Model Degradation

Nishant jairath

7th AUGUST 2022

# Introduction

A machine learning model, trained and tested on historical data, is said to be degraded when it performs poorly with new dataset. The decay in model performance is usually attributed to temporal changes in data and label-feature relationship.

Maintaining model’s output above the benchmark requires continuous tracking of various aspects of the model and taking corrective actions. Therefore, model deployment cannot be considered the last milestone in machine learning lifecycle, but rather a beginning to the next phase called model monitoring as shown in Figure 1 below.

A picture containing table

Description automatically generated

The purpose of this document is to understand the reasons which lead to model degradation, metrics to detect it and its resolution.

## Reasons behind Model Degradation

There are two main root causes behind the decay in prediction accuracy – Data and Model. Below, we further elaborate upon these causes for better understanding:

1. **Data**: Machine learning models are dependent on input data and any changes in the data leads to corresponding changes in model output. Data changes can be attributed to following categories:
   1. *Data Drift*– with time or location, the feature variables can get displaced from their central tendency or dispersion. This can lead to introduction of new variety of data points within the existing feature variables or new labels which did not exist earlier.
   2. *Data Types* – each feature has a specific datatype which can get transformed as part of the data collection or transformation process. A model can then inadvertently begin using the new data type itself which can alter the output.
   3. *Data Quality* – missing values (NaN), duplicates, special characters, outliers, among others hamper the data quality which can impact the model if they are not treated before ingestion.
   4. *Feature Importance* – a model usually has many features with a varied degree of relevance, which are dropped to keep only the ones with significant importance. With time, feature’s importance can undergo a shift, and the model can miss out on including a revised set of features which lead to output degradation.
2. **Model**: Machine learning models can itself lose relevance with time as the predictions made may not remain accurate. Such a scenario can happen due to three main reasons:
   1. *Concept Drift*- the interpretation of data changes even when the distribution of data remains the same. For example – symptoms of a disease can go undergo variation with time and therefore a model trained on historical symptoms may not be able to accurately predict the disease based on new data.

Depending upon the business, a concept drift can occur in different ways such as sudden drift, gradual drift, incremental drift, recurring drift and blips drift. This ultimately defines the cadence of monitoring the deployed model.

* 1. *Hyperparameter Tuning* – model is finalized after fine tuning its hyperparameters which provide the best accuracy of predictions and with time lapse these hyperparameters need to be retrained to capture the essence of latest data. It may also become necessary to change the model altogether if the accuracy doesn’t improve with retuning.
  2. *Exogenous Factors* – Covid is an apt example which impacted machine learning algorithms unexpectedly. Similarly, competitor moves such as advertisements, campaigns, acquisition can impact prediction accuracy as such factors are not part of the model development exercise.

## Metrics to detect Model Degradation

For the above root causes, we can define lead and lag indicators which can help us detect model degradation. These metrics can be plotted, using visualization tools, on an ongoing basis so that team can track them and take necessary action in case of any abnormalities. Below are four factors related to input data which can be tracked to detect degradation:

1. *Data Drift* – variation in the new data can be tracked using metrics such as measures of central tendency (mean, median, mode), measures of dispersion (variance, standard deviation, range, and interquartile range).
2. *Data Types* – running python-based scripts to extract the data type of each column and comparing it with historical data set can bring out the deviations. Consistency in data type is essential for the model to perform predictions accurately.
3. *Data Quality* – script-based checks to create logs of column-level - unique values, number of nulls, duplicates, special characters, among others can help in detecting any significant variation in data quality. Moreover, automated emails can be sent to the team if the variations are beyond a threshold for requisite manual intervention.
4. *Feature Importance* – correlation matrix needs to be built for every new dataset to check if any of the features are colluding. Principal component analysis (PCA) can be used to identify the features which capture the maximum variation in dataset. Further, mutual information (MI) can be used to bring out the importance associated with each feature. Outputs of these three techniques should remain consistent for every dataset and in case of any deviation we will have to retrain the complete model.

All the above metrics and techniques are lead indicators which can help us detect a possible negative impact on output beforehand and therefore can be mitigated proactively. Below we look at three factors pertaining to model operations which can impact its prediction:

1. *Concept Drift* – this is more subtle to observe and therefore we need to execute the model and then determine metrics such as accuracy, Precision/Recall, ROC-AUC, F1 Score in case of classification and mean absolute error, root mean squared error, r2 and adjusted r2 for regression models. All these metrics can help us detect a spurious change in model’s behavior and to rectify it at the earliest.
2. *Hyperparameter Tunning* – same metrics which are mentioned in previous point can be used to understand if we need to retrain the model and find new hyperparameters which yield accurate predictions.
3. *Exogenous factors* – in the wake of any exogenous scenario, a model should be run to track the before and after shift in the output accuracy. Again, the same metrics as mentioned for concept drift will be helpful in identifying the degradation and business team can then ratify if impact is due to the exogenous factors itself.

The metrics shared in the above section are lag indicators which can raise a red flag when the model accuracy drops below a certain threshold. A summary about the model degradation is shown below in Figure 2.

A picture containing timeline

Description automatically generated

## Resolution

Corrective actions for above root causes can be grouped in the following way:

1. **Data cleaning** – any discrepancy discovered during data preparation should be dealt before feeding the data into model. Periodic checks (Weekly/Bi-weekly) should be carried out on data source and log files should be created during the data transformation steps to ensure there are no last-minute surprises.
2. **Weighted data ingestion** – there are temporal changes which impact the data. Model can be fed with weighted data where higher weight is given to recent data over historical so that the underlying shifts can be captured adequately.
3. **Re-labelling** – classification model can be used to provide probability (predict\_proba class) for each data point w.r.t different labels. When the probability of multiple data points lies below a benchmark level for all the labels then we can introduce a new label which categorize all these data points. This is one of the ways of tackling concept shift as well.
4. **Retrain/Rebuild model** – largely for data shift and concept shift, it is necessary to retrain the model on the complete dataset. As part of retraining, new hyperparameters are also looked at which can better predict the real time outputs.

It is a good practice, to develop multiple models and compare their individual accuracies to take corresponding weighted average of outputs from all of them.

Moreover, new models keep emerging on ML landscape and they can be tested for better prediction power. Replacing the old models with latest can help mitigate the emerging temporal shifts.

1. **Manual checks** – to capture impact of exogenous factors, manual checks need to be done for ratifying the cause of deviation in model accuracy. For example - sales prediction of vehicles before covid was 95% accurate but after covid the model accuracy slumped to 75%. In such a case we need to add a new feature to capture this drop which can only be determined through manual intervention. Similarly, business team can guide about adding a feature variable to flag competitor’s promotional activities and account its impact in model’s prediction.
2. **Online model training –** when there is a continuous stream of input data available, models can be trained on an ongoing basis to capture any drifts in decision boundary. This can be achieved through partial\_fit function which helps to train the model using new data on top of the learnings from previous data. This results in continuous adjustment of model parameters and keep the accuracy within the guardrails. When the model’s performance doesn’t meet the benchmark then it is rebuilt using all the data.

## Conclusion

A model’s usefulness is dependent upon its prediction accuracy. With continuous shift in data and model interpretation there is a need to update the model on an ongoing basis. Model retraining and redevelopment can become a costly activity if done very often and therefore, a structured monitoring approach should be followed. Through the lead and lag metrics monitoring, we can decide when to retrain or rebuild a model. It can help us to keep both the budget and the model degradation in check.

Resources:

<https://towardsdatascience.com/why-machine-learning-models-degrade-in-production-d0f2108e9214>

<https://ml-ops.org/content/mlops-principles#monitoring>

<https://medium.com/codait/keeping-your-machine-learning-models-up-to-date-f1ead546591b>

<https://analyticsindiamag.com/concept-drift-vs-data-drift-in-machine-learning/>

<https://www.nature.com/articles/s41598-022-15245-z>

<https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning#mlops_level_1_ml_pipeline_automation>

<http://xplordat.com/2019/04/25/concept-drift-and-model-decay-in-machine-learning/>

<https://neptune.ai/blog/concept-drift-best-practices>

<https://www.kdnuggets.com/2022/01/machine-learning-models-die-silence.html>

<https://paulvanderlaken.com/2020/03/24/ml-model-performance-degradation-production-concept-drift/>

<http://xplordat.com/2019/05/06/fracking-features-in-machine-learning/>

<https://www.elastic.co/blog/beware-steep-decline-understanding-model-degradation-machine-learning-models>

<https://neptune.ai/blog/how-to-monitor-your-models-in-production-guide>