ML Model Monitoring and Re-training in Statistical Organisations

ML 2022 Model Re-training Theme Group[1]

1. Introduction

Background

A machine learning (ML) model is built based on a set of data called "training data". The premise and foundational assumption for machine learning applications in the real world is that the patterns the model has learned from such training data are valid in making a prediction as new data come in. However, ML models, even those build with the highest quality data and carefully engineered, start to decay over time.

The model decay (also known as model drift) can occur for various reasons. First, the underlying phenomenon that is captured in the current data change over time. For example, as new types of jobs and products enter the market, new words can appear in the job descriptions of labour force surveys or in the product descriptions in web-scraped data. Hence the model may potentially be exposed to patterns it was not trained on. The changes can be gradual and isolated but also abrupt and extensive, as happened with market crashes and pandemics (e.g., word "covid" suddenly appears in abundance as the cause of illness or death). Systematic and structural change in the data can be introduced without any change in the underlying phenomenon, for example, through the change of data collection method. The model performance can decrease due to a cumulative effect of errors creeping into the prediction pipeline (e.g., undetected breakdown of a data transformation function for a specific feature variable).

Context for the Statistical Organisations

With growing demands for trusted information, rapidly developing and accessible technologies, and numerous competitors, the statistical organisations have been actively exploring the potential of the ML in its statistical production process. The promising areas range from automation of manual tasks, production of new

[1] This theme group report was written based on discussions and contributions from InKyung Choi (UNECE), Andrea del Monaco (Bank of Italy), Eleanor Law, Shaun Davies, Joni Karanka, Alison Baily (UK ONS), Riitta Piela, Toni Turpeinen (Statistics Finland), Ayoub Mharzi (IMF), Soheil Rastan (UN ESCWA), Kimberley Flak (Statistics Canada) and Susie Jentoft (Statistics Norway). The views and opinions expressed in this report do not reflect the official view of the organisations mentioned.

statistical products based on big data and replacement of status-quo analysis methods with ML [1].

The degradation of the ML model (i.e., a substantial decrease of the model performance) inevitably affects the quality of the statistical outputs that are based on the model predictions. Therefore, the model should be continuously monitored and re-trained as needed to maintain its performance. In this sense, the monitoring and re-training should be considered as a part of the quality management process needed for the application of ML.

ML is a relatively new area of work in the statistical organisation and there is still a lack of experience with the ML models in production. This often leads to the low awareness and understanding of the importance of model monitoring and re-training, making it even harder for the model to be accepted in production and to be trusted by stakeholders, creating a vicious cycle (Figure 1).

As a public organisation and producer of official statistics, the adoption of new technology such as ML could take a long time in the statistical organisations. Many statistical organisations are trying to integrate the ML in the regular production process and structure. While the performance of the ML model (e.g., accuracy) might be a necessary condition for it to be deployed, it is by no means a sufficient condition. Without a proper plan to monitor and re-train the model in production, the model would not be accepted by end-users nor supported by IT.

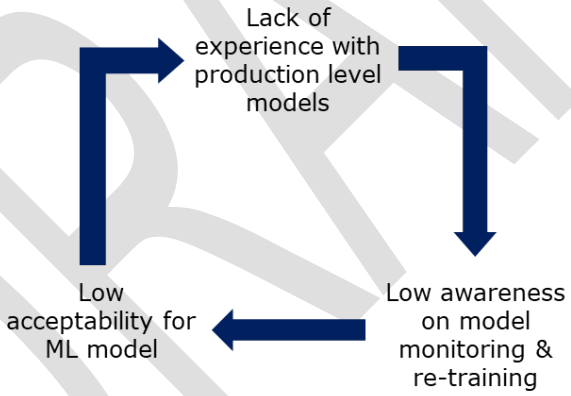

Figure 1. Vicious cycle around the ML model monitoring and re-training

The probability distribution of data is often non-stationary in real world applications, and this creates an ongoing maintenance cost that is often hidden [2] or even overlooked. As with any quality assurance system, the monitoring of the model requires careful planning with engagement of multiple stakeholders. Procedure for monitoring and retraining as well as division of work among stakeholders (e.g., data scientists, IT experts; see more in Section 5) should be established before the model is deployed to production.

Purpose of the Paper

This paper aims to introduce core concepts in the domain of model monitoring and re-training and the methods developed to address them and discuss

practical issues and challenges that might arise. The focus of the paper is strictly around the work of statistical organisations and the context under which they operate, with a goal of increasing the awareness about the importance of model monitoring and re-training so that ML can be integrated in the regular work of statistical organisations in a sustainable manner.

Implementing drift monitoring and model re-training in practice generates various issues. First, monitoring each type of drift incurs costs, hence statistical organisations have to understand which drifts are more relevant for the application area and even feasible to monitor. Also, a data generating process is usually stochastic with noise and it is natural to observe variation in the data and model performance to a certain extent. Therefore, the extent of change that is qualified as "drift" should be determined. Lastly, drift does not necessarily mean the model should be re-trained. The cost of re-training might be too expensive or estimates from the drifted model can be corrected without re-training [14]. Hence, cost and benefits of different options should be evaluated before deciding the model re-training. Figure 2 shows the workflow and decision points around the drift monitoring and model re-training.

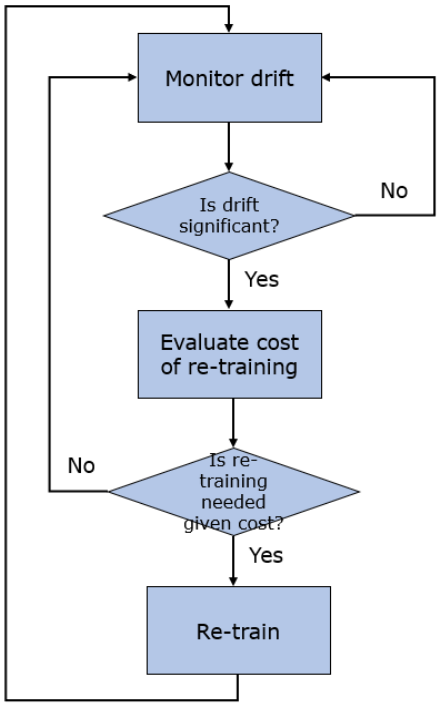

Figure 2. Decision flows around drift monitoring and model re-training

The rest of the paper is organised as follows: Section 2 introduces key concepts; Section 3 and Section 4 discusses monitoring and re-training in more depth with practical considerations from the perspectives of the statistical organisations. The overarching issues that cut across the workflow that should be addressed at a corporate level are examined in Section 5. The paper concludes in Section 6 with concluding remarks and suggestions for future work.

2. Key Concepts

ML algorithms can be broadly categorised into supervised learning, unsupervised learning and reinforcement learning [16, 17]. The environment in which the training dataset is provided can be grouped into offline learning (all learning instances are presented simultaneously) and online learning (instances are presented one at a time) [4]. Much work on ML model monitoring and re-training focuses on the online setting applications where there is continuous data stream such as customer click and GPS data [5, 6]. However, statistical organisations are not usually a provider of a real time individual service based on a constant data stream. This is beginning to change, but currently most statistical products usually have a fixed periodicity (e.g., quarterly, monthly) and the sample data arrives or/and is processed in batches. Also, most ML application areas in the statistical organisations currently focus on supervised learning algorithms due to, for example, the difficulty of asserting any quality statements for the predicted results for unsupervised learning. For these reasons, this paper focuses on the supervised learning and offline setting as many ML examples in the official statistics field fall under this combination.

Drift refers to a change in an entity with respect to a baseline. The "entity" can be the ML model itself, in which case the drift generally refers to a situation where the predictive performance of a model changes. The "entity" can also be the dataset on which the model was trained.

To formalise the key concepts, let us denote X, a set of features and Y, a target variable with a joint distribution $P(X,Y)$. For the supervised learning in a batch environment, we have a data set $S = \{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_n, y_n)\}$ of covariates and targets with an unknown mapping $Y = f(X)$ binding the former to the latter. A prediction model is then obtained as $\hat{f}_S$. At the time of model development, the training set have both feature and target variable, but when the model is in production, we normally only have features and predicted target values:

$$S^{new} = \{(\boldsymbol{x}_1^{new}, \hat{y}_1^{new}), \dots, (\boldsymbol{x}_m^{new}, \hat{y}_m^{new})\} = \{\left(\boldsymbol{x}_1^{new}, \hat{f}_S(\boldsymbol{x}_1^{new})\right), \dots, \left(\boldsymbol{x}_m^{new}, \hat{f}_S(\boldsymbol{x}_m^{new})\right)\}$$

In a typical machine learning problem setting, the aim is to predict the value of the target variable given the values of the covariates. For example, in a classification problem where the target variable Y takes one of the values $c_1, \dots, c_k$, the model $\hat{f}_S$ assigns the category that has the highest conditional probability $\hat{y}_j = \hat{f}_S(\boldsymbol{x}_j) = argmax_c \hat{P}_S(y = c|\boldsymbol{x}_j)$. Hence, the degradation of the model means that the estimated conditional probability learned from data S, $\hat{P}_S\left(y = c|\boldsymbol{x}_j\right)$, does not hold anymore.

The "concept drift" is a change in the joint distribution (i.e., $P_S(\boldsymbol{X},Y) \neq P_{S_{new}}(\boldsymbol{X},Y)$) and "model drift" refers to a change of predictive performance of model $\hat{f}_S$ learned from training data set S (i.e., $\hat{P}_S(Y|\boldsymbol{X})$ is not valid for making prediction for new data whose conditional distribution is $P_{S_{new}}(Y|\boldsymbol{X})$). Concept drift makes the pattern that the model learned from training data S outdated. In this sense, concept drift can be considered as a cause and the model drift as an effect.

Other important terms in the field are feature drift, target drift and posterior drift. Using the Bayes theorem, the change of the conditional probability can be decomposed as below:

$$P(Y, \boldsymbol{X}) = P(Y|\boldsymbol{X})P(\boldsymbol{X})$$

Therefore, the concept drift can be attributed to feature drift which refers to a change of distribution of feature (also called "covariate drift"), target drift which refers to a change in the distribution of target variable Y and posterior drift refers to a change in the posterior distribution.

In the literature, there are many terms describing drifts but with a lack of standard terminology established [3, 9, 23]. We will not go into details about the review or comparison of different terms (for the extensive review of taxonomy, see references aforementioned), but instead focus on four types of drift concepts with terms aforementioned. Table 1 summarises these concepts with examples under the scenario that an ML model is used for classifying job descriptions from survey data.

Drift types		Definition	Example
Model drift		Change of the overall performance of model	Model was deployed 2 years ago with accuracy 95% and its accuracy has dropped to 75%
Concept drift	Posterior drift	Change in the conditional distribution $P(Y\|\boldsymbol{X})$	Word "PC" used to indicate "Police Constable", so the job description that contained the word was likely to indicate the person was police, but these days "PC" more frequently used for "Personal Computers".
	Feature drift	Change of distribution of feature $P(\boldsymbol{X})$	Word "data science" starts appearing more frequently in recent survey data
	Target drift	Change of distribution of target variable $P(Y)$	There are more computer related jobs in recent survey data

Table 1. Types of drifts

It is important to note that the occurrence of any of concept drift, feature drift, target drift or posterior drift does not necessarily lead to the model drift. For example, if the distribution of feature X change in a way that model is still valid (right side diagrams in Figure 3), the model performance would not decrease. Also, if concept drift happens only for the very rare class and affect the model performance only very marginally, the change of overall model performance would look limited while there are significant changes in performance for that particular class.

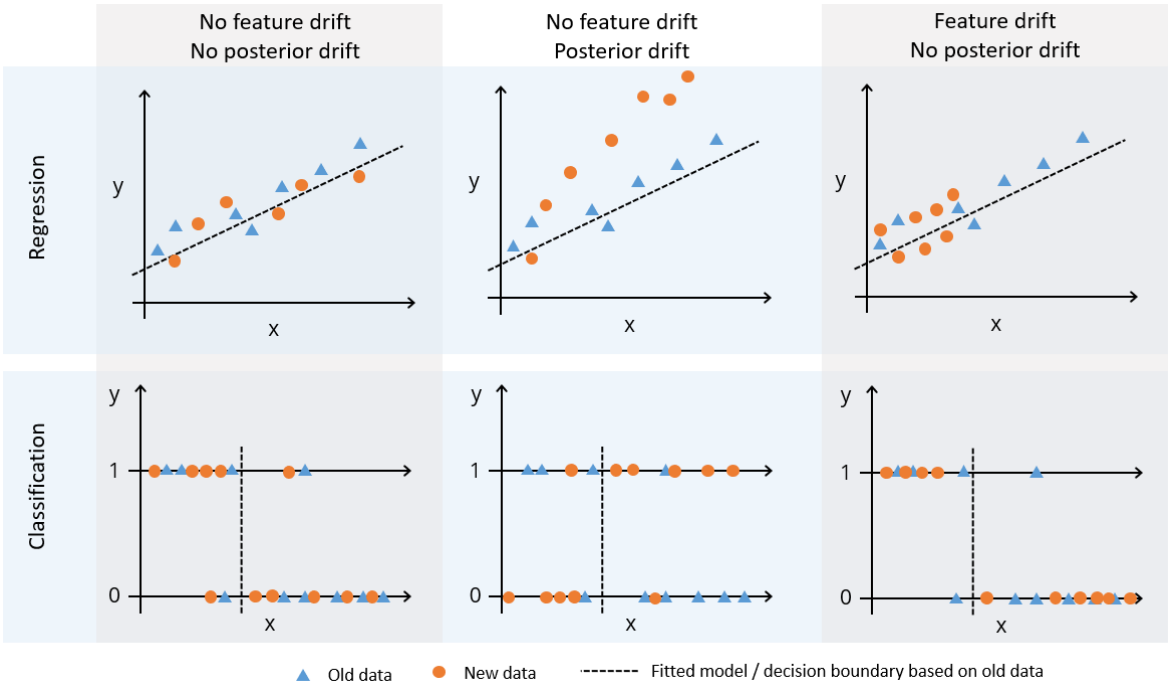

Figure 3. Drifts in regression and classification setting

The drift can be related to a concept familiar in statistical organisations: representativity. It can be argued that there is actually no change in the underlying phenomenon, but rather, the data that was used to develop the ML model was simply not a representative sample (e.g., there was sampling bias in the data and a certain demographic category was not included in the data) [3]. Indeed, it might be hard to differentiate (or quantify the contribution of) a sampling bias from a true change in the population in practice without additional knowledge or research controlled over some period of time to rigorously identify the real cause of change. Whether the change was due to sampling bias or drift, however, model should still be monitored and re-trained if the change is unacceptable. Discussion on the representativity and quality of training data is beyond the scope of this report and in the rest of the report, it is assumed that the change was due to drift, rather than spurious error due to the poor quality of data (for more discussion on this topic, see [26]).

3. Drift Monitoring and Detection

In [9], the monitoring and detection approaches are grouped into 4 types: data distribution-based, performance-based, multiple hypothesis-based and contextual-based. In this paper, we focus on two more prominent approaches (data distribution-based and performance-based), relating them to different types of drifts introduced in the previous section and discuss practical issues.

3.1. Performance-based Approach

The principal idea behind the performance-based approach is that if there is a drift and changes are introduced in the new data that had not been captured by the model then the model would not perform as well as it did. Therefore, this approach monitors performance metrics (e.g., accuracy) and requires a collection of true values of target variables $\{y_i^{new}\}_{i=1..m}$ to be able to determine the error of the predicted values $\{\hat{y}_i^{new}\}_{i=1...m}$. There have been many detection methods developed to address different types of scenarios with different strengths. In this report, three basic methods are explained (for the extensive review of performance-based methods, see [9]).

Drift Detection Method (DDM)

DDM [18] is one of early works on the drift detection and many subsequent methods were developed using DDM as a basis. DDM is often discussed in the context of streaming scenario where sample arrives one at a time, but the method is equally applicable for the batch scenario.

Consider a set of records $\{(\mathrm{x_i}, y_i)\}_{i=1,..t}$. At each point, the error (i.e., $\mathrm{y_i} \neq \hat{y}_i$) follows a Bernoulli distribution. DDM monitors error rate p_t which follows a binomial distribution with its standard deviation $\mathrm{s_t} = \sqrt{p_t(1-p_t)/t}$. DDM alarms for and detects drifts when

$$p_t + s_t \geq p_{min} + 2 * s_{min}$$

$$p_t + s_t \geq p_{min} + 3 * s_{min}$$

respectively, where p_{min} and s_{min} are updated values when a new record is processed and $p_i + s_i < p_{min} + s_{min}$. DDM is available via python library such as scikit-multiflow.

There are extensions and modifications of DDM. The Early Drift Detection Method (EDDM) [21] aims to address the difficulties of DDM in detecting gradual changes by introducing Hoeffding Drift Deviations Method (HDDM) uses Hoeffding's inequality to detect drifts [25].

Statistical Test of Equal Proportions Detection (STEPD)

STEPD [19] is based on the well-known statistical hypothesis testing of two sample proportions. It compares the error rate at the time of the model

deployment p_0 and that of the current period p_t. Denote n_0 and n_t as the size of samples at the deployment time and given time point t; and p^* as the combined error rate $((n_0 p_0 + n_t p_t)/(n_0 + n_t)$. The test statistic with a continuation correction:

$$\frac{(p_0 - p_t) - 0.5\left(\frac{1}{n_0} + \frac{1}{n_t}\right)}{\sqrt{p^*(1-p^*)\left(\frac{1}{n_0} + \frac{1}{n_t}\right)}}$$

then follows the standard normal distribution. This statistic is monitored, and warning is activated when the p-value of the above statistics is below significance level set.

Variations of STEPD extends this method, for example, using Fisher's test instead of two sample proportions test [24]. Two-sample proportion test in STEPD is available via python library such as scipy.

Adaptive Windowing (ADWIN)

ADWIN [22] utilises the sliding window W with an adaptive size. It detects drifts, if there are two large enough sub-windows, W_0, W_1, with sample sizes n_0, n_1 respectively, that have averages that are distinct enough:

$$|p_0 - p_1| \geq \epsilon_{cut} = \sqrt{\frac{1}{2m} \ln \frac{4(n_0 + n_1)}{\delta}}$$

where m is a harmonic mean of n_0 and n_1 and δ is a user defined global permissible error. If drift is detected, the window size shrinks by removing records at the tail. A less conservative cut point and a more memory efficient method (ADWIN2) can be found in [22]. ADWIN is available via python library such as scikit-multiflow.

Error Intersection Approach (EIA)

Most work on drift consists of classification tasks where the target is categorical variables as remarked in [9, 23]. The Error Interaction Approach (EIA) [21] is developed for regression tasks. It was inspired by the paired learner method and utilises two models: a simple but reactive model that is quick to adapt to short-term changes and a complex model that is stable but good for general structure. EIA compares the performance (e.g., root mean squared error) and switch between the models; thus, drift is detected when the error curves "intersect".

The performance-based approach inevitably creates non-negligible cost as human coders or experts are needed to prepare a new ground truth data. In the absence of ground-truth values, the monitoring of predicted values can be used as proxy ($\{\hat{y}_i^{new}\}_{i=1\ldots m}$). In the case of classification, the ML model often gives out a measure of uncertainty (probability of predicted category). The decrease in

these probabilities can signal a potential change as it means the model is becoming less sure about its prediction.

In some organisations, the ML model is used as a supporting tool for humans [10, 11]. For example, when the ML is used for coding and classification, the model suggests the top 5 most likely codes and a human makes the final decision, so that ML facilitates the manual decision making as opposed to completely automating the decision process. In such an implementation scenario, the (formal or informal) feedback provides a valuable indication of a change in performance quality. Human verification is naturally a part of pipeline and the cost of collecting the ground-truth values might not be as high as starting over.

The significance level that triggers the model drift (e.g., p-value in STEPD) should be defined in consultation with the business owners and end-users. Care should be taken when using the overall performance metrics as an indicator for drifts because the performance degradation for an individual class (when the target variable is categorical) or segment (when the target variable is continuous) may be not visible in the overall metric, in particular, when it is for a rare class (or a small segment).

3.2. Data Distribution-based Approach

Data distribution-based approaches compare the distribution of training data used for the development of the model with that of the new data. As it is often difficult and expensive to obtain the true values of the target variable $\{y_1^{new}, y_2^{new}, \dots, y_m^{new}\}$, this is often done for input features (i.e., $\{x_1, x_2, \dots, x_n\}$ vs. $\{x_1^{new}, x_2^{new}, \dots, x_m^{new}\}$).

For the traditional data that statistical organisations have usually worked on, many features (input variables) take numeric values (e.g., age, income) or a limited number of categories (e.g., education level, gender). For these data types, the distribution of features can be formulated with contingency tables, histograms, empirical cumulative distribution and so on. Monitoring of the change of the distribution can be done through measures of distances between two distributions, for example, Kullback-Leibler Divergence or Hellinger Distance and testing such as Pearson's chi-squared test or Kolmogorov-Smirnov test where applicable. One difficulty with a data distribution-based approach is that when the feature space is high dimensional, the number of feature combinations quickly grows. For multi-dimensional space, one can use Mahalanobis distance or Hotelling's *T* distance (for a review of various distance measures, see [15]).

While the theory of comparing distributions may be conceptually straightforward, the distribution of features can be often difficult to formulate in practice, in particular for new types of data. For the non-traditional unstructured data, it is not straightforward to establish even the feature space. Textual data, which is one of the most popular ML application areas in the statistical organisations, is an example of such data. In the text analysis, the raw data are texts written in a natural language (e.g., "I am a cook working at a restaurant" for job description).

For imagery data, which is another popular ML application area, the issue is even more complicated because the feature space may not be selected manually by the model developer, but by the model itself. The deep learning algorithms that are often used for imagery data analysis may extract and engineer features on their own.

To apply ML algorithms, they are prepared and transformed into a vector space [1]. For example, if one uses the Term Frequency-Inverse Document Frequency (TF-IDF) method, the vector space is created from the word tokens. If a word embedding method is used, the vector space is determined by the embedding used. Therefore, compared to the case of traditional categorical or numeric data, the feature space is not set clearly. Therefore, one should first establish the quantitative feature space

3.3. Comparison of the Two Approaches

The performance-based approach measures directly the degradation of the model which is the ultimate purpose of the monitoring. The data distribution-based approach can be trickier to establish a corporate-wide monitoring system as the method depends heavily on data type, data preparation methods and so on. As mentioned in Section 2, it can lead to false alarm (i.e., feature drifts happens but model drifts do not happen) [9].

However, monitoring feature drift is still important even when the model performs well for several reasons. First, monitoring the input data is the first line of defence and provides a warning flag before the model performance changes noticeably. There might be changes in progress without manifesting as a decrease of prediction performance yet, in particular, when the performance measure is aggregated across different output labels. Secondly, monitoring feature drift is important to ensure user trust. The end-users of the models are often those who have a close connection with data (e.g., subject matter experts). When there are noticeable changes, or even error, in the input data but the models are performing just as well, it raises questions about the model, solidifying the conception that ML model is a “black box”, which negatively affects the buy-in of the users. Lastly, monitoring of data distribution can help inform the decision as to which re-training approaches to take (see more in Section 4).

4. Model Re-training

Re-training costs

Once the model drift is detected, one should decide whether to retrain the model or not. While in the ideal situation, it would be always recommended to retrain, in practice, there are various financial non-financial cost factors that need to be considered.

- **The impact of the model drift** can vary depending on the application areas and use cases of the ML model. For example, if the model is used for recommending most likely codes for human experts during intermediate processing, the impact might be more limited to a small increase of inefficiency in the production process. On the other hand, if the model is used directly for the final statistics (e.g., sentiment indicator, crop yield estimates), the drop of model performance affects the credibility of the statistics and hence that of the statistical organisations. As the credibility is non-negotiable value for the official statistics, in this case, the model needs to be retrained. Ideally, some kind of impact assessment would be carried out, by simulation or otherwise, to understand the sensitivity of produced statistics to errors in an ML model used as part of the statistical production process.
- The resources needed for retraining should also be taken into consideration. Depending on the severity of the drift, and in the case that it cannot be automated, the retraining might require an **involvement of data science and ML experts**. As these skills are still rather limited in some statistical organisations, the experts would need to be pulled away from other projects they are working on to retrain the model. The retraining of the model might require involvement of staff in other areas as well, for example, an approval from the management or a clearance for deployment from IT. **Obtaining a new data set for re-training** could incur significant costs for the statistical organisations as it often requires human experts to label the data. In some cases, labels can be available before a need for re-training arising. For example, if human experts make the final decision based on the predictions made by the model, or if there are quality control processes where a set of model predictions is reviewed by human experts, the labels would be available.
- Other potential costs include a risk of introducing changes in time series (e.g., when the model is used for the final statistics that are regularly released) and the disruption for production processes (e.g., when the model is used for intermediate processing).

Re-training approaches

There are several approaches for model re-training. If there are only non-significant concept drifts (i.e., changes in the distribution of input data or label) but a significant model drift (i.e., changes in the prediction error), this may

indicate that the pattern learned from the data in the previous round do not hold anymore. The simplest approach would be to fix everything in the pipeline except model parameters (e.g., pre-processing processes, hyperparameters), and re-fit the model again with a new data set to obtain new model parameters. If the model performance does not increase after this, it may indicate that there might be issues with input data (e.g., the input features are no longer relevant for the prediction task). In such cases, feature engineering needs to be done again or new features should be collected. While the minimum goal of the re-training is to meet the quality requirements originally defined by the user, it is possible that the nature of the task may change so that other metrics become more relevant, for example, increasing class imbalance may cause accuracy to be a poor measure. Ideally, retraining would maintain or improve the level of performance at the time of deployment. If performance cannot be recovered, one could also consider rebuilding the model altogether, e.g., by testing different ML algorithms.

It is important to note that the re-training strategy should be discussed and established based on these cost factors and re-training options before deploying the model, not at the time when re-training is needed. Also, some costs can be avoided and reduced when planned in advance, for example, by establishing contingency plans in case of any issues with the models.

5. Cross-cutting Issues

At the early stage of ML maturity, the development of a model is often given the attention and research is focused on experimenting different ML models to improve accuracy. However, just like any new capability[2] in statistical organisations, an ML solution should be transferred to production and managed as a corporate support (i.e., cross-cutting activities required by the organisation to deliver its work programme efficiently and effectively[3]). Statistical capabilities have several dimensions: institutional setting, people, process, technology, information, standards, and method[4]. While all of these dimensions are crucial to work together to integrate ML into statistical organisation, we focus on three aspects that are particularly relevant to the model monitoring and re-training.

Process - division of work

The development of the ML solutions requires a multidisciplinary collaboration. The main expertise involved in the process can be grouped into four categories:

- Data science expertise developing the machine learning model
- Subject-matter expertise providing deep understanding of data and guiding the development the model
- Statistical expertise designing the quality assurances (e.g., for training data, for validation of models)
- IT experts providing a platform to host the model and developing necessary services for users to monitor models

It is important to note that the level of involvement of each expertise changes over the stage of the development. For example, in the PoC development phase, the data scientists may be most involved in the training and testing the ML models based on the user needs. Information on infrastructure and any IT constraints should be taken into account. In the transition phase, the involvement of IT experts increases to put the solution in the production line. Statistical methodology expertise (e.g., sampling) is needed to develop quality control procedures. The set of monitoring metrics and courses of action (e.g., for a significant sustained drift, consider re-modelling, otherwise, take automatic retraining) should be set and clearly documented in collaboration between all those involved. In the production phase when, ideally, a hand-over of ML solution is completed, the role of data scientists becomes minimal, and it is

[2] Capability is defined as "an ability that an organization, person, or system possesses. Capabilities are typically expressed in general and high-level terms and typically require a combination of organization, people, processes, and technology to achieve" (TOGAF 9.1). Examples of capabilities in the context of works of statistical organisations include seasonal adjustments, data visualisation, risk management (https://statswiki.unece.org/display/DA/CSDA+2.0+-+X.+Capabilities).
[3] Generic Statistical Activity Model (GAMSO; https://statswiki.unece.org/display/GAMSO)
[4] https://unece.org/fileadmin/DAM/stats/documents/ece/ces/2018/CES_10_E_Statistical_capacity_development_strategy.pdf

mainly the business owner and IT experts who are involved with using, monitoring and maintaining the solution in production.

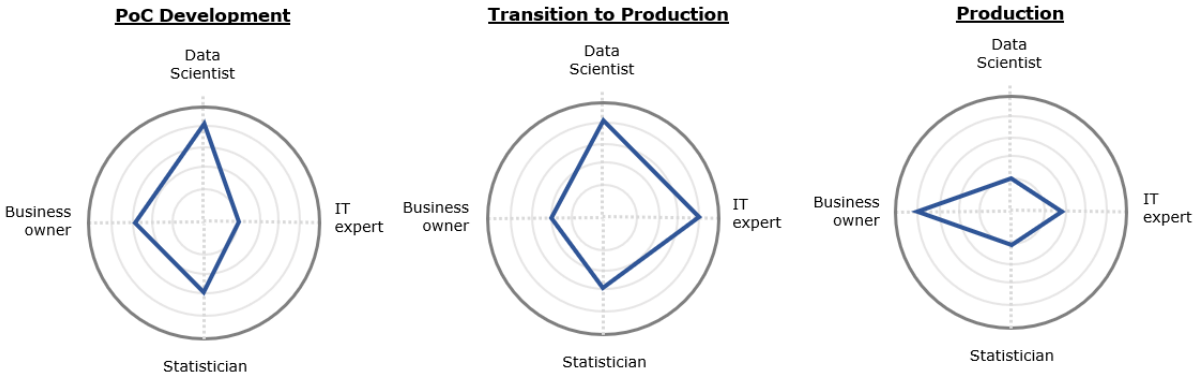

Figure 4. Change of roles across different stages

From the organisational point of view, this change of involvement should be clearly understood, specified and agreed among all actors throughout the beginning to the end. With a growing number of machine learning projects, there will be increasing demand for the data science expertise to develop new models. Without proper division of work or plan for corporate support, data scientists would end up maintaining existing models as well as developing new models. With more and more experience accumulated, it is also important to standardise the process by capturing milestone points, core information needs at each point, and common templates for documentation.

Information – metadata

Metadata (data about data) has historically played an important role in the statistical organisation as its main business is the production of statistical data. Metadata contain information on quality and structure of data, methodology used for the analysis or administrative details that help users to understand the data.

With a growing need for digital information, metadata has now come to mean any descriptive information about some objects of interest [13]. Considering the ML model as the object of interest, any information that is needed to understand the model can be seen as its metadata. Metadata for a model foremost should contain details on the model itself (e.g., model version, developer(s), performance metrics at the time of deployment, hyperparameters used), but also include information on the data used (e.g., data lineage, data owner) and the pipeline that generated the model (e.g., pre-processing) as both input data and processes that generated the model could have a great influence on the model.

Just like with statistical data, the metadata could make the models more transparent and re-usable. Having a minimal set of metadata and proper documentation of this information is particularly critical for model monitoring and retraining as they take place after the initial model development (sometimes

after few months or years) and are conducted by those who were not involved in the model development. Models without proper documentation are not reproducible, hence model metadata is an important element for the responsible use of ML. During the transition period, the set of metadata needed for the selected monitoring and re-training methods (see Section 3-4) also needs to be set up. For example, if a data distribution-based approach is used, one would need the distribution metrics corresponding to features and combinations of features as agreed to monitor. ML lifecycle management platform services that provide a registry that captures information on models and relevant artifacts automatically could streamline the collection and management of the metadata.

6. Conclusion

ML models are developed based on data, and once deployed, they start decaying over time as the underlying phenomenon changes. With statistical organisations expanding the scope of ML applications and trying to move the ML solutions into production, it is important to take into consideration the monitoring and re-training as parts of the governance and maintenance plan for the ML model. Continuous monitoring and re-training are also an important principle of ML Operations (MLOps)[5] which aims at facilitating the productionisation of ML models. With the process steps related to ML being automated, the manual intervention is minimized and retraining can also be triggered and performed automatically when the MLOps automation level is sufficiently high.

In the official statistics community, there are more and more organisations investigating this topic[6]. The further works on how to implement the monitoring and re-training system and connect them to other components in a broad ML infrastructure and environment would be needed (for more discussion on ML environment, see [27]).

[5] MLOps defined as "a paradigm, including aspects like best practices, sets of concepts, as well as a development culture when it comes to the end-to-end conceptualization, implementation, monitoring, deployment, and scalability of machine learning products" (https://arxiv.org/abs/2205.02302)

[6] See, for example, model monitoring experiments from UK Office of National Statistics: https://github.com/ONSBigData/drift_detection_model_retraining/blob/main/Concept%20Drift.ipynb

References

[1] UNECE (2021) "Machine Learning for Official Statistics"

[2] D. Sculley, et. al. (2015) "Hidden Technical Debt in ML System", in Advances in Neural Information Processing Systems 28, NIPS

[3] J. G. Morento-Torres, et. al. (2012) "A unifying view on dataset shift in classification", Pattern Recognition, vol. 45, pp. 521-530, 2012

[4] Pat Langley (1996) "Elements of Machine Learning"

[5] Paulo Mauricio Gonçalves Jr., et. al. (2014) "Comparative Study on Concept Drift Detectors" Expert Systems with Applications 41(18):8144-8156; DOI:10.1016/j.eswa.2014.07.019

[6] Sonia Jaramillo-Valbuena, et. al. (2017) "Performance evaluation of concept drift detection techniques in the presence of noise"

[7] UNECE Statistics Wiki (accessed in August 2022), Machine Learning for Official Statistics Studies and Codes

[8] Beck, M., F. Dumpert, and J. Feuerhake (2018) "Machine learning in official statistics" arXiv:1812.10422. DOI: https://doi.org/10.48550/arXiv.1812.10422

[9] F. Bayram, et. al. (2022) "From concept drift to model degradation: An overview on performance-aware drift detectors" Knowledge-Based Systems; DOI: https://doi.org/10.1016/j.knosys.2022.108632

[10] Standard Industrial Code Classification by Using Machine Learning (2020), from the UNECE HLG-MOS Machine Learning Project (accessed December 2022): https://statswiki.unece.org/display/ML/Studies+and+Codes

[11] Automated Coding using the IMF's Catalog of Time Series (2020), from the UNECE HLG-MOS Machine Learning Project (accessed December 2022): https://statswiki.unece.org/display/ML/Studies+and+Codes

[12] UNECE (2009) "Common Metadata Framework – Statistical Metadata in a Corporate Context: A Guide for Managers"

[13] National Academies of Sciences, Engineering, and Medicine (2022) "Transparency in Statistical Information for the National Center for Science and Engineering Statistics and All Federal Statistical Agencies" The National Academies Press. https://doi.org/10.17226/26360

[14] Q. A. Meertens, et. al. (2022) "Improving the Output Quality of Official Statistics Based on Machine Learning Algorithms", Journal of Official Statistics, Vol. 38, No. 2, 2022, pp. 485–508, http://dx.doi.org/10.2478/JOS-2022-0023

[15] Igor Goldenberg and Geoffrey I. Webb (2018) "Survey of distance measures for quantifying concept drift and shift in numeric data", Knowledge and Information Systems 1-25, https://doi.org/10.1007/s10115-018-1257-z

[16] I. Goodfellow, Y. Bengio, and A. Courville (2016) "Deep learning", MIT Press, p. xxii+775

[17] Murphy, Kevin P. (2012) "Machine Learning: A Probabilistic Perspective", MIT Press

[18] J. Gama, P. Medas, G. Castillo, P. Rodrigues (2004) "Learning with drift detection", in Advances in Artificial Intelligence - SBIA 2004, pp. 286–295.

[19] K. Nishida, K. Yamauchi (2007) "Detecting concept drift using statistical testing", in: International Conference on Discovery Science, Springer, pp. 264–269.

[20] Lucas Baier, Marcel Hofmann, Niklas Kühl, Marisa Mohr and Gerhard Satzger (2020) "Handling Concept Drifts in Regression Problems – the Error Intersection Approach, https://doi.org/10.48550/arXiv.2004.00438

[21] Baena-Garcia, M., del Campo-Avila, J., Fidalgo, R., Bifet, A., Gavalda, R., Morales-Bueno, R. (2006) "Early drift detection method", In: Proc. ECML/PKDD 2006, Knowledge Discovery from Data Streams, pp. 77 - 86

[22] A. Bifet, R. Gavalda (2007) "Learning from time-changing data with adaptive windowing", in: Proceedings of the 2007 SIAM International Conference on Data Mining, SIAM, pp. 443–448.

[23] Geoffrey I. Webb, Roy Hyde, Hong Cao, Hai Long Nguyen, Francois Petitjean (2016) "Characterizing Concept Drift", https://doi.org/10.48550/arXiv.1511.03816

[24] D.R. de Lima Cabral, R.S.M. de Barros (2018) "Concept drift detection based on Fisher's Exact test", Inform. Sci. 442, 220–234.

[25] I. Frias-Blanco, J. del Campo-Ávila, G. Ramos-Jimenez, R. Morales-Bueno, A. Ortiz-Diaz, Y. Caballero-Mota (2014) "Online and non-parametric drift detection methods based on Hoeffding's bounds", IEEE Trans. Knowl. Data Eng. 27 (3) 810–823.

[26] The final report from the ML 2022 Theme Group on the Quality of Training Data, available on UNECE wiki page (accessed December 2022) https://statswiki.unece.org/display/ML/Machine+Learning+Group+2022

[27] The final report from the ML 2022 Theme Group on Infrastructure, available on UNECE wiki page (accessed December 2022) https://statswiki.unece.org/display/ML/Machine+Learning+Group+2022