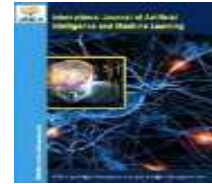




International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: <https://www.svedbergopen.com/>



Research Paper

Open Access

Governed Self-Service Data Quality Gates for Cloud Migration: Distribution- and Model-Aware Cutover Controls

Naga Ravindranatha Reddy Mallipeddi

Independent Researcher

Abstract

Self-service data quality tools allow the lay user to run their enterprise data quality checks as part of the cloud migration, but usually as just one fixed rule, run against a fixed baseline. I argue that the self-service gate should only be trusted for a cutover decision if it detects data distribution shift, and also takes the importance of the changed feature to the model downstream into account. Supported by practitioner experience with migration validation tooling and the drift detection, feature attribution, and self-service governance literatures, I describe a model of what the gate needs beyond accessibility (to be distribution-aware and model-aware) and who is responsible for a cutover decision that the non-specialist-run gate has supported (the team, or the organization).

Keywords: self-service data quality, data drift, model sensitivity, cloud migration, cutover governance, feature importance, data governance

Introduction

The self-service quality gate as it is usually built

Data quality has generally been seen as a specialist task since understanding the quality of data requires context most consumers of the data do not have (Redman, 1998). I have watched cloud migration programs adopt self-service data quality tooling for a straightforward reason anyway: specialist data quality staff are a bottleneck, and the people closest to what a dataset is supposed to mean are usually the business users who consume it, not the engineers who move it. Dashboards and low-code interfaces allow a business analyst rather than a data engineer to define a check against the migrated dataset to be executed before final signature. Low-code monitoring interfaces are already available for exactly this type of purpose for ML pipelines, allowing users with non data engineering backgrounds to directly define and interpret quality checks (Kourouklidis et al., 2020).

What self-service tooling has generally not changed, in every migration programme I have seen it introduced into, is what the check itself actually tests. Most self-service gates run a fixed rule against a fixed baseline: a value must fall within a stated range, a null rate must stay below a stated ceiling, a row count must match within a stated tolerance. The rule is simple to implement and easy to explain, but it ignores the fact that because the overall distribution may have changed so much, the old threshold may no longer mean what it used to mean to an expert on the topic. The other is whether the particular field that tripped or passed the rule is one that the downstream system that consumes the migrated data actually cares about. A gate can be completely accessible to a non-specialist and still be inadequate to the decision it is being used to inform and the two properties get confused for each other more often than I think is generally appreciated.

Scope and contribution

I ultimately make three contributions. To my knowledge, the article distinguishes, for the first time, between accessibility and adequacy in the context of a self-service data-quality gate in the self-service tooling literature, which has mainly focused on the former property. The article proposes a two-axis model of monitoring adequacy beyond a flat rule check: a distribution-drift check, wired from existing concept-drift-detection methods, and a model-sensitivity weighting, wired from existing feature-attribution methods, so that a drift can be viewed through two lenses: is it real, and is it a drift that would affect something that the downstream model actually cares about? It proposes a new

accountability model for this specific circumstance: cutover decision informed by a check a non-specialist configured and ran, which is a genuinely different governance problem than cutover decision a specialist signed off on personally.

From a literature review, the self-service BI and analytics tooling has been studied in terms of technical and governance matters as an organizational problem for a while now (Clarke et al., 2016; Lennerholt & van Laere, 2019, 2020), and so we know that self-service tooling changes who is responsible for a check without necessarily changing what the check catches: this is a known gap in such literature, which we treat as our starting point rather than the final lesson. Finally, a large body of concept drift research covers exactly the distribution-shift problem that the flat rule check cannot: a general taxonomy of types of drift, and algorithms for detecting such drift with known statistical properties (Bifet & Gavalda, 2007; Gama et al., 2014; Moreira dos Reis et al., 2016; Yurdakul & Naranjo, 2020). Third, feature importance or model sensitivity methods identify which inputs impact a model's prediction. These methods, which were originally developed to create interpretable models, could be adapted to predict which data changes warrant escalation (Altmann et al., 2010; Karlaš et al., 2024; Lundberg & Lee, 2017). Because none of these three literatures is focused on cloud migration cutover decisions, migration-specific data quality and risk-assessment research has not considered drift-awareness or sensitivity-awareness in its data validation frameworks (Islam et al. 2017; Jamshidi et al. 2013; Manjunath and Hegadi 2013). We connect the two literatures.

Materials and Methods

Defining a governed self-service gate

By a governed self-service gate (gsg) I mean a flat rule check with two additional checks. The first is a distribution-drift check, that is a statistical test on the incoming batch, to see if the distribution has changed sufficiently from the distribution the rule was trained on, in such a way that the old threshold can no longer be trusted to mean what it meant when it was set. The second is a model-sensitivity weighting, a score derived from the downstream model's own feature importance scores that indicates how much the downstream model would care if that particular field changed. These are used as an input alongside the flat rule, to provide a 3-part signal about what would happen if the field is changed: does it pass, is the underlying distribution different, and does the field matter to what's going to happen next.

Concrete examples may help. If a self-service gate indicates that a customer-address field may not have more than five percent null values, and the current month's data migration has only four percent null values, then that rule will pass this month. A distribution-aware test would see that the four percent figure is far outside the range for which the five percent cutoffs were chosen, because the upstream source system started truncating incomplete records instead of sending all of them with nulls. This quietly changes how the "four percent null" bit is interpreted. A model-sensitivity-aware test would ask whether the churn-prediction model cared a lot or a little about the address field. If the model actually cares much more about tenure and usage patterns than address completeness, a null-rate wobble in that field is a low-priority signal even if it is a real one. However, neither addition will affect whether the original rule passed. Both influence how much you can trust the pass.

Method

This article is based on my own experience of building migration validation and reconciliation tooling, and on a comparative synthesis of empirical research on self-service analytics governance, concept drift detection and feature attribution methods. The method is comparative: a flat rule check, a drift-aware check, a model-aware check, and a fully governed check combining both are all conducted side by side in the same class of migration scenario to highlight where the blind spots of each configuration actually manifest in practice, rather than being assumed *a priori*. This general observation about the accumulation of risk over time from unmonitored data assumptions is not specific to migration, and production ML systems ultimately have to deal with the same risks. For example, a validation stage wired directly into the training and deployment pipelines is a common architectural pattern for production ML systems (Baylor et al., 2017; Sculley et al., 2015). A governed self-service gate is effectively the migration-specific analog of that architectural instinct.

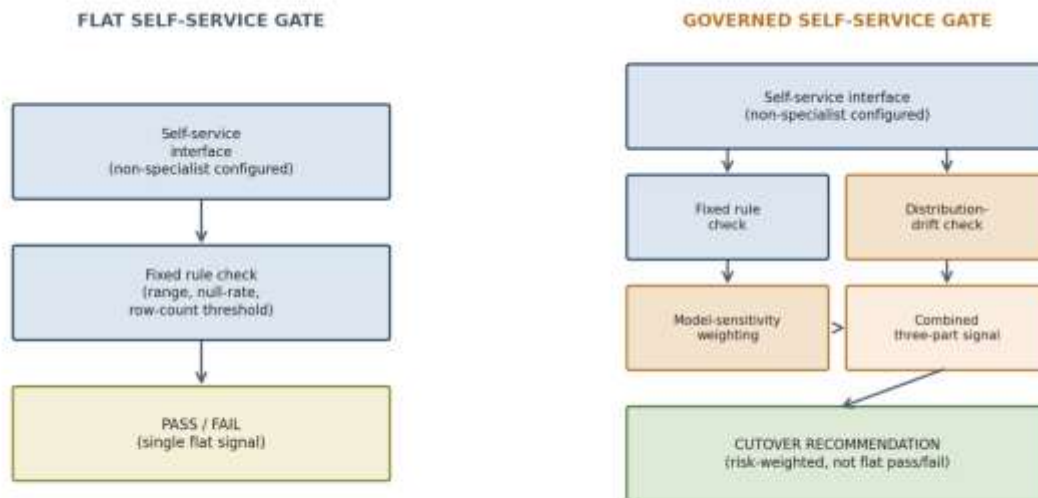


Figure 1. Architecture contrasting a flat self-service rule check against a governed gate with a drift detector and a model-sensitivity weighting layer positioned between the self-service interface and the cutover decision.

Table 1

Gate Type	Detects Shift?	Distribution	Weights Change by Model Relevance?	Signal Produced
Flat rule check	No	No	No	Single flat pass/fail
Drift-aware only	Yes	No	No	Pass/fail plus drift flag, unweighted
Model-aware only	No	No	Yes	Pass/fail plus relevance weight, no drift context
Fully governed	Yes	Yes	Yes	Risk-weighted, three-part signal

Table 1. Gate components compared across flat rule check, drift-aware, model-aware, and fully governed configurations.

Results and Discussion

What a flat rule check misses

A flat rule check fails in two directions I have seen happen. It can pass a batch whose values individually sit inside the configured range while the batch as a whole has become statistically unlike the data the rule was calibrated against - a distribution shift a range check was never built to see. And it can flag, or fail to flag, a change with no reference to whether that change matters: a null-rate spike in a field the downstream system ignores gets exactly the same escalation treatment as an identical spike in a field a critical report depends on entirely. Self-service governance research documents a similar pattern at the organizational level: decentralizing responsibility for running a check shifts accountability for data quality to users who often lack the specialist context to interpret what a check result actually means, without a corresponding shift in the sophistication of what the check itself detects (Clarke et al., 2016; Lennerholt & van Laere, 2019). But this tooling is more convenient to use. And detection does not become more adequate. And I have seen migration programmes get quietly burned in that gap.

The failure mode is usually invisible until it isn't. The gate passing every batch for a few months creates an implicit trust relationship. And that implicit trust relationship is what makes the eventual miss so expensive: nobody looks back at the check that always returned green. A slow distributional drift can sneak for longer than you'd think through an implicit trust wall before anything downstream visibly breaks, and by then the batches that would have made it visible are long gone. Apart from the technical drift, there is also organizational drift. If some non-specialist configured a rule a few months ago, and the profile of the data has since changed, that person has no incentive to go back and review their work.

On the other hand, the specialist who would notice the change is not part of the normal process. Self-service tooling is built to remove the specialist as a bottleneck, but it also removes them as a safety net. There is nothing in a flat rule check that provides a safety net.

What a distribution-aware gate adds

Concept drift detection provides exactly this missing functionality. It is not a new field, but a mature and well-understood area of research. In particular, adaptive windowing detectors guarantee that the probability of falsely detecting drift at any point in time is bounded below by a confidence level selected by the user. Furthermore, memory and per-item processing requirements grow logarithmically in window size (rather than stream size) (Bifet & Gavaldà, 2007). The runtime savings of this incremental KS test detection scheme are obvious. For instance, an incremental KS test was found to be two orders of magnitude faster than a conventional KS test in a benchmark where almost perfect classification was achieved by requesting the true labels for less than half of all incoming samples (Moreira dos Reis et al., 2016). The population stability index is a common metric used in industry for measuring population shift. A conventional interpretation is that movements below a value of 0.10 are negligible, between 0.10 and 0.25 are moderate, and above 0.25 are important. However, using these fixed cutoffs leads to false positives and false negatives that are not rate-constant over different sample sizes. Threshold values normalized for the sample size have also been proposed as an alternative (Yurdakul & Naranjo, 2020). But a rule tuned to a distribution from last month cannot decide whether batches have drifted this month. Drift detectors are what can answer that question. A drift detector is something the article does not have to invent, because it can rely on statistical tools that are already well-known in the concept-drift literature.

Table 2

Drift Scenario	Flat Threshold Outcome	Drift-Aware Gate Outcome
Values stay within range but batch distribution shifts	Passes silently; shift undetected	Flags divergence from calibration baseline
Sudden, large shift in a monitored field	Fails, but rule already stale for future batches	Fails and recalibrates detection baseline
Gradual, incremental shift over many batches	May never cross a fixed threshold	Adaptive windowing detects accumulated divergence

Table 2. Detection outcomes compared between a flat threshold and a drift-aware gate, across drift scenario types.

What a model-aware gate adds.

Feature-importance and model-sensitivity methods provide the second missing capability: a way to weight a detected change by how much it actually matters. SHAP attribution is, with formal proof, the only additive feature-attribution method that simultaneously satisfies local accuracy, missingness and consistency, the three properties that make an attribution trustworthy enough to act on rather than merely suggestive (Lundberg & Lee, 2017). It is known that naively computing permutation importance is biased towards categorical predictors with larger numbers of levels, but bias-corrected permutation importance can completely remove this artifact in simulation, and causes the classification error to drop about 10 percent relative to the naive version when used as an auxiliary for feature selection (Altmann et al., 2010). However, at the scale at which a real migration is likely to take place, it is too slow to run exact Shapley-based attribution over an end-to-end pipeline (which is why we have approximation methods). One method has been described as being up to four orders of magnitude faster than a standard Monte Carlo Shapley approximation while maintaining a level of accuracy at identifying which upstream records actually drove a downstream outcome (Karlaš et al., 2024). Methods that rank which features in the data are responsible for a model prediction may also be used within a migration cutover gate to judge which detected changes in the data require human intervention, and which may be logged and ignored.

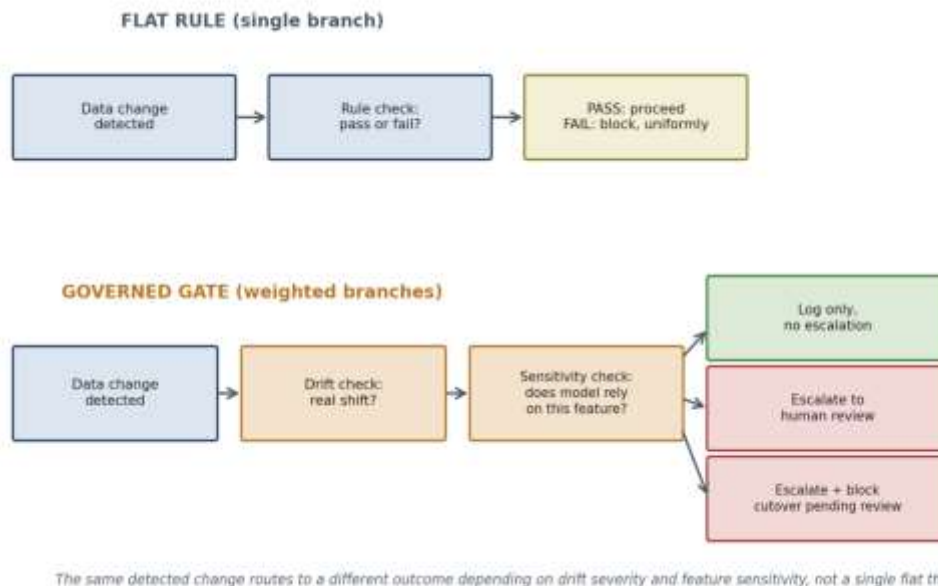


Figure 2. Decision flow showing a detected change routed through drift and sensitivity checks before reaching a cutover recommendation, contrasted with a flat rule's single pass/fail branch.

Accountability when the gate runs itself

But wiring a self-service gate to be both drift- and model-aware doesn't actually resolve the accountability issue a self-service gate raises. In fact, it makes it more of a problem (something I didn't fully appreciate until I saw it up close). Once a non-expert has performed the test and a cutover is taken based on that, someone must have responsibility for that decision. There is some research derived from self-service governance for analytics tools that provides insights into the practicalities of this accountability (Clarke et al., 2016; Lennert & van Laere, 2020). Work on risk management in the context of migration also provides useful analogies for cutover as a probabilistic judgment rather than a simple pass/fail determination (as is more easily represented in a three-way gate signal than a binary pass/fail signal), see for example Islam et al., 2017.

That said, a few well-understood stable tables will not justify the engineering for a drift detector and a sensitivity-weighting layer. In the literature on migrating data, guidelines for estimating risk are often confined to estimating how much to invest in the rigor of validation according to the size and variability of what is being migrated (Jamshidi et al., 2013; Islam et al., 2017). This does not mean that every self-service gate has to have both of these things bolted onto it, just that this particular decision should be made explicitly based on actual data volatility associated with the migration and downstream model dependency, and not just because flat rule check was the only thing that self-service tool shipped with. There is also a version of this argument which gets the emphasis backwards, and I want to head it off directly: wiring in drift-awareness and model-awareness does not mean handing the underlying statistical judgment to the non-specialist who configured the gate. The self-service interface allows the business analyst to choose what to check and to see the result. The drift detector and sensitivity weighting operate behind the scenes and influence the analyst's recommendation without requiring the analyst to understand a KS-test statistic or a Shapley value. Accessibility and adequacy are not in tension once the architecture is right. The non-specialist keeps their access. The check itself now knows things a flat rule never could.

Table 3

Migration Scale / Data Volatility	Recommended Gate Posture	Rationale
Small, stable, low-volatility tables	Flat rule check acceptable	Wiring cost exceeds expected benefit
Moderate scale, some volatility, model-consuming downstream	Add model-sensitivity weighting first	Cheapest of the two components to justify; highest payoff where a model consumes the output

Large-scale, high-volatility, or regulated migrations	Fully governed gate (drift-aware and model-aware)	Investment tracks risk; supports risk-weighted cutover judgment rather than a flat threshold
---	---	--

Table 3. Applicability matrix mapping migration scale and data volatility to recommended gate posture.

Conclusion

Self-service data quality tooling closes an access gap: more people can run the check without waiting for the special person who can set it up. It does not close the detection gap: whether the check detected a real distribution shift and whether the check weighed that shift by what that shift actually touched downstream. Both gaps have different fixes, and closing only the access gap can make for a false sense of governance if the detection gap is left open. This is the failure mode this article seeks to name. However, a governed self-service gate (which is distribution-aware, model-aware and access-aware), will give a non-specialist user a check whose answer actually corresponds to a determination of when it is safe to make the cutover. None of the techniques this article is drawing on to build that gate are new or exotic. Although drift detection and feature attribution are established with predictable properties, there are fewer precedents for connecting the two to find out what a migration cutover means. Closing that particular gap is a small engineering effort compared to the cost of a migration decision made on a check that only ever looked adequate.

Conflict of Interest

None declared.

Acknowledgment

None.

References

1. Altmann, A., Tološi, L., Sander, O., & Lengauer, T. (2010). Permutation importance: A corrected feature importance measure. *Bioinformatics*, 26(10), 1340–1347.
2. Baylor, D., Breck, E., Cheng, H.-T., Fiedel, N., Foo, C. Y., Haque, Z., Haykal, S., Ispir, M., Jain, V., Koc, L., Koo, C. Y., Lew, L., Mewald, C., Modi, A. N., Polyzotis, N., Ramesh, S., Roy, S., Whang, S. E., Wicke, M., Wilkiewicz, J., Zhang, X., & Zinkevich, M. (2017). TFX: A TensorFlow-based production-scale machine learning platform. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
3. Bifet, A., & Gavaldà, R. (2007). Learning from time-changing data with adaptive windowing. *Proceedings of the 2007 SIAM International Conference on Data Mining*, 443–448.
4. Clarke, P., Tyrrell, G., & Nagle, T. (2016). Governing self-service analytics. *Journal of Decision Systems*, 25(sup1), 145–159.
5. Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4), Article 44.
6. Islam, S., Fenz, S., Weippl, E., & Mouratidis, H. (2017). A risk management framework for cloud migration decision support. *Journal of Risk and Financial Management*, 10(2), Article 10.
7. Jamshidi, P., Ahmad, A., & Pahl, C. (2013). Cloud migration research: A systematic review. *IEEE Transactions on Cloud Computing*, 1(2), 142–157.
8. Karlaš, B., Dao, D., Interlandi, M., Li, B., Schelter, S., Wu, W., & Zhang, C. (2024). Data debugging with Shapley importance over end-to-end machine learning pipelines. *Proceedings of the International Conference on Learning Representations*.
9. Kourouklidis, P., Kolovos, D., Matragkas, N., & Noppen, J. (2020). Towards a low-code solution for monitoring machine learning model performance. *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*.
10. Lennerholt, C., & van Laere, J. (2019). Data access and data quality challenges of self-service business intelligence. *Proceedings of the 27th European Conference on Information Systems*, Research Paper 37.
11. Lennerholt, C., & van Laere, J. (2020). User-related challenges of self-service business intelligence. *Information Systems Management*, 38(4), 309–323.

12. Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 4765–4774.
13. Manjunath, T. N., & Hegadi, R. S. (2013). Data quality assessment model for data migration business enterprise. *International Journal of Engineering and Technology*, 5(1), 101–109.
14. Moreira dos Reis, D., Flach, P., Matwin, S., & Batista, G. (2016). Fast unsupervised online drift detection using incremental Kolmogorov-Smirnov test. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1545–1554.
15. Redman, T. C. (1998). The impact of poor data quality on the typical enterprise. *Communications of the ACM*, 41(2), 79–82.
16. Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.-F., & Dennison, D. (2015). Hidden technical debt in machine learning systems. *Advances in Neural Information Processing Systems*, 28.
17. Yurdakul, B., & Naranjo, J. (2020). Statistical properties of the population stability index. *Journal of Risk Model Validation*, 14(4), 89–100.