Accenture Labs

Delivering Improved Insights with Automated Analytics

Model management framework simplifies model deployment at scale

High performance. Delivered.

accenture

Strategy | Consulting | Digital | Technology | Operations

With the growth of Internet of Things (IoT), big data and artificial intelligence, organizations are actively building analytics platforms to discover insights and predict outcomes to deliver improved business results. Analytical models are at the heart of these platforms, applying machine learning techniques on data at massive scale in real time to generate useful insights. As companies in industries such as oil and gas, utilities, financial services, healthcare and life sciences adopt these platforms, the need to manage and especially test a wide range of models will increase.

In order to find the best model, data scientists must quickly yet thoroughly test hundreds and even thousands of hypotheses on a given dataset and compare models in a brute force manner. Models must be trained, compared and monitored before deploying into production, requiring many steps to take place in order to operationalize a model's lifecycle. Furthermore, building different types of models requires system administration skills in a variety of analytics engines.

Accenture Labs has developed a model management framework that automates, simplifies and accelerates the lifecycle management of models at scale—from build, train, test and deploy all the way to monitor, retrain and redeploy or retire.

The framework is also agnostic to the analytics engines that are used to build models. This approach doubles as an elegant solution for sharing models and business knowledge between data scientists, domain experts and business analysts.

While existing vendor tools provide some capabilities for managing models and deployment, the framework created at Accenture Labs provides a complete set of abstractions in the modeling process that enables a catalog approach to discover the best models for analytics. By helping to simplify the process to identify and deploy the most predictive model on data, business insights are delivered faster and more accurately.

Traditional vs. hybrid architectures: enabling real-time analytics for new use cases

Given the unprecedented amount of data available from mobile devices, social media and increasingly the IoT, organizations are turning to big data platforms to collect, process and analyze data.

These use cases require a big data architecture stack to process data at a massive scale. In addition, it needs to handle real-time data, which is defined as data that is processed in sub-seconds or seconds from the time it arrives to the system to when the results are derived.

The availability of up-to-date data allows for use cases that were not possible in a purely batch-oriented data pipeline. For example, a utilities company can build analytical models that analyze weather data to predict a severe weather event and move their staff onsite to counteract against outages. Similarly, an oil and gas company can use sensor data from oil wells to predict when a pump will fail in order to proactively fix the pump, allowing for cost savings and increased revenue.

In order to architect a system for both batch and real-time analytics, a hybrid architecture such as the Lambda architecture can be employed. The Lambda architecture is a pattern made popular by Internet companies. Its name derives from the two legs of the Lambda character and comprises a batch layer that processes historical batch data, and a speed layer that processes data quickly in sub-seconds to seconds. As a result, organizations can perform real-time analytics on data that combines up-to-date data with historical data views.

We focus on addressing model lifecycle management in such an architecture that balances batch and real-time needs.

Challenges with modeling in current state

Even though a real-time data analytics system can deliver significant benefits, it still presents issues in the lifecycle management and modeling process as there is no easy way to monitor, retrain and redeploy the models. In general, data scientists collect the data they are interested in, prepare and stage the data, apply different machine learning techniques to find a best-of-class model, and continually tweak the parameters of the algorithm to refine the outcomes.

Automating and operationalizing this process is difficult. For example, on a Lambda architecture, the data scientist must code the model, select parameters and a runtime environment, train the model on batch data, and monitor the process to troubleshoot errors that might occur. This process is repeated iteratively on different parameters and machine learning algorithms, and after comparing the models on accuracy and performance, the model can be deployed in the speed architecture.

Additional challenges that organizations may face with manual modeling include a lack of:

Resource expertise. Analytical modeling in a real-time, streaming architecture requires individuals with expertise in both data science and engineering. A data scientist codes a model relevant to the use case, recognizing patterns in the data, and selecting or writing a machine learning algorithm for the model. A skilled system administrator operates the runtime environments for the model, which requires an understanding of big data systems and architectures. Finding individuals with competency in both data science and engineering is a challenge.

Runtime abstraction. The need for skills in systems engineering for analytical modeling outlines another problem: the lack of runtime abstraction. Data scientists are generally comfortable with machine learning and statistical computing programming languages such as R. However, they may not be trained in systems engineering to troubleshoot and operate a runtime environment.

Given the lack of abstraction, the process for training and deploying a model involves manual steps, requiring the data scientist to remotely connect to the runtime environment and execute terminal commands to launch models and check on the status. The data scientist repeats this process many times, tweaking the parameters of the algorithm to train many different models. As a result, the data scientist loses productivity because they spend more time on system administration—operating the runtime environment—and less time on training accurate models.

Model support. When the number of model types and runtime environments increases, an abstraction for the environments becomes critical. Runtime environments cannot support all types of models. As a simple example, a model built with R libraries will execute in an R environment, but it may not successfully run in a Spark environment if there are missing R dependencies in the Spark libraries. So when the data scientist wants to train a model that runs in a new runtime, he must learn to operate that environment before being able to train the model.

Central repository. Currently, there is no standard method for comparing, sharing or viewing models created by other data scientists, which results in siloed analytics work. Without a way to view models created by others, data scientists leverage their own private library of machine learning algorithms and datasets for their use cases.

As models are built and trained by many data scientists, the same algorithms may be used to build similar models, particularly if a certain set of algorithms is common for an organization's use cases. Over time, models begin to sprawl and duplicate unnecessarily, making it more difficult to establish a centralized library.

Clearly, a manual modeling mechanism is too inflexible and not scalable for an organization's data scientists. In light of these challenges, there is an opportunity to improve model management. Current vendors and research projects in the space sometimes provide management of models, but they generally do not provide enough abstractions, automation or support for real-time deployment of models. For example, some tools provide abstractions to train models in analytics environments such as Hadoop or Spark, but do not provide enough extensibility to support additional runtime environments. Existing tools often lack the ability to deploy models against a live stream for real-time results, which is a necessary capability for end-to-end model management.

Model management framework simplifies approach

Accenture Labs has developed a flexible and extensible model management framework to help simplify the interfaces for analytical modeling at scale on a Lambda architecture. In this framework, data scientists can more easily train and deploy analytical models in various runtime environments. The framework abstracts the data engineering throughout the lifecycle management and modeling process, reduces the complexity of training and deployment, and shares models in a way that is consumable. Our reference architecture for this framework proposes the following:

Real Time Analytics

Runtime Environments

- Distributed Computing
- Scientific Computing

Model Management

- Runtime Verifier
- Deployment and Scheduler
- Monitoring Service
- Model Store
- Trained Model Store
- Metadata Store
- API
- Templated Model Interfaces

Users

- Data Scientists
- Business Analysts

Runtime verifier: Ability to determine which runtime environments can support a model prior to execution, helping to enable faster iterations for model training.

Deployment and scheduler: Automatic training, deployment and scheduling of models on runtime environments, obviating the need to operate the runtime environments during the modeling process.

Monitoring service and model metadata store: A service that monitors the status of the model during its execution on the runtime environment for the user, as well as any metadata about its execution that is then stored for the user.

Model store and trained model store: A library of models of commonly used machine learning algorithms that can be trained on historical datasets, as well as trained models for deployment against a live stream for real-time results.

Application programming interfaces (API): Exposes functionalities with API endpoints for users to verify, train, deploy and monitor models on runtime environments.

Templated model interfaces: Interfaces that abstract the complexity of machine learning algorithms, allowing users to specify the inputs and outputs for the model.

Using this framework, lifecycle management and modeling is a simpler five-step process:

1 A data scientist codes and builds a model for the use case and uploads it to the model management framework. Alternatively, she views a catalog of models that were uploaded by other data scientists and leverages existing algorithms.

2 After selecting the model, the data scientist specifies the input historical batch data, the algorithm parameters, as well as the input and target features for the model. The framework verifies the model against the available runtime environments and informs her which runtime environments support the model.

3 After selecting the runtime environment, the model management framework connects to the runtime environment and trains the model on the historical dataset for the data scientist, abstracting away the complexity of operating the runtime environments. The framework also monitors the status of the model as it is trained or deployed, and it reports the metrics on how that model performed on the training dataset.

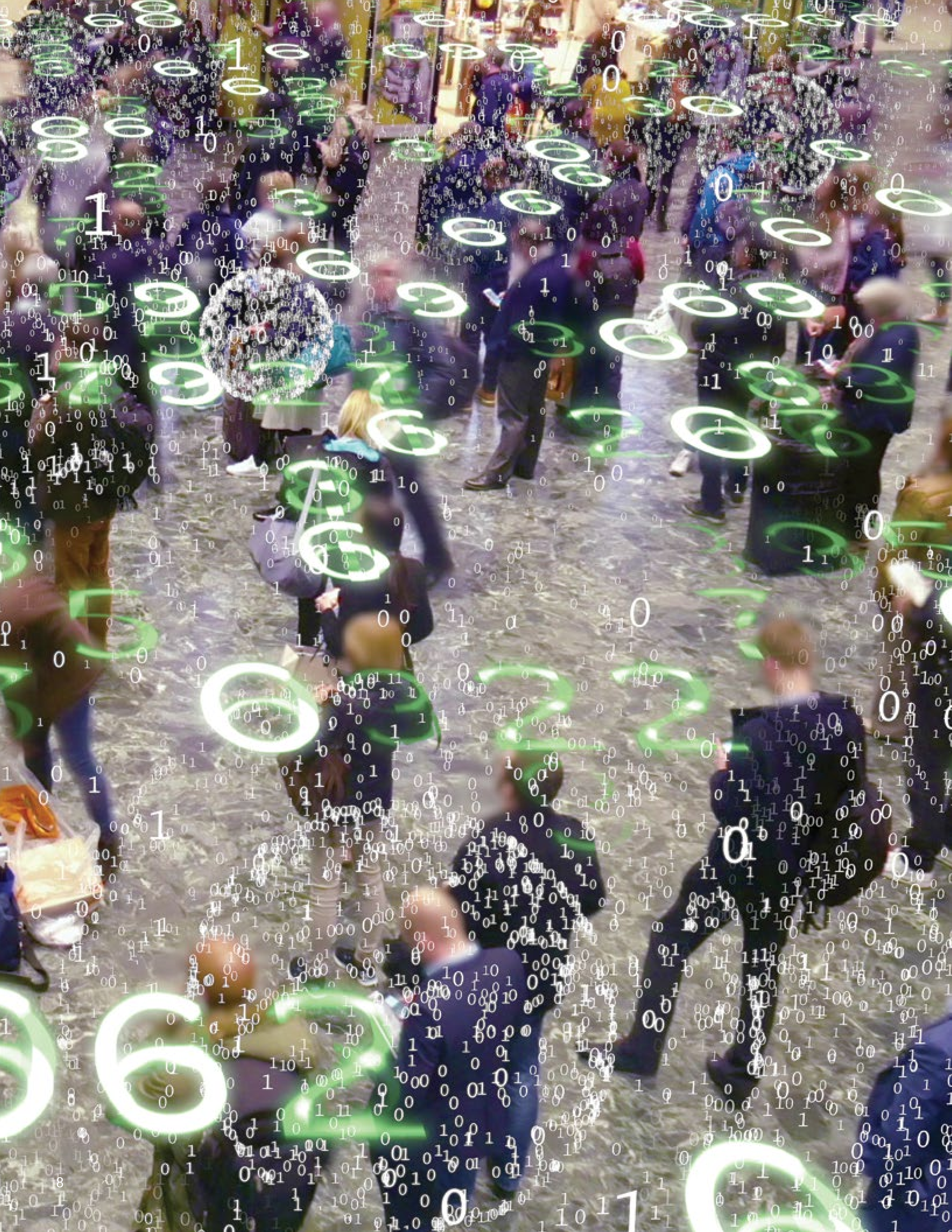
4 The data scientist can iteratively repeat this process to train models on varying input parameters, and then compare these models visually in a champion-challenger view to select the best model.

5 Finally, the data scientist promotes the best model to a live stream for real-time results.

There are many benefits to using Accenture Labs' model management framework over other tools or a manual approach. First, the framework obviates the need for a data scientist to manually operate the runtime environment because it will train, monitor and deploy the model for the user. In this way, the data scientist does not have to connect to the runtime directly, open a terminal and execute manual commands to operate the runtime environment. The framework can also extend this support to additional environments as needed, which means that it is no longer cumbersome to support new models as the number of models grows.

Second, a centralized model view allows for models to be compared in a champion-challenger fashion to determine which model performed the best in accuracy or performance on the historical dataset. The data scientist can then promote this model to a live stream for real-time results, using the same engineering abstractions provided in the training process.

Finally, the automatic ability to train and deploy the models, as well as the various model views available, means the modeling process is no longer limited to specialized individuals with data science and data engineering skills. This empowers other stakeholders such as business analysts to try model deployment, and it makes it easier for them to extract insights from the data by leveraging models built by data scientists.



Putting the model management framework to the test

In a proof of concept, Accenture Labs implemented the model management framework with the core functionalities exposed via APIs, as well as a front-end interface that wraps the API calls for end users.

To start, the team uploaded example model templates for machine learning algorithms such as linear regression, logistic regression, decision trees, random forest and time series forecasting.

Data scientists or business analysts can select a dataset in Amazon S3 for training, and our prototype can train models in runtime environments such as Spark on Hadoop YARN and Spark in Amazon's cloud ElasticMapReduce service, abstracting away the interfaces to the runtime environments.

The model manager can also deploy trained models to a real-time stream, and our team tested this against Kafka and Amazon Kinesis. Additionally, we packaged the model manager into containers using Docker and successfully deployed the model manager in on-premise environments and in the cloud.

Conclusion

Accenture Labs' model lifecycle management framework uses advanced machine learning technology to provide a set of comprehensive abstractions for analytical modeling—one that is both easier to use and quicker to produce results. The framework is unique from existing tools because of its lifecycle management of models and its agnostic approach to runtime environments. It can be extended to support additional environments, which increases the number of models that can be leveraged in the data pipeline. The abstractions provided for the runtime environments, the champion-challenger view and the shared library of models both operationalizes and democratizes access to analytics.

Using this framework, an organization can create an ecosystem for harvesting analytical models, providing an application store-like experience for data scientists and business analysts to discover the best models and promote them for use. As companies rely more heavily on real-time big data analytics in the digital age, the ability to manage, train, deploy and share models that turn analytics into action-oriented outcomes is essential.

Contact us

Jean-Luc Chatelain
CTO Accenture Analytics
jean-luc.chatelain@accenture.com

Andrew Musselman
Chief Data Scientist
andrew.musselman@accenture.com

William Gatehouse
Managing Director, Accenture Digital
william.r.gatehouse@accenture.com

Contributors

Teresa Tung
Technology Labs Fellow
teresa.tung@accenture.com

Chris Kang
Associate Principal
chris.kang@accenture.com

Carl Dukatz
R&D Senior Manager
carl.m.dukatz@accenture.com

About Accenture

Accenture is a leading global professional services company, providing a broad range of services and solutions in strategy, consulting, digital, technology and operations. Combining unmatched experience and specialized skills across more than 40 industries and all business functions—underpinned by the world's largest delivery network—Accenture works at the intersection of business and technology to help clients improve their performance and create sustainable value for their stakeholders. With approximately 373,000 people serving clients in more than 120 countries, Accenture drives innovation to improve the way the world works and lives. Visit us at www.accenture.com.

About Accenture Labs

Accenture Labs invents the future for Accenture, our clients and the market. Focused on solving critical business problems with advanced technology, Accenture Labs brings fresh insights and innovations to our clients, helping them capitalize on dramatic changes in technology, business and society. Our dedicated team of technologists and researchers work with leaders across the company to invest in, incubate and deliver breakthrough ideas and solutions that help our clients create new sources of business advantage.

Accenture Labs is located in six key research hubs around the world: Silicon Valley, CA; Sophia Antipolis, France; Arlington, Virginia; Beijing, China; Bangalore, India, and Dublin, Ireland. The Labs collaborates extensively with Accenture's network of nearly 400 innovation centers, studios and centers of excellence located in 92 cities and 35 countries globally to deliver cutting-edge research, insights and solutions to clients where they operate and live.

For more information, please visit www.accenture.com/labs.

Copyright © 2016 Accenture
All rights reserved.

Accenture, its logo, and
High Performance Delivered
are trademarks of Accenture.

This document makes descriptive reference to trademarks that may be owned by others. The use of such trademarks herein is not an assertion of ownership of such trademarks by Accenture and is not intended to represent or imply the existence of an association between Accenture and the lawful owners of such trademarks.

16-2129U/12-0680