Chapter 2. Ten Things to Know About ModelOps

1. ModelOps Is the Enterprise "Operating System" for AI/ML Algorithms

While AI/ML and deep learning methods get the most attention in the press, blogs, and on the web, the reality is that for the majority of applications the model *authoring* may be the simplest part. The supporting logic, processing, deployment, and controls—and in particular setting up an efficient delivery system of solutions infused by AI/ML—are the much bigger challenge.

ModelOps, therefore, is a kind of operating system for AI/ML models. That is, it creates a platform and set of tools to deploy algorithms in a reliable, repeatable, compliant, safe, and efficient way. Like the operating system on a laptop or mobile phone, it provides access to the right algorithm and to the right place at the right time when it is needed.

Many Types of Models

To reiterate, *models* are just transformations of data, in order to extract useful information. There are many types of models that can convey useful information, help make better decisions, or drive effective automation. Continuing with the operating system metaphor, it is a mistake to limit ModelOps to just one "app": ModelOps tools must deploy AI algorithms built on any platform or provided by any cloud service.

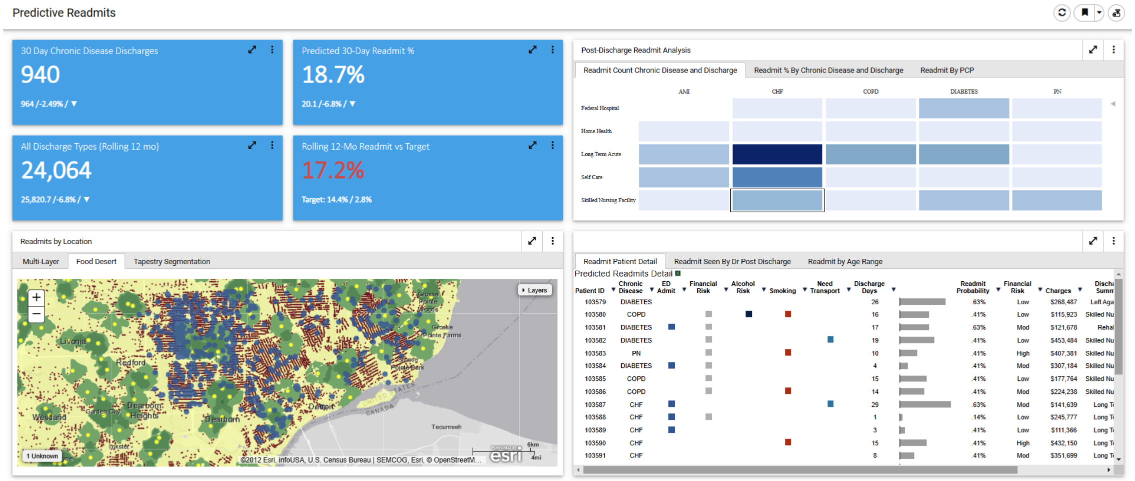
Any statistical analysis, aggregation, set of rules, summary, or BI dashboard represents information, abstractions, and inferences from observed data that guide strategies and decisions. All of the above are "models," and should be managed and governed in the same way to ensure accuracy; consistency; compliance with regulatory, best-practice, and policy guidelines; efficiency; profitability, and so on. Next, we'll dig into several examples of models.

BI dashboards

Traditional BI tools summarize means, percentages, trends, and so on, over meaningful categories. For example, sales by customer segment and region may be summarized in a dashboard in order *to inform strategy decisions*, solve a problem, or just understand a problem or domain. Those dashboards act as models because they assume that the information summarized in them—derived from historical data—is diagnostic for predicting future events, conditions, or trends. But what if the model is wrong and the information summarized in the BI dashboard is not related to what drives consumer choices across regions or segments? Or, what if the segments are not defined correctly, or if the data is not reliable? What happens if the consumer segmentations are based on categories that could be deemed offensive by some, or are outright not permitted by law? Many organizations are run by BI dashboards, and it is a bit frightening to think what can happen if the models they rely on are wrong. That is why they must be managed by ModelOps.

Intelligent BI

This problem is greatly compounded when BI dashboards are connected to AI/ML prediction models, which is increasingly the case. In fact, BI and analytics *are* merging (see, for example, Gartner 2021a). Figure 2-1 shows a typical dashboard that incorporates traditional reporting elements, together with AI-based elements for forecasting. Not only does this view provide actionable insights regarding current hospital discharges but also *expected/predicted* readmission rates, along with analytics-based insights regarding the dominant causes of readmissions (see the heatmap). The dashboard depicts the hospital's *model* of what drives the financial impact of current and expected readmissions. Modern BI platforms from most vendors provide end-to-end AI/ML for business users: users can select data, build (automatically) AI/ML models, explore predictions and related results, and share them with other stakeholders and decision makers. Again, BI dashboards are models; when there are thousands of models that the organization relies on, they must be managed via ModelOps.

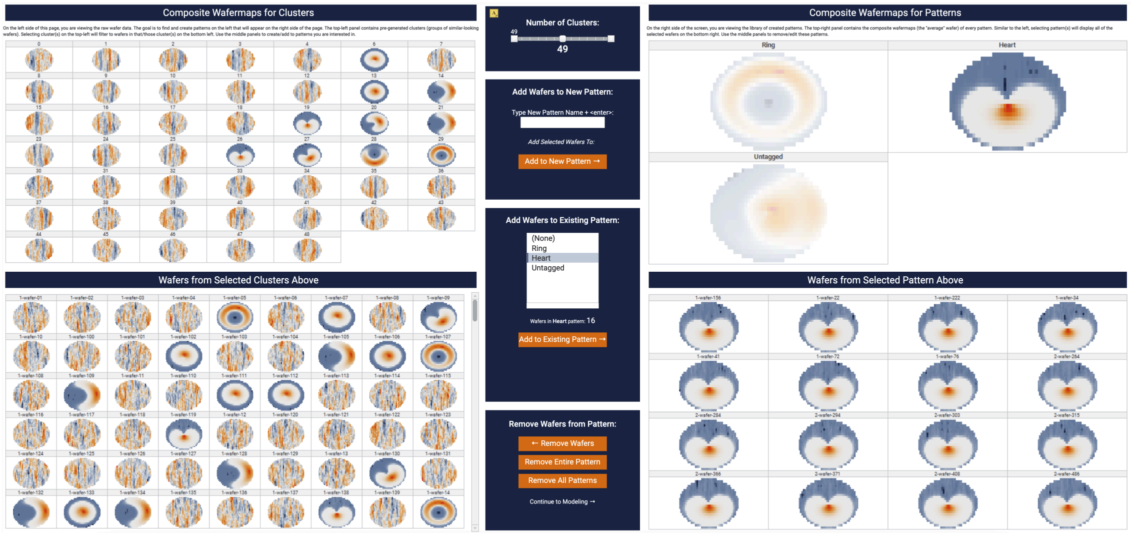


Predicted Readmits Detail

Patient ID	Chronic Disease	ED Admit	Financial Risk	Alcohol Risk	Smoking	Need Transport	Discharge Days	Readmit Probability	Financial Risk	Charges	Discha Summ
103579	DIABETES						26	.63%	Low	$268,487	Left Aga
103580	COPD		■	■	■		16	.41%	Low	$115,923	Skilled Nu
103581	DIABETES	■	■				17	.63%	Mod	$121,678	Rehal
103582	DIABETES		■			■	19	.41%	Low	$453,484	Skilled Nu
103583	PN		■		■		10	.41%	High	$407,381	Skilled Nu
103584	DIABETES	■	■				4	.41%	Mod	$307,184	Skilled Nu
103585	COPD		■				15	.41%	Low	$177,764	Skilled Nu
103586	COPD		■		■		14	.41%	Mod	$224,238	Skilled Nu
103587	CHF	■				■	29	.63%	Mod	$141,639	Long T
103588	CHF	■	■				1	.14%	Low	$245,777	Long T
103589	CHF	■					3	.41%	Low	$111,366	Long T
103590	CHF				■		15	.41%	High	$432,150	Long T
103591	CHF						8	.41%	Mod	$351,699	Long T

Figure 2-1. Example dashboard delivering predictive readmit info via TIBCO Omni-HealthData® solution

BI dashboards are just visualizations of underlying statistical or rule-based summaries. Regardless of how those are rendered—visually or as tables—if they are to be used to inform or drive organizational processes, they need to be vetted and managed. This is true in people-facing businesses—when delivering offers to potential customers, approving or denying credit to applicants for loans, or evaluating insurance claims. This is also true in manufacturing applications (for example, when implementing statistical process control charts and summaries, or when relying on dashboards for root cause analysis of quality problems). As an example, Figure 2-2 shows a BI-like application from the semiconductor industry for "wafer mapping": the goal is to identify in batches of semiconductor wafers common fault patterns of various types. Users can select wafers with specific fault patterns or cluster common patterns to identify common/repeated fault patterns. This information can then be related to upstream machine and sensor data to guide root cause analyses and preventive maintenance efforts—ultimately to maximize yield.


Composite Wafermaps for Clusters
Number of Clusters:
49
49
Add Wafers to New Pattern:
Type New Pattern Name + <enter>:
Add Selected Wafers To:
Add to New Pattern →
Add Wafers to Existing Pattern:
(None)
Ring
Heart
Untagged
Wafers in Heart pattern: 16
Add to Existing Pattern →
Remove Wafers from Pattern:
← Remove Wafers
Remove Entire Pattern
Remove All Patterns
Continue to Modeling →
Composite Wafermaps for Patterns
Ring
Heart
Untagged
Wafers from Selected Clusters Above
Wafers from Selected Pattern Above

On the left side of this page, you are viewing the raw wafer data. The goal is to find and create patterns on the left that will appear on the right side of the page. The top-left panel contains pre-generated clusters (groups of similar-looking wafers). Selecting cluster(s) on the top-left will filter to wafers in that/those cluster(s) on the bottom left. Use the middle panels to create/add to patterns you are interested in.

On the right side of the screen, you are viewing the library of created patterns. The top-right panel contains the composite wafermaps (the "average" wafer) of every pattern. Similar to the left, selecting pattern(s) will display all of the selected wafers on the bottom right. Use the middle panels to remove/edit these patterns.

Figure 2-2. Intelligent TIBCO Spotfire® BI/modeling dashboard for wafer classification and modeling

AI/ML prediction models, feature engineering models, and transformations

Of course, AI/ML prediction models as well as statistical models are what is most commonly referred to as “models,” and ModelOps is the required tool to manage them. But in practice, it is *model scoring flows* and *model scoring pipelines* (see the definition of terms at the end of Chapter 1) that perform all computations required to turn data into information, recommendations, or automated actions. When there are thousands of model flows and pipelines to be managed, then it is important to manage steps as reusable operations. In practice, many of the transformations, data preparation, and other steps required before a model can be built or scored are common across the applications in similar domains. For example, credit risk models typically require the computation for each applicant or account of the current debt from all creditors. Those computations (data transformations or creation of new *features*) are applicable regardless, whether applied to the determination of a limit on a credit card or to an application for a mortgage loan. Such common and useful data preparation, transformation, and rule-based steps also must be managed through ModelOps.

Figure 2-3 shows a simple scoring pipeline to transform data for a typical credit-default-risk scoring model—often referred to as a *credit scoring model* to determine applicants’ eligibility for credit or certain credit limits.

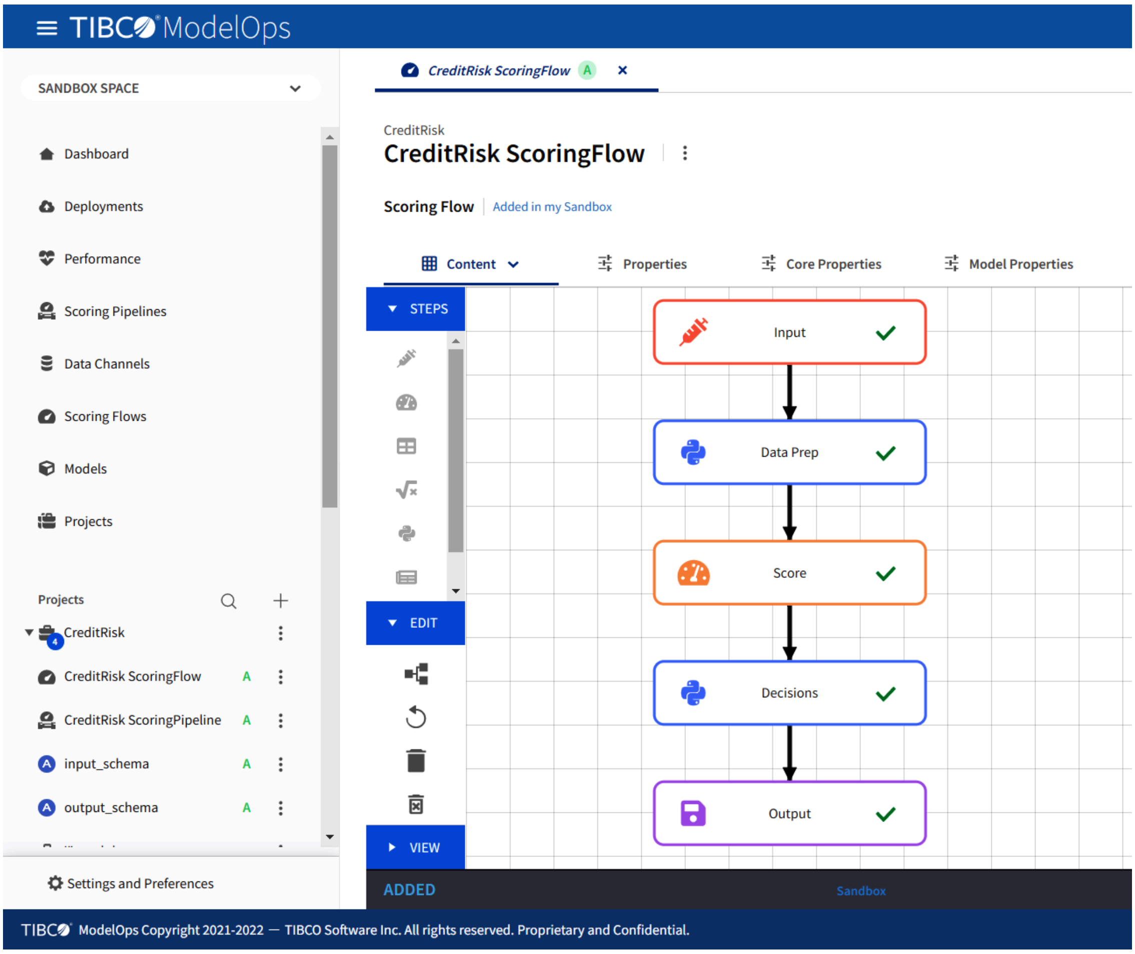


Figure 2-3. A simple scoring pipeline in TIBCO ModelOps

Data enters this pipeline as *Input*. The *Data Prep* step will clean and prepare the data, for example, identify erroneous inputs and recode or flag them. Also, and typically in credit scoring applications, segmentation rules are applied in this step, for example, to age in years to an age *category*. In general, data fields are transformed and/or combined to create *features* for the actual prediction mode. The transformed data is passed to a *Score* step, which is the actual prediction model that converts the upstream data to predictions and prediction probabilities. Finally, the *Decisions* step will take the numeric default-risk predictions and convert them to actual final recommendations or decisions. This example illustrates a specific example of the typical steps that are required to convert raw data into final, actionable information. However, in practically all deployments scenarios for predictive analytics, these steps will be required.

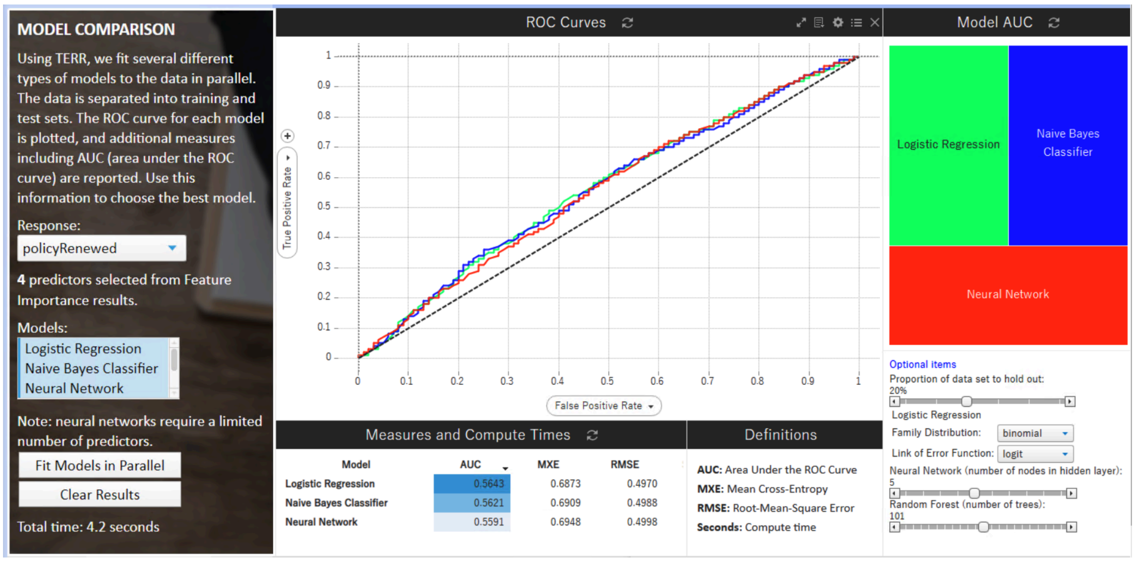
Many Different People/Roles Build Models

The previous section pertains to the different computational steps that represent or are part of "models." Returning to the operating system metaphor, there are many different types and services and apps required; an operating system must coordinate all of them. The next interesting and often misunderstood questions are: Who ensures they are secure? Who mediates their deployment? How do we ensure they are safe and reliable?

Data scientists are specialists and won't be the only ones to author your models

In many ways, the delivery of results and useful insights from models can be thought of as a "delivery service." What *specifically* is being delivered is outside the purview of that service; what matters is that the delivery is smooth, fast and efficient, consistent, and robust. Looking at it this way is consistent with the expanded view of "models" discussed earlier (see "Many Types of Models") and expands the organizational roles and personas who can and increasingly *do* contribute models, to include business users who are *not* data scientists or AI specialists. In fact, partly because data scientists are currently in incredible demand and commanding very high salaries, anecdotal evidence seems to support that many organizations have curtailed their investment into data scientists after failing to realize expected returns on investment.

At the same time, great progress has been made in many open source and commercial data science authoring platforms to automate much or nearly all of the model-building tasks, enabling business stakeholders and users (i.e., nondata scientists or AI experts) to contribute useful models. For example, intelligent BI tools can incorporate end-to-end AI/ML model building and sharing to empower business users with faster, better, and AI/ML-driven insights. Figure 2-4 illustrates such an interactive BI-like dashboard that allows business users and nondata scientists to build ("author") prediction models using a point-and-click user interface, for example, to select predictors and prediction algorithms, or to auto-build a best prediction model.



MODEL COMPARISON

Using TERR, we fit several different types of models to the data in parallel. The data is separated into training and test sets. The ROC curve for each model is plotted, and additional measures including AUC (area under the ROC curve) are reported. Use this information to choose the best model.

Response: policyRenewed

4 predictors selected from Feature Importance results.

Models:
Logistic Regression
Naive Bayes Classifier
Neural Network

Note: neural networks require a limited number of predictors.

Total time: 4.2 seconds

Measures and Compute Times

Model	AUC	MXE	RMSE
Logistic Regression	0.5643	0.6873	0.4970
Naive Bayes Classifier	0.5621	0.6909	0.4988
Neural Network	0.5591	0.6948	0.4998

Definitions

AUC: Area Under the ROC Curve
MXE: Mean Cross-Entropy
RMSE: Root-Mean-Square Error
Seconds: Compute time

Optional items
Proportion of data set to hold out: 20%
Logistic Regression
Family Distribution: binomial
Link of Error Function: logit
Neural Network (number of nodes in hidden layer): 5
Random Forest (number of trees): 101

Figure 2-4. Rapid modeling via TIBCO® Dynamic Pricing Accelerator

Linking AI/ML to organizational outcomes

To be clear, data scientists and experts can be useful and are often necessary when exploring new domains, innovative applications and approaches, or building something “special.” But the reality and common experience is that much (and often most) of the value of AI/ML modeling is derived by infusing large numbers of existing processes and decisions with AI/ML models that work “well” and better (often much better) than before. ModelOps provides the operating system that enables this, by vetting, managing, monitoring, and evaluating the various types of models, data preparation steps, resulting reports and dashboards, and KPIs.

ModelOps is the link between AI/ML operations at scale to value realized at the bottom line

ModelOps connects all stakeholders, contributors, users, domain experts, data and data science experts, and executives to insights, actions, and value. Thinking of ModelOps as a tool for data scientists to collaborate can be a costly mistake, because it disconnects and places one possible group of contributors in silos from the value chain that is required to make AI/ML useful. The leading industry analysts have consistently observed that most models never make it into production and achieve the desired business impacts. And the ones that do go to production take much longer to get there than desired and expected. One of the factors is the lack of collaboration between the data scientist and other stakeholders like the business/domain experts and IT. Many of the points discussed below will expand on this scope.

2. There Are Many Algorithms and Many Model Formats: Deliver Them All

Operating systems reliably deliver access to computing resources at the right location when they are needed. Just like the operating system delivers access to apps created anywhere, by any developer, ModelOps delivers various types of AI/ML algorithms created in different formats, by different authors.

Data scientists use various tools—commercial, open source, or models purchased from third parties. Also, scoring pipelines are built from multiple reusable steps for data preparation, feature engineering, scoring, and post-scoring decisioning. So, support for various model formats—and pipelines built from steps authored in different formats—is *required*.

Model Formats

To reiterate, a model is a transformation of data to indicators, values, or other outputs that contain the useful information derived from the data. Those transformations can be formatted and "encapsulated" in many different ways. Constraining a ModelOps solution to only a subset of them will limit which models can or cannot be used.

For example, Python is a popular programming language and authoring tool that programmers and data scientists use to build models. So, supporting the deployment of (reusable) Python data preparation steps, transformations, or features (result of feature engineering) is important. But statisticians often require specific functionality only found in R (a popular programming language for statistical modeling). Also, as discussed previously in point 1, data scientists, programmers, and statisticians may increasingly become *not* the primary authors of models; instead, models will be built by business users and stakeholders (sometimes called the *citizen data scientist*; see, for example, Gartner 2021c). Their models may be made available to ModelOps in some proprietary formats (for example, as TIBCO Statistica® workflows), or as an API and through function calls supported by a third-party web service. There are also other formats for prediction models that aspire to be portable across applications, such as PMML (Predictive Model Markup Language) and PFA

(Portable Format for Analytics) (see Data Mining Group 2021), or ONNX (Open Neural Network Exchange) (see ONNX 2021).

An effective ModelOps solution must be architected to include an extensible model format framework, so that any and all such current or future formats can be supported.

Support Your Model Builders, Don’t Constrain Them

The corollary is that by supporting all formats of models, fewer constraints are placed on the model-building process and the people and roles that support it. To reiterate, ModelOps is the operating system for delivering AI/ML and analytics to the business, in a way that delivers value. ModelOps should *not* be a vehicle for delivering AI/ML models for data scientist experts, in their preferred environment, language, or platform. Doing so would only encourage disconnects and deepen the divide between business goals and data scientists.

3. First, Do No Harm: Anticipate Risks Rather Than React to Damage Done

While regulatory frameworks for the use of AI/ML models—when applied to humans—are being proposed or in effect around the world, the bottom line is that most of those regulations are the result of “damage” that has happened already. Regulation is usually reactive, not proactive.

It is important to think of the application of AI/ML models in terms of risk management. ModelOps tools and procedures should include mechanisms that anticipate and gauge risks to reputation and customer commitment, and protect against waking sleeping dogs (negative uplift)—the known phenomenon when customers can become mindful in response to some service upgrade offer, evaluate alternatives, and cancel all existing services (“I did not realize how much I paid for the service in the first place”; see, for example, Firn 2020 for an overview).

There are many ways that models can cause harm.

Offending Customers

A model may be perfectly accurate for predicting customer propensity and be consistent with regulatory requirements. Yet when the model is deployed to the real world it may cause significant negative unintended side effects: customers can be offended, feel unfairly treated, or reconsider their relationship and commitments to a brand or service.

A famous example was the case of a national retailer that successfully predicted who among their customers was likely pregnant and provided those customers (customer *households*) with relevant offers (such as coupons for diapers or baby food). The unanticipated problem was that those customers in some cases may not have been aware of their state themselves, nor did their loved ones, guardians, or parents (in the case of minors; see Hill 2012). The publicity that ensued was definitely unanticipated, unwanted, and likely damaging.

Risky Predictions for Manufacturing

These types of problems can also occur when AI/ML models are applied to manufacturing. For example, one of the authors of this book early in their career was involved in a consulting project aimed at optimizing combustion processes for low emissions. A complex neural network produced near-perfect predictions based on historical data, and using that model certain process inputs were predicted to lower emissions significantly. During the implementation of the model-based predictions, the equipment operators in the control room were asked to set those process inputs for a test. As it turned out, in order to do so, they needed to “find” a very large wrench and manually adjust valves in a generally inaccessible part of the plant. In other words, this had never been done before, and making such adjustments seemed very risky to the experienced supervising process engineers.

As it turned out and after looking at the historical data from which the neural network model was built, there was indeed very little empirical data in the specific area (combinations of values for process inputs). In fact, the prediction of lower emissions relied on the neural network’s ability to interpolate from adjacent points (combination of process input values)—but there was no direct evidence in the data that this would work.

More careful analyses and experimentation revealed that the solution suggested by the neural net could have done some serious damage to the furnace.

Effective model monitoring can watch for such actual “local lack of evidence” by not only considering the absolute permissible ranges of input values, but also the similarities of input values for multiple predictor variables to what has previously been observed.

Considering Wrong Predictions

In general, model deployments through ModelOps processes and tools should be configurable so that the risks associated with wrong predictions can be anticipated and assessed. This is obvious when AI/ML is applied to predict, for example, the efficacy of a medical procedure or therapeutic regimen: what happens if it doesn’t work, and what are the possible side effects. But this is generally important when deploying and managing models that target humans.

For example, one of the authors of this book used to receive catalogs through the mail for self-defense weapons, brass knuckle rings, assault knives, and other “fighting gear.” This was clearly a wrong prediction of that author’s “propensity for hand-to-hand combat.” As a professional involved for many years in building propensity models for various purposes, a first point of curiosity was what specific data in what database may have led to the erroneous prediction. But such inadvertent consequences of wrong predictions can also be embarrassing, offensive, discriminatory, unfairly withhold some benefit, prevent a capable job applicant from finding a job, and so on.

Anticipating Risk

To reiterate, the tools and methods for deploying and managing models should be able to surface the risks that may be associated with a model *before* it is put into production. That aspect of predictive models and the consequences of their deployment has received relatively little attention so far, but it will become important as thousands of models are managed that are driving all critical processes of an organization. For starters, a flexible “smart” BI tool should be integrated with or at least connected to models, their metadata, and their predictions in historical and most recent data. Predictive model accuracy is only one measure that *can* be important; in reality it is rarely the most important metric. For example, predictive models—even when they are accurate—can cause grave damage to an organization and its reputation.

4. Aim to Build Your Organization’s IP Secret Sauce Through Repeatable Processing Steps

Ultimately, the value of AI/ML models is a function of how much better they are and how closely they align with the unique capabilities and goals of your organization. In other words, you want to be better than your competition with respect to how “smart” and effective your AI/ML models are. Building out repositories of reusable steps for scoring pipelines that work and differentiate means you are building unique intellectual property (IP) critical for your organization’s success and perhaps survival.

Previous sections in this book discussed the broad scope of “models,” to include data preparation, feature engineering, prediction models and pipelines, and ways to transform insights into actions delivered to the right place at the right time. This way of looking at model scoring pipelines will unlock the real value for organizations looking for competitive differentiation. For example, imagine if every insurer predicted insurance risk exactly in the same way. It would be very difficult to differentiate competitive insurance premiums and remain profitable.

Managing and Accumulating IP Through Models

Many high-tech manufacturers of electronic components and devices or other complex machinery consider it as critical IP how they monitor and analyze data, predict quality, maintenance needs, and so on. In some cases, that IP can be the *defining differentiation* that makes those manufacturers successful. This means that the ability to build out, document, retain, and refine certain computational approaches (models) is critically important.

Again, ModelOps should be thought of as the operating system enabling analytics and AI/ML as key organizational activities. As described earlier in this book, models can come in various forms and formats, and should be built from reusable steps efficiently assembled and supporting many model scoring pipelines. Effective ModelOps thus supports cumulative IP. When done correctly, ModelOps will be the organizing deployment system for an organization's all-things-analytics IP.

Disorganized Analytics Across Silos

Too many organizations today implement their analytics activities by department, plant location, and so on. In other words, many organizations do not have a mechanism to accumulate (for future applications) analytic steps and computations that worked well for particular applications. A particularly experienced engineer might drive significant innovations, but when that engineer leaves, some of the supporting knowledge that was created and is needed to maintain that IP leaves with them. As a result, those left behind have no repository to leverage and build on the insights and best practices possibly derived over decades.

To reiterate, the successful implementation of analytics and AI/ML is no longer the singular domain of data scientists and their tools. Instead, there are many organizational roles and stakeholders required in order to bring a model to production in a way that aligns with organizational goals and KPIs. To make AI/ML successful across the organization, ModelOps must manage the IP contributed by many individuals over time and combine that IP into innovative pipelines consisting of reusable steps.

5. ModelOps Is About Maintaining and Expanding Your IP

The value of IP usually fades over time. This is also true for AI/ML models and scoring pipelines and the reusable steps that make up a scoring pipeline. For example, when first discovered, a certain customer segmentation model that puts prospects into buckets with different propensities may be incredibly useful. But over time, the usefulness of any one customer segmentation model for predicting successful best-next-action/offers can change quickly. Effective model management is not just about monitoring model accuracy; it is about managing projects and best practices, and developing best practices about what works and what doesn't, why, and for how long. Effective management and deployment of models must be agile, efficient, and automated as much as possible.

Model Decay (Drift): Changing Populations, Data, and Relationships

First, here is a brief overview of the factors that can make prediction models less accurate and effective over time. Typical AI/ML models are computed from historical data by identifying repeated patterns diagnostic of the future. For example, a model based on historical data may predict credit default risk from an applicant's occupation, amount of revolving debt, and income. This model may have been derived from the historically typical population of applicants, consisting, for example, of 40% of applicants younger than 35 years and 60% of applicants 35 years old or older. Over time and inevitably, various things will affect the accuracy of the model; due to changing conditions and circumstances around the respective application domain, model accuracy will drift and decay.

Population drift

The age distribution across typical applicants for credit can change and change quickly. New professionals may relocate to the geographic area served by the bank; a new development or expansion by existing large employers or builders may attract a larger number of older individuals, and so on. In other words, the characteristics of populations can drift, and in an extreme case, a model built primarily on data of older applicants is now used to predict the credit worthiness of younger applicants. Obviously, this will impact the validity and accuracy of the prediction model.

Concept drift

Relationships between the inputs to a model and their relationships to the predicted values (e.g., credit default risk) may change over time. For example, consumer debt may be negatively related to income (less debt for those with higher income) in the historical data from which a model was derived. But changing interest rates, for example, may turn that relationship positive, so that those with more income are comfortable with taking on more debt. This is a simplified example, of course, and in reality it is nearly impossible to detect when such “concepts” (the relationships between variables) shift. But as a result, model accuracy will most likely decay.

Data drift

Population drift is one way that input data to a model can change and be different from the data used to build a model. Perhaps more common are changes to upstream data—how the data is collected, encoded, scaled, etc.—that are not compatible with how the model was built. As a simple example, suppose the measurement of *income* changes in the example model for credit default risk. Instead of income measured per month, it is now measured, encoded, and sent to the model for prediction as annual income. Obviously, the resulting model prediction would be wrong, and likely very wrong.

Flexible Monitoring of End-to-End Scoring Pipelines and Agile Deployment

There are many reasons why model accuracy will decay over time; the question is how to architect for this fact of a model’s life.

Managing end-to-end scoring pipelines

As described earlier, scoring pipelines for prediction models used to enhance organizations’ critical (“production”) processes consist of multiple steps, starting with inputs, numerous data preprocessing and transformation steps, actual model scoring, postscoring decision rules and transformations, and final outputs. Every one of these steps must be governed and monitored. Data inputs should be managed as controlled artifacts (e.g., virtual views), tested, and approved for real-world production use. Rules should be inserted into the scoring pipelines to verify value ranges, reasonable input combinations, etc. Rules can also be inserted downstream of the data preparation and model scoring steps to validate reasonable values and relationships between values. Finally, the output of the scoring pipeline itself should be a managed artifact; for example, an output channel for the final predictions validated and approved for production use. Sometimes, dedicated model scoring pipelines must be deployed only to prepare and validate the data to be used as inputs into downstream model scoring pipelines.

Flexible monitoring

Given the threats to model accuracy and usefulness, and the risks associated with inaccurate or unjustified data, or model predictions that can result in unintended consequences, flexible monitoring of all aspects of an end-to-end scoring process is a must. Again, by building pipelines of reusable data-processing steps, this task becomes much easier. For example, a regular dedicated analytic scoring pipeline could create regular reports of "data health" for data inputs ("channels") that feed multiple predictive model scoring pipelines. That way, data can be validated and changes can be detected—for example, when data is encoded differently than before or when shifts in population statistics are observed. Also, if scoring pipelines are made up of reusable processing steps, then remedial steps can be performed much more efficiently and often automatically to all models in the dependency trees (that depend on data that changes).

The Bane of AI/ML at Scale: Big Code

It should be obvious from this discussion that the worst way of building and managing models is as big "blobs" of programming code. Too often, a team of data scientists may deliver a voluminous set of functions and subroutines authored in a big Python notebook, for example. The Python code may intertwine data preparation, validation, model scoring, and decision rules—all in one big document—that is then handed to IT for implementation. It is still not uncommon for some organizations that follow this approach to require over a year to implement, for example, an insurance fraud model. Obviously, the creativity of fraudsters is less encumbered by "process," and the resulting fraud models in productions are not very effective.

When models are created and deployed as large "chunks" rather than made up of small reusable steps, agile management, unit testing, and debugging of such models become near impossible.

Governed Inputs/Outputs; Small Reusable Steps; and Efficient, Targeted Monitoring That Enables Automation

To summarize, an effective ModelOps system must be able to manage and govern not only scoring models, but entire scoring pipelines, as well as the individual steps of those pipelines. This means that data inputs must be managed (e.g., as validated/approved virtual views to actual data), reusable, and often rule-based operating steps must be defined to validate data and protect against various causes of model drift and degradation, and output channels and methods must be managed and validated. In short, the entire scoring pipeline must be managed by ModelOps. Furthermore, by building, validating, and reusing common operational steps used in multiple scoring pipelines, organizations can build out their differentiating IP.

Future proofing and nurturing creativity

The individuals and roles primarily responsible for authoring scoring models—such as business analysts, stakeholders, or data scientists—also greatly benefit from this approach: ModelOps should be model-authoring-format agnostic, and not constrain the creativity of those building models. However, by managing data inputs/outputs and data preparation steps through ModelOps, their task is greatly simplified by being able to access data “approved and validated for modeling,” and efficiency is greatly enhanced because new models will seamlessly fit into the bigger system of managed data inputs/outputs and data processing steps.

Enabling automation

By managing reusable steps contributing to one or more scoring pipelines, many of the tasks required for maintaining scoring pipelines can also be automated. For example, suppose model drift is observed in a model for predicting credit default risk. Instead of starting a new “project” assigned to a team of data scientists to retrain or rebuilt the entire scoring pipeline, only the specific model scoring step can be retrained. For example, only a specific regression model can be automatically retrained to refine the parameters of the linear function that predicts risk from inputs prepared by upstream data preparation steps.

6. The World Is Changing Fast: Think Agile Model Deployment and Streaming Data

Too often, “traditional” data science and AI/ML look at the analytic process as (1) learning from past/historical data (“training”), and (2) deploying those models to new data going forward. But the information extracted from historical data may have a very short shelf life because of fast-changing business conditions, competitive environments, changing preferences and propensities, fashions, and fads. As discussed in the previous section, an efficient agile model life cycle to move models to production is essential and must involve the provisioning and management of data sources (inputs) and sinks (outputs)—and all the steps in between. Also remember that the values of inputs and their relationships can change very fast; therefore, effective management of data-in-motion (low-latency streaming data) can be the key factor driving the value of AI/ML.

Batch and Real-Time Data—Use Cases

The majority of the discussion of AI/ML and ModelOps in general is still mostly limited to use cases for batch data. The traditional (e.g., CRISP) model for AI/ML described earlier defines a cycle of understanding and learning from historical data, and then applying those insights to score new batches of data. For example, a bank may score all accounts nightly to update the predicted risk of credit default, a retailer may regularly score all customers for their propensity to respond to certain solicitations and offers, and a manufacturer may submit all data collected daily from machines and sensors to detect wear and tear and predict maintenance needs.

Addressing real-time scoring requirements

Of course, AI/ML models derived from historical data sets (batches) can be applied to real-time scoring use cases. For example, a credit default risk model can be called from an application for accepting credit applications online and in real time. The critical metric in that case is the maximum acceptable latency between the request for scoring (to a scoring service) and the receipt of results, including risk scores, allowable credit limits, acceptance/denial of credit, and auxiliary information such as reasons (so-called *reason codes*) if the credit application is rejected. Obviously, low-latency (fast) scoring is important in this case, ideally in the lower 100's of milliseconds per request; other applications require scoring latencies in the low 10's of milliseconds, or single milliseconds. The latter is required to support intelligent, advanced automation solutions.

Many organizations will build separate services to support low-latency scoring applications. However, this adds complexity that may not be necessary. Given the right ModelOps solution and architecture, low-latency scoring services can be supported directly by the ModelOps servers and services, without the need to build out a separate application or service.

Responding to a Dynamically Changing World

But things get more complicated when conditions and relationships change quickly between inputs and outputs that are to be predicted or anticipated. Previous sections discussed model drift, and the need to monitor the accuracy of models, which will inevitably decay over time. In many applications, model drift can happen very quickly. Technically, many processes in many domains are (increasingly) dynamically unstable; i.e., the relationships between variables, concepts, and process inputs and outputs can change very quickly and in novel ways.

For example, the recent changes due to the COVID-19 pandemic in 2020 and 2021 with respect to work, consumer behavior, travel, and virtually all aspects of life happened quickly, were unprecedented, and mostly were unexpected. But changes to consumer preferences, fashions, and fads happen all the time. In automated and process manufacturing, sometimes thousands of automated processing steps and measurements interact to create complex and dynamically unstable processes where new quality problems can be impossible to anticipate from historical data that is more than a few weeks old (or less).

Effectively, platforms for building and deploying models must be able to adapt quickly, and sometimes learn directly and in real time from the data streams generated by some process. Industry analysts at Forrester®[1] acknowledged the value and requirements for real-time analytics in the recent Streaming Analytics Wave report (Gualtieri 2021). While this is an area of ongoing development and research (see, for example, the brief application overview at Hill 2021), the need to adjust quickly to new conditions and realities across various domains and industries is making support for dynamic learning from real-time data an important requirement for ModelOps tools.

Figure 2-5 provides an example use case of how dynamically adjusting models can outperform static models built from historical data samples when applied to an ongoing best-next action use case. The diagram on the top shows the data flow of the incoming real-time, best-next action data (for example, collected at the point of sale).

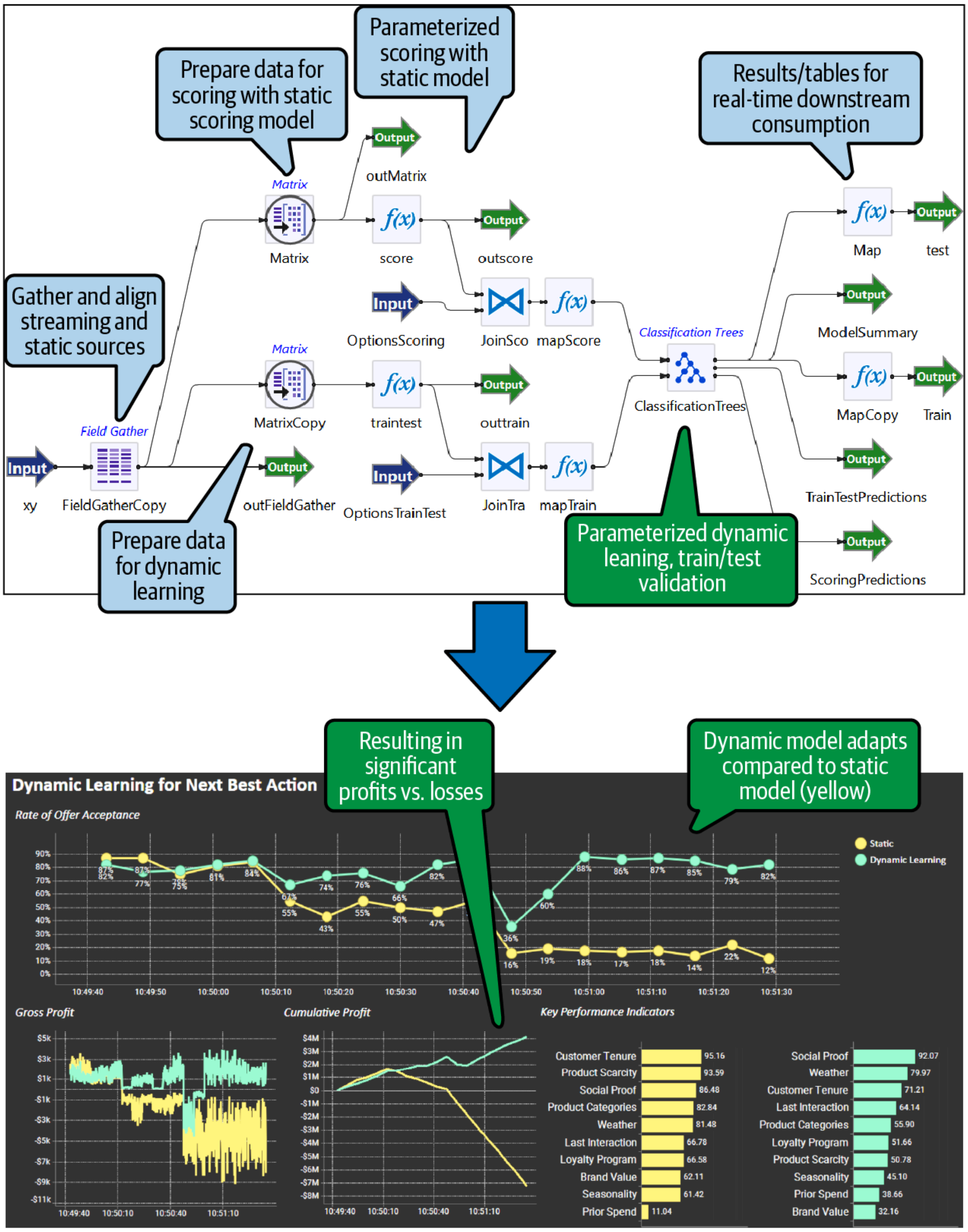


Figure 2-5. Dynamic/adaptive learning example via TIBCO® Streaming

After some data preparation, the “Parameterized dynamic learning, train/test validation” box (bottom of the top diagram) points to the step where a prediction is made from the most recent (dynamically) updated model.

The effectiveness of the static and dynamic model is monitored via the real-time dashboard shown in the bottom diagram of Figure 2-5; the green line and bars track the accuracy and value of the dynamic model, while the yellow line and bars track the accuracy and value of the static model. In short, about two-thirds through the displayed time range, both line charts at the top—indicating accuracy over the most recent predictions—point downward. But while the static model (the yellow line) shows continued decreased accuracy and effectiveness, the dynamic model automatically adapts and “recovers” to the previously observed accuracy and effectiveness. Other charts (bottom center of the bottom diagram) indicate the cumulative profitability attributable to the dynamically updating model over time.

This simple example provides a good demonstration of the value of agility and responsiveness to changing market or other conditions affecting model accuracies, and the value of inserting continuous model-building/calibration methods directly into the data stream. In general, efficient support for scoring and analytics against low-latency, real-time streaming data is important now, will become more important in the future, and will be a must-have for organizations aiming to differentiate their AI/ML-infused capabilities and processes from competitors.

7. Plan for Success and Monitor for Value (Not Just Accuracy)

The effective monitoring of models and their predictions is absolutely essential for responsive/agile management of deployed AI/ML-based models. But the monitoring must go beyond just simple accuracy, model drift, or other aspects of the data or models. Effective model monitoring must be aligned with business goals, KPIs, risks, and so on. It is important to think through what “success” and “failure” means for the organization with respect to the process that is informed by an AI/ML-based model. For most applications and use cases, success/failure is not defined by a single number but is multidimensional. Being able to surface model effectiveness through BI or reporting tools that reflect all aspects of model cost, value, and risks to the right stakeholders is critically important.

Every Project Starts at the End—or It Is Likely to Fail!

Most practitioners have learned that it is important to define clearly and up front when the project is done, how to know that it is done, and how to know that it is a win. Aligning the measurements of "success" with clearly identified goals is a must. While this seems obvious, it is not easy. A frequently cited statistic originally reported by Gartner is that 85% of all big-data "data science" projects fail (e.g., see O'Neill 2019); Venturebeat (2019) estimated that 87% of data science projects never make it to production. According to White (2019), only 20% of analytic insights actually deliver real business value. There are obviously a number of reasons for lack of measurable success. One important reason is that AI/ML projects can easily take on "lives of their own," focusing on technical details, predictive accuracy, and just being "cool." Success requires planning, alignment with clear goals, and the right tools and technologies.

Planning for How Model Scoring Is to Be Consumed—and Valuable

Results of model scoring can be consumed in many ways, and the tooling must accommodate all of them. It is always a good idea to ask what an "ideal result" and outcome would look like. For example, suppose the model can make 100% accurate predictions! How is that translated into value? And how is *value* measured?

If a process (e.g., for initial screening of applicants for credit) is 100% successful, is that screening to be automated and performed in real time as part of a customer-facing web interface? If so, results must be delivered via APIs. Perhaps the model will be used to allow the organization to offer more competitive interest rates, specifically and accurately tailored to each applicant. Such a model may be consumed in a batch process scoring all existing customers and prospects. Scoring pipelines may also be created to compute reports for regulatory compliance, with full audit trails.

ModelOps tooling and processes must support all modes of delivery of results and manage the scoring pipelines end to end. ModelOps must support and manage flexible reporting, visualizations, interfaces for API and web services, etc. Without that, the result of model scoring pipelines are just "numbers," which themselves have no value.

8. The Future of AI Will Be Regulated—ModelOps Helps Ensure Compliance

To reiterate, regulatory frameworks for AI/ML models are being considered or already in effect around the world (see, for example, Burt 2021). Not being able to demonstrate compliance through documented model approvals, audit trails, version control, etc., can be extremely costly and hazardous to your organization. It is a significant risk inherent in the application of AI/ML, as discussed in a previous point.

The previous points have discussed many aspects, necessary tools, and process capabilities for ModelOps that must be considered in order to support compliance with current and future regulatory frameworks.

Requirements for Analytics in Medical Device and Pharma Manufacturing Offer Guidance for What's to Come

A good guide of what's to come with respect to regulations for AI/ML are the best practices that must be implemented in FDA-regulated manufacturing environments. A complete review of those practices for good manufacturing practice (GmP), good documentation practices (GDocP), and generally good "anything" practices (GxP) is out of scope for this brief review (see, for example, US Department of Health and Human Services 2011, 2021). In most general terms, manufacturers in this industry must demonstrate process understanding and risks, backed up by validated analytics (scoring pipelines). Some of the key components of validated analytics solutions are:

Detailed requirements, traceable through design and test plans

This is very similar to best practices in software manufacturing (see also the next bullet point), and really the formal way to state the main point:

requirements for analytics (to demonstrate process understanding, process

verified/validated. In practice, that means that detailed documentation is created to describe which specific computations are to be performed and how, and how they will be tested. For example, for so-called continued process verification (CPV), computational approaches for quality control charting (a key analytic tool or "model") must be documented and the implementation validated. As described in the next section, the manner in which vendors/creators of ModelOps and analytic tooling are verifying *their* best practices and high quality is extremely important, because it simplifies the validation and compliance tasks for the users of the tools.

Approval processes, electronic signatures

Once analytics scoring pipelines are created, they must be approved before they can be used to support a validated (GxP) production process. "Approval" here means that the designated role of "approver" will sign (electronically) the respective documentation. Various rules apply to ensure that the records and their approval cannot be faked in any way. For example, the FDA requires that approvers cannot be the same individuals or part of the group of individuals who create the scoring pipelines ("it requires four eyeballs to commit fraud…").

Version control, audit logs

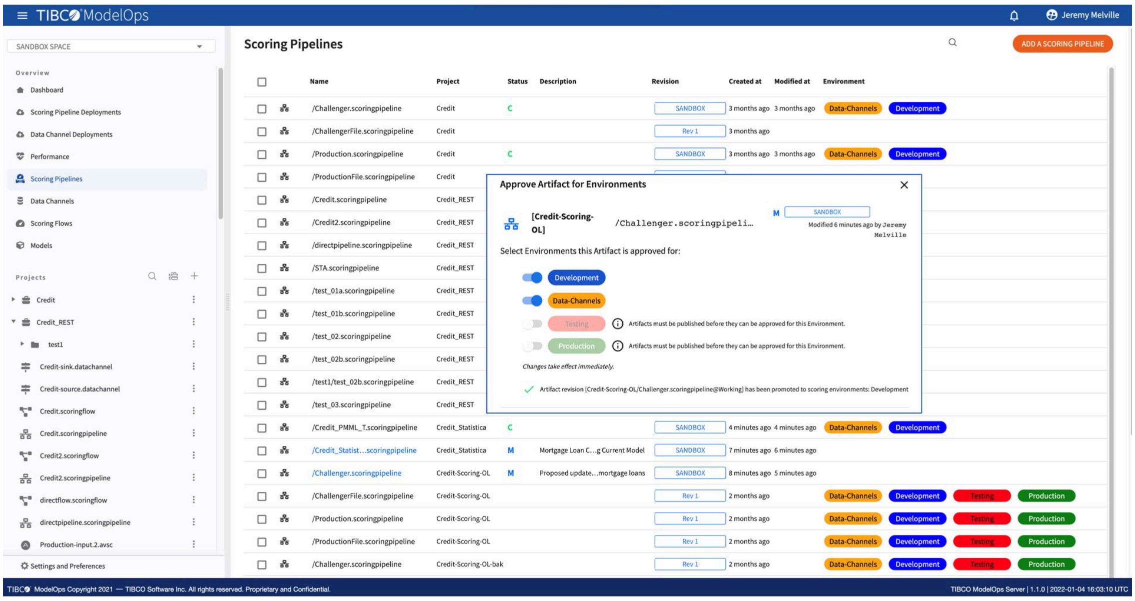
Finally, once all analytic scoring pipelines have been created, documented, tested, and approved, they must be locked down. That is, the ModelOps software must ensure that the records documenting best practices (GxP) cannot be altered. In practice, ModelOps tools must implement granular roles, version control, and audit logs to demonstrate to auditors that the required processes were followed and not compromised or altered.

ModelOps Capabilities, Beyond Efficient Deployment and Scoring

Any analytics that are people facing—be it to support the allocation of credit commensurate with risk, decisions to hire, promote or fire, or just to provide a "special discount" or coupon—will dis/advantage the persons predicted in some way. And those persons will demand that transparent, justified, verifiable, and fair best practices were followed to make the predictions.

To reiterate, the future of AI/ML applications predicting people will be regulated and shaped by best practices that can be verified by auditors. All AI/ML applications in manufacturing will also be regulated—perhaps voluntarily—because ultimately any supplier of manufactured goods will need to demonstrate to their customers that best practices were followed, and risk (of low quality, defects) is minimized.

Figure 2-6 provides an example screen from TIBCO® ModelOps, summarizing the deployment environments, approvals, and other metadata applicable to the scoring pipelines managing through the platform. In addition, current and emerging regulatory environments guiding the application of AI and machine learning require that the ModelOps platform support version control, audit logs, granular role-based access (e.g., of "approvers"), and flexible analytics specific to different domains to demonstrate process understanding and explainability, computational validity, and accuracy. These capabilities of ModelOps are critical components for bringing AI/ML or any analytics to production environments.



Scoring Pipelines

☐	Name	Project	Status	Description	Revision	Created at	Modified at	Environment
☐	/Challenger.scoringpipeline	Credit	C		SANDBOX	3 months ago	3 months ago	Data-Channels Development
☐	/ChallengerFile.scoringpipeline	Credit			Rev 1	3 months ago		
☐	/Production.scoringpipeline	Credit	C		SANDBOX	3 months ago	3 months ago	Data-Channels Development
☐	/ProductionFile.scoringpipeline	Credit						
☐	/Credit.scoringpipeline	Credit_REST						
☐	/Credit2.scoringpipeline	Credit_REST						
☐	/directpipeline.scoringpipeline	Credit_REST						
☐	/STA.scoringpipeline	Credit_REST						
☐	/test_01a.scoringpipeline	Credit_REST						
☐	/test_01b.scoringpipeline	Credit_REST						
☐	/test_02.scoringpipeline	Credit_REST						
☐	/test_02b.scoringpipeline	Credit_REST						
☐	/test1/test_02b.scoringpipeline	Credit_REST						
☐	/test_03.scoringpipeline	Credit_REST						
☐	/Credit_PMML_T.scoringpipeline	Credit_Statistica	C		SANDBOX	4 minutes ago	4 minutes ago	Data-Channels Development
☐	/Credit_Statist…scoringpipeline	Credit_Statistica	M	Mortgage Loan C…g Current Model	SANDBOX	7 minutes ago	6 minutes ago	
☐	/Challenger.scoringpipeline	Credit-Scoring-OL	M	Proposed update…mortgage loans	SANDBOX	8 minutes ago	5 minutes ago	
☐	/ChallengerFile.scoringpipeline	Credit-Scoring-OL			Rev 1	2 months ago		Data-Channels Development Testing Production
☐	/Production.scoringpipeline	Credit-Scoring-OL			Rev 1	2 months ago		Data-Channels Development Testing Production
☐	/ProductionFile.scoringpipeline	Credit-Scoring-OL			Rev 1	2 months ago		Data-Channels Development Testing Production
☐	/Challenger.scoringpipeline	Credit-Scoring-OL-bak			Rev 1	2 months ago		Data-Channels Development Testing Production

Approve Artifact for Environments ✕

[Credit-Scoring-OL] /Challenger.scoringpipeli… M SANDBOX
Modified 6 minutes ago by Jeremy Melville

Select Environments this Artifact is approved for:

- ● Development
- ● Data-Channels
- ○ Testing ⓘ Artifacts must be published before they can be approved for this Environment.
- ○ Production ⓘ Artifacts must be published before they can be approved for this Environment.

Changes take effect immediately.

✓ Artifact revision [Credit-Scoring-OL/Challenger.scoringpipeline@Working] has been promoted to scoring environments: Development

TIBCO ModelOps Copyright 2021 — TIBCO Software Inc. All rights reserved. Proprietary and Confidential.

TIBCO ModelOps Server | 1.1.0 | 2022-01-04 16:03:10 UTC

Figure 2-6. Example summary screen of approvals, environments for scoring pipelines, and status in TIBCO® ModelOps

9. Expect Quality and Process Best Practices from the ModelOps Operating System

There is another implication to the previous point: increasingly, all consumers of any software—including software for ModelOps—will demand that best practices were followed in the creation of that software. How can a user of a ModelOps solution ever achieve compliance, efficiency, and value if the *suppliers* of the software themselves do not follow best practices when making their software? If tools for managing models are themselves "no good," how can the process for managing models (and compliance, efficiency, value) be successful, reliable, and compliant?

Obviously, it is equally important that the tools used to manage AI/ML models are of high quality and built consistent with best practices. There is a reason why industry certifications such as ISO9001, ISO/IEC 27001, SOC 2,[2] PCI DSS AOC, HITRUST, and others were established and are often required for ModelOps software. Software vendor audits are already common and required for such software before it can be used in pharmaceutical or medical device manufacturing, or in financial services. The deployment of AI/ML models will require the same diligence, and quality certifications of ModelOps software are and will become increasingly important.

10. ModelOps Is a Required Tool for Managing AI/ML Model Deployment: Start Small, but Start Now

Finally, an obvious point: AI/ML has transformed many industries; it will continue to do so, and it is hard to see how a modern organization or business can survive without it. The focus with respect to AI/ML and analytics has definitely shifted away from computational algorithms and methods to considerations of best practices, value, risk, and ROI. Modern cloud-based architectures have also made it possible to scale up (or down) quickly and efficiently for the applications of scoring pipelines, automation solutions, scoring services, analytic reports or intelligent BI solutions, or AI/ML-infused applications.

ModelOps and its various facets must even be part of small first steps, because otherwise analytic silos, "gurus" (critical individual resources who "have the knowledge"), competing interests, disconnected data science teams, and any number of issues will quickly fester. There is significant experience available from experts and expert vendors to help with the first steps, augment existing processes, fix missteps, and ultimately be successful with AI/ML.

[1] Forrester is a registered trademark of Forrester Research, Inc.

[2] The American Institute of CPAs (AICPA) developed SOC.