) and drill down to granular information for investigation when necessary.

We need to weigh the impact of drop in model performance in order to choose an appropriate time scale.

Tools and practices

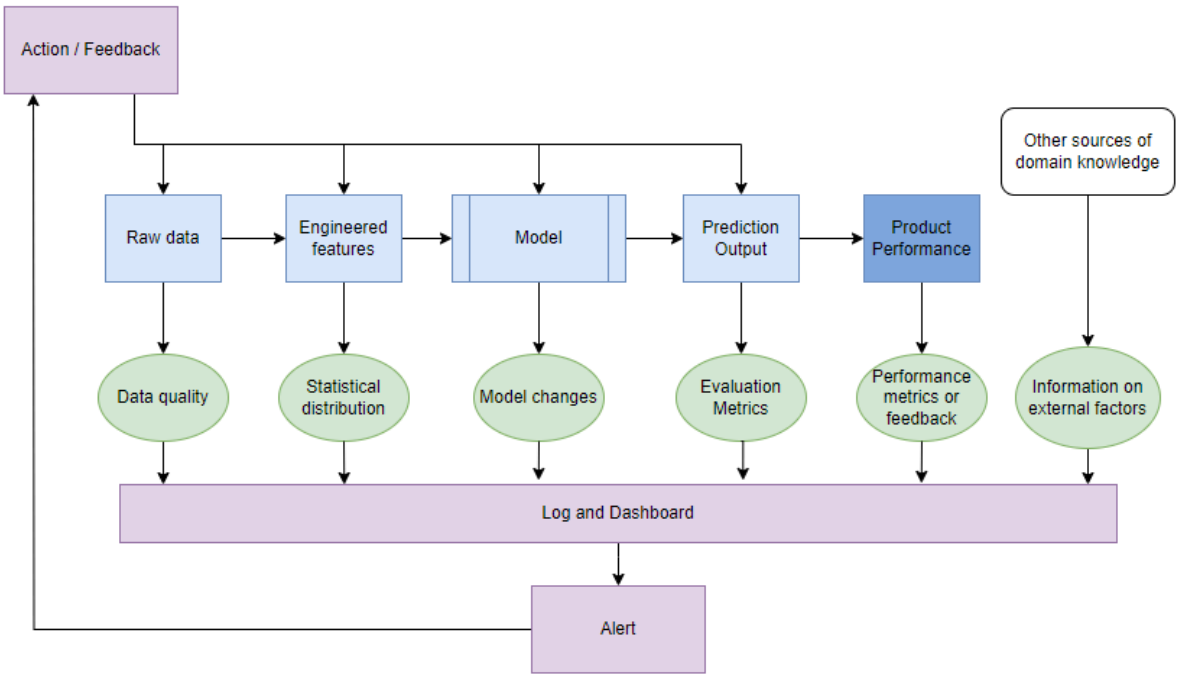
Systematic logging of any changes in features, conditions or parameters with timestamp and traceable ID may be useful for identifying degradation causes.

Dashboards help to visualise metrics, useful for both ML engineers and stakeholders if there are signs of model performance degradation.

Alert notification – with the right threshold condition, message description, and to the appropriate stakeholder allows for quick response.

Tools such as Evidently (an open-source ML observability platform) allows for continuous monitoring of deployment with dashboards.

Below is an example of integration of model monitoring into an end-to-end ML pipeline.



References:

Designing Machine Learning Systems by Chip Huyen
https://www./topics/model-drift
https://www.datacamp.com/tutorial/understanding-data-drift-model-drift