Abstract

In production environment, ML models deal with real time data that are influenced by external variables, not always identifiable during model development and training. In the worst case, they change over time bringing the model predictions to be no longer useful. Data drift breaks the fundamental assumption of machine learning models: data distribution is static and past data is representative of future data. Therefore, ML algorithms face with a constant stream of new, changing data, and the need to be regularly updated in order to maintain their predictive ability.

For the time being, most ML prototype and artifacts struggle with production wall. In most of cases, the goal is to deploy a trained model as a prediction service to production. The monitoring module is absent, and performances are manually checked implying no control on model degradation, as far as the search for result interpretability.

A further stage of maturity is reached when the monitoring module is present and used to observe how performances change over time. Although, until the module is used with a *reactive* approach, meaning handled to exploit the decrease of performances and then act, its potential remains unexplored. In fact, a *proactive* monitoring is more advantageous: an AI algorithm is implemented and used inside the module to predict in advance the decay of the model. In this way, preventive actions may be taken to sustain the model and not have a negative impact on business. In fact, in advanced frameworks, the monitoring module is a vital element of the ML pipeline that consent to proactively detect model performance decay through which preventive actions may be taken to sustain the model and not have a negative impact on business.

This work proposes the implementation of an AI-based monitoring system that proactively identify and analyze performances decay of AI models. The monitoring module is context agnostic and receives as input the model delivered by development environment, both business and performance metrics related to the model to be controlled, and production data. The framework is composed by three parallel streams that are evaluated as new data arrive. The first stream carries out a deterministic analysis and evaluates the drift of model performances from the preceding timestep to the current.

The second stream is characterized by a proactive approach, through predictive analytics on time series data, to forecast a drift in performances and take corrective measures at current time. A State of Art Deep Learning model is trained on a dataset whose features evaluate the statistical distance among batches and the target variables are the metrics related to monitor. The predictive model learns to associate shift in input distributions with those related to the target, and lagging the target consent the forecast.

Each time a new batch of production data arrives a new record is added to the dataset, the predictive model is updated and a new prediction for the next step is available. If the difference between the current metrics and the predicted on overcome predefined thresholds the monitoring module triggers the retraining of the model.

The last stream applies AI Explainability via Causal Inference to detect and generate reports that describe how features influenced and determined shift in performances. Reports are automatically generated and give a comprehensive view on the drift causes.