Don’t Let Models Derail You

Strategies to Control Risk with Model Monitoring

WHITEPAPER

Table of Contents

Executive Summary

For business leaders, the north star aspiration is to build a "model-driven" business. To achieve this vision, leaders hope to weave models into the fabric of business operations to disrupt industries, beat competitors, and drive unprecedented growth. And it is not a select few who see models as essential. 75% of executives responding to a recent survey by Accenture believe that their companies will most likely go out of business if they can't scale data science successfully within the next five years. But amidst the rush to become a model-driven business, leaders should ask: Am I doing enough to prevent the real and serious consequences of bad models? Experienced executives are well aware of the risk. A recent survey by Wakefield Research and Domino Data Lab revealed that 82% of data executives at US companies with more than $1 billion of annual revenue say their company leadership should be concerned that bad or failing models could lead to severe consequences for the company. Outcomes of bad models include:

- Loss of brand reputation - Amazon's gender-biased AI recruiter
- Financial loss - Investor Sues After an AI's Automated Trades Cost Him $20 Million
- Loss of freedom - AI is sending the wrong people to jail
- Physical harm or loss of life - Uber Self Driving Car Fatality

Concern stimulates action, and there are three options for business leaders. One is to pray and hope that models won't fail (and therefore do nothing). Two is to delegate the job of watching over deployed models to each business unit and its respective AI team to figure it out on their own (and maybe accomplish something). Three is systemic: to implement automated, continuous, and standardized model monitoring across the enterprise.

One is crazy. Two is lazy. Three is the way forward and should be based on the technology, principles, and best practices of Enterprise MLOps. To help today's business and IT leaders understand how to prevent unintended consequences from bad models, our whitepaper describes how and why models can turn bad. We then review instructive use cases of model-risk prevention. We conclude with a summary of recommendations for leaders to help their organizations prevent fallout from bad models.

Why Good Models Go Bad

To understand how to prevent negative impact from models, it helps to understand why they go bad in the first place.

Models are probabilistic and trained on historical data. This means models deployed into production carry forward characteristics of the data used to train them, including any hidden biases. It also means their output will change if the relationship between the incoming data and the predicted target drift apart.

Catching the early warning signs of model degradation requires a new class of monitoring systems. Instead of looking only at typical software attributes such as usage, latency, uptime, and cost metrics, model-based systems must also consider data quality, data drift, and model quality – all indicators of a model's "health" or intrinsic goodness.

Even the best performing model will eventually degrade for a variety of reasons: changes to products or policies can affect how customers behave; adversarial actors can adapt their behavior; data pipelines can break; and sometimes the world simply evolves. Any of these factors lead to data drift and concept drift, which can result in a drop of predictive accuracy. For example, if a cellular provider raises a limit on the number of text messages deliverable to a device per minute, that may affect security risk models or fraud models that rely on message frequency as an input.

Or consider a model labeling an image of a father working on his laptop at home while his son plays beside him. If the model was trained prior to the recent shift to at-home work, it may categorize this as "leisure" or "relaxation" rather than "work" or "office," despite the overwhelming majority of white-collar workers now working from home. As the world changes, models degrade.

Root Causes of Model Degradation

Data Drift – The patterns in production data that a deployed model uses for predictions gradually diverge from the patterns in the model's original training data, which lowers predictive power of the model.

Concept Drift – Occurs when expectations of what constitutes a correct prediction change over time – despite there being no change in the input data distribution.

How Organizations Deal with Model Degradation

Pray and Hope for the Best

- Works…until it doesn't (at the most inconvenient time)

Re-Train Models Periodically

- Too low/high a frequency (ineffective or added costs)
- How do you know re-training was effective?
- What about abrupt drift?

Do Ad-Hoc Drift Tests

- Inefficient and unreliable
- Non-standardized. Each team (every person!) does their own thing
- Lack consistent stakeholder visibility

Continuous and Standardized Monitoring

- Notifies stakeholder immediately
- Always reliable and consistent
- Validates that model remediation worked

Lessons from a Good Model Going Bad

To illustrate how business results can veer from predictions based on model degradation, let us learn from lessons experienced by a model-driven leader: Instacart. In this case study, we see why an automated model-monitoring solution could be helpful in rapidly identifying model deviations and getting business back on track.

Empty Carts for Instacart

Online grocery shopping service Instacart used a data science model that usually enjoyed a 93% accuracy rate for predicting whether a particular product would be available at a given store. In March 2020, that accuracy rate suddenly plunged to 61% for many products after customers drastically changed their shopping behavior in the face of the nascent COVID-19 pandemic.

Under normal circumstances, shoppers would typically buy things like toilet paper infrequently. But after pandemic lockdowns were in effect, within the space of about a week they were wiping out store supplies of goods such as toilet paper, hand sanitizer, eggs, flour, and other household essentials.

Instacart adapted by changing the timescale of the shopping data it fed into its AI models from weeks to only ten days, making its models more responsive to quickly changing shopping habits. They made a tradeoff between the volume of data used to train their model and the "freshness" of data, Instacart's machine learning director Sharath Rao told Fortune magazine

Framework for Keeping Models on Track

Deploying a successful model monitoring effort fits squarely in larger transformative initiatives as executives seek to scale model velocity. For successful model monitoring, consider a six-point framework for maintaining AI/ML models. The framework takes a holistic approach by describing how model monitoring fits into a broader initiative to maintain models at scale in an automated, standardized way across the enterprise. The six points are as follows.

1. **Well-documented purpose** – Models should align with business goals and purposes; otherwise, they will grow stale and lose potency.

2. **Data lineage details** – Data underlying models should be captured in detail, including how the data was prepared to ensure a model can be reproduced and trusted. This information also may be useful for documenting responses to auditors.

3. **A full lifecycle tracking system** – The Enterprise MLOps platform should automatically link model runs with specific data versions to document changes made to model elements during the experimental build process. Recording the evolution of a model helps elucidate what the organization is modeling, why they were built, and how adjustments in data inputs and assumptions contributed to the current state of a model.

4. **A model registry –** Used to track the model version history where each version is fully reproducible with the same elements performed by experiments in changing data, code, software, and hardware platforms.

5. **Validation routines –** These document code reviews, report on the various explanations about ethical and bias checks and obtain the stamp of approval from its users. Validation routines include service level agreements and other functional tests and comments about the model's general production readiness.

6. **An open model monitoring system –** Captures data drift, ground truth, measurement accuracy, and provides drill-down capabilities to troubleshoot signals and detect anomalies. These systems can automatically alert stakeholders when certain thresholds are exceeded

How to Ask About Eight Top Capabilities for Detecting Model Drift

- ✓ Is model drift monitoring automatic?
- ✓ Are stakeholders automatically notified of drift?
- ✓ Does monitoring cover models deployed in non-standard form factors?
- ✓ Can technical and business stakeholders view model health in one simple report?
- ✓ Are models tracked in a model repository that facilitates versioning and deployment?
- ✓ Are model explainability and bias checking covered?
- ✓ Does the system document model lineage?
- ✓ Is a model building workbench with CI/CD capabilities integrated for model refactoring?

Under the Hood of Model Monitoring

Let's turn our focus to key issues for implementing the framework's step six, model monitoring.

McKinsey states in their State of AI in 2020 global survey, there are three clear differences in how high-performing companies approach AI compared to laggards. The study notes:

High performers understand how frequently models need to be updated and refresh them based on clearly defined criteria.

They use automated tools to produce and test models.

And they track model performance and explanations to ensure that outcomes and/or models improve over time.

Automation must be core to any scalable approach for tracking model drift and model quality across all models, features and predictions. Hence the use of an Enterprise MLOps platform to implement automation of required capabilities for model monitoring.

Models exist in a variety of locations and form factors. They can live on a server sending predictions to a database, on a scalable compute fabric with super low latency, or be embedded in complex applications requiring application log mining to cull their inputs and outputs. In each case, we need the data flowing into and out of each model; these data come in a variety of shapes and sizes. A monitoring system must be able to gather data from anywhere, compute drift-like metrics, and present those metrics in a single pane of glass to stakeholders.

If models turn bad, they'll need to be retrained or completely rebuilt. A model repository can provide a central home for models to keep track of this monitoring, updating, versioning, and redeploying. A strong system to manage model and data lineage is a key feature of such model repositories. So is integration with model-building workbenches, code repositories, and pipeline tools to aid in model refactoring via CI/CD principles.

Finally, even when data and concept drift are not present, model bias can cause massive fallout. Model explainability should be a key feature in a model monitoring initiative.

4 Ways to Remediate Bad Models

Retrain

If a model has drifted, it can be retrained with fresher data, along with its associated ground truth labels, that is more representative of the prediction data. However, in cases where ground truth data is available, the training data set can be curated to mimic the distribution of prediction data, thereby reducing drift.

Rollback

Sometimes rolling back to a previous version of the model can fix performance issues. To enable a rollback, you need to maintain an archive of each version of the model. You can then evaluate the performance of each prior model version against the current production version by simulating how it would have performed with the same inference data. If you find a prior version that performs better than the current model version, you can then deploy it as the champion model in production.

Fix the Pipeline

While draft may occur because the ground truth data has changed, sometimes it happens when unforeseen changes occur in the upstream data pipeline feeding prediction data into a model. Retraining with fresher data sourced from the data pipeline may fix the model, or fixing the data pipeline itself may be easier.

Repair the Model

In some instances you just need to repair a model in a development environment. To diagnose the cause of the model degradation it helps to use a platform that supports reproducibility, where you can effectively simulate the production environment in a development setting. Once a suspected cause is identified you can choose the best method for repairing the model, whether modifying hyperparameters, or something more invasive.

Conclusion

Model monitoring is akin to using antivirus software on a laptop: everything runs fine without protection – until it doesn't. The instant a computer is infected, it is obvious antivirus protection was needed all along. In a similar way, many organizations are running models in production today and are betting major aspects of their business on the success of those models. Operating a model-driven business without an automated, continuous, and standardized model monitoring process exposes companies to many risks of failing models. If your organization is looking to automate, centralize, or standardize model monitoring capabilities, we invite you to learn about specific best practices for model monitoring at scale in this article, Maintaining Data Science at Scale. Leaders interested in the value of an integrated, end-to-end approach to data science may also be interested in the Forrester Consulting's "The Total Economic Impact™ of the Domino Enterprise MLOps Platform" for more information about how to create your own analytical flywheel and achieve real breakthroughs in data science transformation initiatives.

About Domino

Domino powers model-driven businesses with its leading Enterprise MLOps platform that accelerates the development and deployment of data science work while increasing collaboration and governance. More than 20 percent of the Fortune 100 count on Domino to help scale data science, turning it into a competitive advantage. Founded in 2013, Domino is backed by Sequoia Capital and other leading investors. For more information, visit dominodatalab.com.