The concept of Drift in Analytics and Machine Learning and how to control it in QA?

Avik Maity

Abstract— There is little doubt that Big Data's emergence has ushered in tremendous potential and opportunity for businesses to make better-informed decisions and gain real time insights. It was almost evident that such disruptive technology will also introduce technical risks – the primary among which is commonly called Drift. Drift issues are no more limited to just the Data Analytics landscape. A second type of "Drifting" was identified when Machine Learning came into the picture for predictive modelling after being integrated with Analytics systems for data. The industry was worried that this "drift" related risk, was not getting covered in traditional QA practices, nor was there a tool which can outright remove the "drift" issues from data. This was primarily because these issues are not due to any piece of code, or deployment or part of any feature change – which can be "tested" via test cases.

In this white paper, we will discuss on the two types of Drifting – Data Drift and Concept Drift, and will try to explore practices which the QA teams can also adopt to reduce the risk.

Index Terms— Big Data, Analytics, Data Drift, QA, Machine Learning, Data Model, Data Warehouse

1 DATA DRIFT W.R.T ANALYTICS

What is Data Drift?

The one aspect which will always be synonymous with Big Data is – Diversity. The sheer variety of data and the volume causes unpredictable and unending "mutation" of data characteristics – i.e., Data attributes can get corrupted. This is a natural phenomenon, not caused by any particular code/database defect or environment. This eventuality of data is what we refer to as "Data Drift". Mobile interactions, sensor logs, and web clickstreams are examples of what can cause Data Drift as these records are constantly subjected to tweaks from Business or updates on platform systems.

Types of Data Drift.

The industry has classified Data Drift into 3 types. Below is a short description of each of them and an example.

Structural Drift: This is the most commonly occurring Drift and takes place when the source data schema is changed. The possible causes are constant adding, deletion, modifying fields in the table over a considerable time period.

Semantic Drift: This occurs when the meaning and representation of the data changes, rather than an actual change on its structure. For e.g. – in a particular quarter, there may be a surge noticed in sales numbers, however it can be a false-positive. The actual reason for the spike can be a change introduced in product, and it has been falsely represented.

Infrastructure Drift: This effect occurs when unwanted changes occur in the underlying software or systems. For e.g. – Infrastructure drift may occur when multiple source tables in the Big Data are having separate governance rules and doesn't take into consideration the synchronization.

Impact of Data Drift:

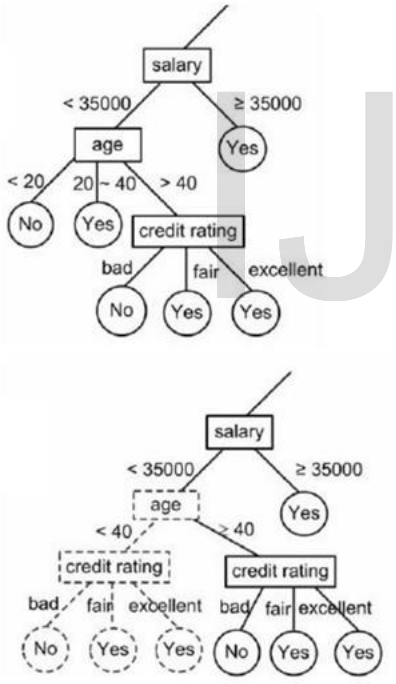
- Damages reliability and productivity of downstream data analysis
- Data corrosion leads to Data fidelity issues (Data cleanliness)
- Chain reaction when corroded data is saved (drifts) undetected into data stores
- False analysis and loss to business

- Overhead for Data engineers who need to clean the mess

POSSIBLE QA ROLE TO CHECK DATA DRIFT

Creating Prototype model: In collaboration with architects and business analysts QA team can create a prototype of the data model. The prototype can define the threshold conditions of data and QA team can recreate those conditions to check for “overfitting” decision tree – after regular intervals (e.g.: at the end of major releases)

Identify decision steps where model faces multiple outcomes: While planning a decision tree, decision points where potentially there can be multiple results, can be identified. At these points, it may be possible for the model to learn something that holds true in general or only discover patterns that hold within one certain dataset. It is at these points where tidy data is critical.



The figure illustrates how a decision step with multiple outcomes can get corrupted due to drifting.

Evaluate objective and categorization: QA can review the data rules with system analysts to check if too many categories are assigned to data which introduces unnecessary variables. This can be part of QA’s Static testing plan.

Cross validate data to optimize solution: When data is pruned from unnecessary categories/attributes, QA can create a post tree procedure to check and match if the validation is returning the same results as the original prototype. If the mismatches are more, then pruning should be stopped and prototype needs to be redesigned.

Using software: There are freeware available in the market (e.g., StreamSets) which are specifically built to clean Drift Data issues. These software typically can read data from local file systems, configure data drift rules, convert data types, load the data into distributed file system (Hadoop) and also provision for alerts, whenever data is corrupted.

A step-by-step sample usage with StreamSets, can be found in the below URL:

https://dzone.com/articles/data-quality-checks-with-streamsets-using-drift-ru

CONCEPT DRIFT w.r.t MACHINE LEARNING

What is Concept Drift?

As per Wikipedia - In predictive analytics and machine learning, the concept drift means that the statistical properties of the target variable, which the model is trying to predict, change over time in unforeseen ways. This causes problems because the predictions become less accurate as time passes.

Data can change over time. This results in poor and degrading predictive performance in predictive models that assume a static relationship between input and output variables.

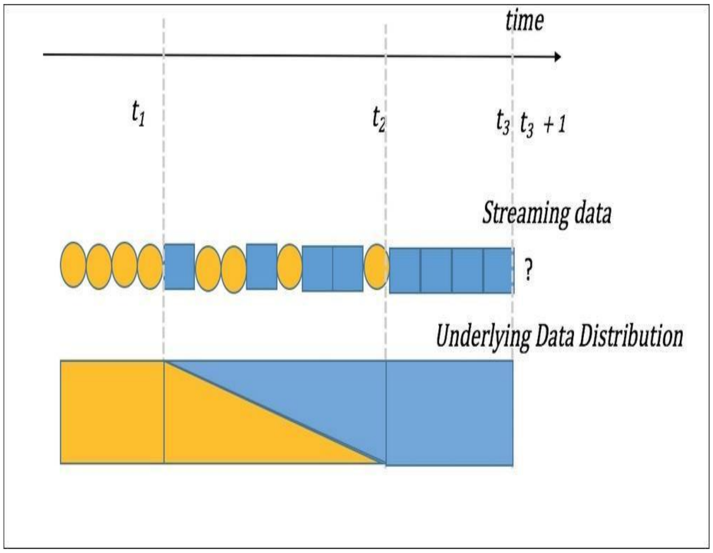
Predictive modelling is essentially a simple functional equation:

$Y = f(x)$

, where Y is the predictive value, f = mapping function and x = input data.

In certain cases, relationship between input and output data changes, meaning, that there are changes in underlying mapping function. The predictions made by a model trained on historical data will no longer be as correct, as a prediction made by a model trained on recent data. If a mechanism can be devised to identify and detect these changes, then it is possible to update the “trained” model to reflect correct changes.

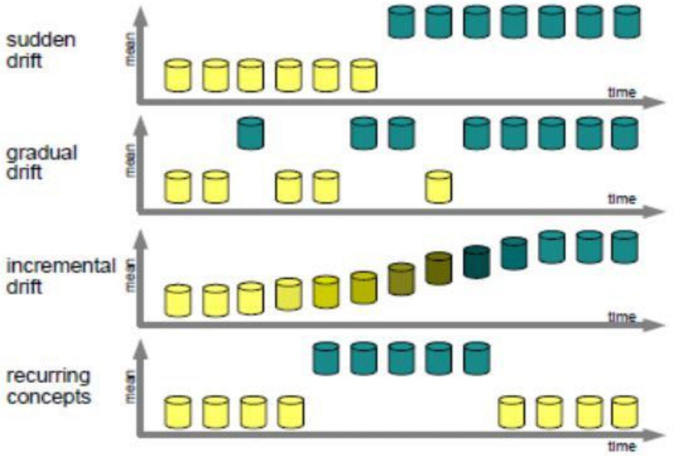
For example, a retail system may be predicting the purchasing behavior of a customer, which may have gone recent change due to strength of the economy, however – strength of the economy may have not been explicitly present in data, which means the prediction may be wrong.



Concept drift illustrated by the gradual change in color from yellow to blue in the bottom panel. Sampled data reflects underlying change in data distribution, which must be detected, and a new model learned.

Indre Zliobaite in the paper titled “Learning under Concept Drift: An Overview” provides a framework for thinking about concept drift and the decisions required by the machine learning practitioner, as follows:

- **Future assumption:** a designer needs to make an assumption about the future data source.
- **Change type**: a designer needs to identify possible change patterns.
- **Learner adaptivity**: based on the change type and the future assumption, a designer chooses the mechanisms which make the learner adaptive.
- **Model selection**: a designer needs a criterion to choose a particular parametrization of the selected learner at every time step (e.g., the weights for ensemble members, the window size for variable window method). This framework may help in thinking about the decision points available to you when addressing concept drift on your own predictive modeling problems



The figure illustrates the typical types of concept drift known so far.

WHAT QA CAN DO TO CONTROL CONCEPT DRIFT?

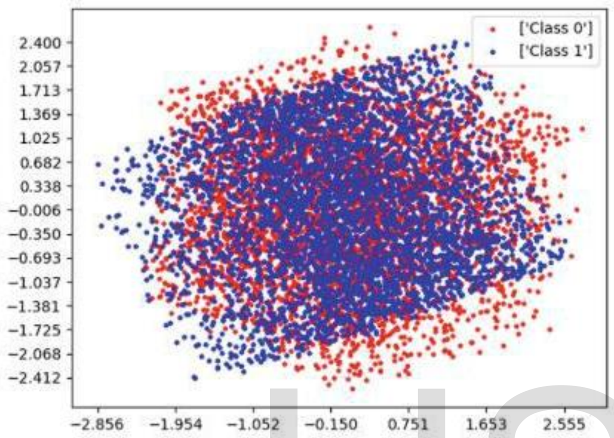
Static Model: This helps in creating a baseline and starting point for comparison with parallel methods. A single model needs to be developed, assuming that concept drift has not occurred (Static). The skill of this model needs to be monitored and when a drop is seen, intervention will have to be taken. This model will require a strong collaboration between architects and QA.

Periodically Re-fit: The static model created, needs to be updated regularly with data collected from a prior period. This will involve back-testing the model in order to select suitable historical data. A good strategy is to do sampling of data and test in smaller portions

Periodically update: Some machine learning models can be updated. This is an efficiency over the previous approach (periodically re-fit) where instead of discarding the static model completely, the existing state is used as the starting point for a fit process that updates the model fit using a sample of the most recent historical data.

Learn the Change: this is a boosting type approach, where the original static model remains untouched, but a parallel new model learns the correct predictions based on relationships on latest data.

Hyperplane: Hyperplane is an idea to detect concept drift by categorizing data classes as 0 and 1 and choosing threshold values as 0.7, 0.3, and 0.2. By analyzing the threshold breaches, the most vulnerable drifts can be identified. Below is a 2D example of Hyperplane.



The following type of observations are derived from a typical Hyperplane:

Average Accuracy	Threshold		
	0.7	0.3	0.2
With Drift Detection	90.25%	90.71%	100%
Without Drift Detection	90.25%	90.54%	90.55%

CONCLUSION

A modern day warehouse must have the following:

- Always up and running
- Ad-hoc SQL Correct answers on any schema
- Terabytes to petabytes of data
- Mixed real time inserts, ETL, batch and interactive workloads
- Thousands of concurrent users

It is evident, that with above provisions the warehouse will be subjected to undetected errors and issues. In this piece, we have tried to identify the biggest problem of them all – Drifting. The industry doesn't see a pure QA role in Drift control, at the moment, but the points we discussed above can be food for thought to build new solution. A very detailed and mathematical solution can be further researched in the dossier – Big Data Analytics, 6th Annual Conference by N.S. Punn and S. Agarwal. They have also proposed a future solution for problems we might face after controlling drift. It will make for some interesting reading.

REFERENCES

- *Big Data Analytics, 6th Annual Conference Dossier – N.S. Punn and S. Agarwal*
- *Understanding Machine learning Algorithms – Jason Brownlee*
- *Wikipedia.com*