On monitoring of machine learning models

Dmitry Namiot, Evgeniy Ilyushin, Ivan Chizov
Lomonosov Moscow State University

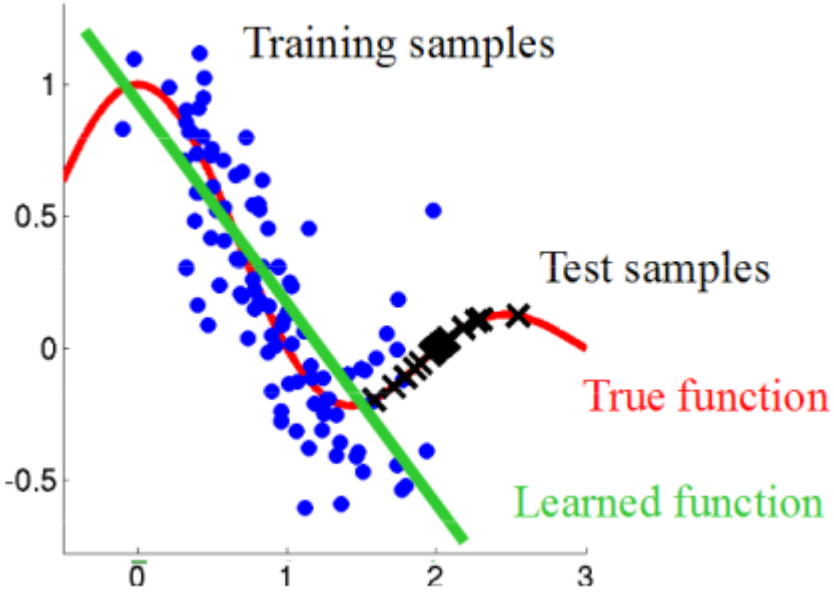
dnamiot@gmail.com

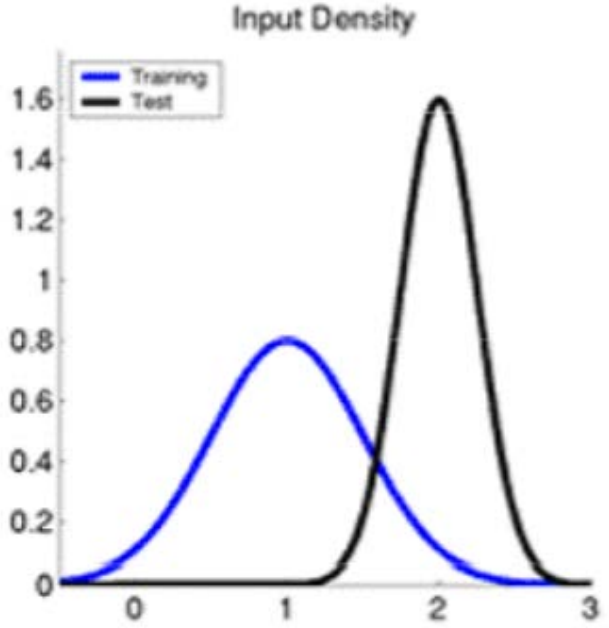
DCCN 2022

Outline

- The machine learning model has completed the training phase and is put into production
- How to make sure the system works correctly?
- What should be checked while the system is running?
- This article is devoted to the issues of monitoring machine learning systems.

Covariate shift





Why do you need to monitor your models?

- Data distribution changes

  Why are there sudden changes in the values of my features?

- Training-serving skew

  Why is the model giving poor results in production despite our rigorous testing and validation attempts during development?

Why do you need to monitor your models?

- Model/concept drift

  Why was my model performing well in production and suddenly the performance dipped over time?

- Black box models

  How can I interpret and explain my model's predictions ?

Why do you need to monitor your models?

- Concerted adversaries

  How can I ensure the security of my model? Is my model being attacked?

- Model readiness

  How will I compare results from a newer version(s) of my model against the in-production version(s)?

Why do you need to monitor your models?

- Pipeline health issues

  Why does my training pipeline fail when executed? Why does a retraining job take so long to run?

- Underperforming system

  Why is the latency of my predictive service very high? Why am I getting vastly varying latencies for my different models?

Why do you need to monitor your models?

- Cases of extreme events (outliers)

  How will I be able to track the effect and performance of my model in extreme and unplanned situations?

- Data quality issues

  How can I ensure the production data is being processed in the same way as the training data was?

Data shift

- X is feature (covariate) space, Y is label space
- P(X): distribution of features
- P(Y): distribution of labels
- P(X | Y): distribution of features given specific labels
- P(Y | X): distribution of labels given specific features
  - *This is what ML models are trying to learn!*

Data shift

- Covariate shift
  - $P(Y \mid X)$ is the same but $P(X)$ changes
- Label shift
  - $P(Y)$ changes but $P(X \mid Y)$ is the same
- Concept shift
  - $P(Y \mid X)$ changes but $P(X)$ is the same

Measure drift of Independent Features

1. Monitor Distribution of each feature
2. Monitor the Statistical Features
3. Monitor the distribution of multivariate features

Monitor Distribution of each feature

If we observe a change in the distribution of engineered or raw features of the inference data, we can expect a decline in model performance. Some of the popular statistical techniques are:

- KL (Kullback Leibler) Divergence Test
- KS (Kolmogorov Smirnov) Test
- Chi-square Test

Monitor the Statistical Features

- One needs to monitor the statistical features of the inference and baseline data, to observe the divergence in the dataset. Some of the statistical features are:
  - Range of possible values (quantiles, mean, max, min)
  - Number of missing or NULL values
  - Histogram distribution of numerical features
  - Distinct Values of Categorical features

Monitor the distribution of multivariate features

- Machine learning models develop some interactions between the features to make predictions.
- If the pattern or distribution between the features is changed then it may lead to a decrease in model performance.
- The technique to detect the multivariate feature distribution is:

  Cramer’s Phi Test

Measure drift of Dependent Features

- The dependent feature (target label) for the inference target class maybe not be present upfront in production.
- Once the dependent feature is present, there are various techniques to measure the drift and come to a conclusion of whether the model performance has deteriorated or not.

Distribution of Target Class

- For the classification task, the target class label is categorical in nature. The idea is to compare the distribution of target class labels between the inference data and base data.
- For regression tasks, the histogram plot, or statistical feature of the continuous target label can be used to measure the drift in the data.

Monitor Inference Model Performance

- Once the actual target class label is made available, then the model drift can be detected by evaluating and comparing the performance of the model on standard metrics.
- If the model metrics show less than expected numbers, the model needs to be re-trained.
- What does it mean ‘retraining’ for critical software (avionics, etc.)?

Open problems

- Data streams and streaming solutions
- How to deal with 24/7 models in critical areas?
- If you want to compare the distribution of the input data with the original distribution, then where is this original distribution stored?
- If we have an embedded system, then this information needs to be stored directly on the embedded platform? There it can become available to strangers.

Alibi detect

- Alibi Detect is an open source Python library focused on outlier, adversarial and drift detection.
- The package aims to cover both online and offline detectors for tabular data, text, images and time series.
- Both TensorFlow and PyTorch backends are supported for drift detection
- https://github.com/SeldonIO/alibi-detect