Getting AI/ML and DevOps working better together

[f] (https://www.facebook.com/share.php?
u=https%3A%2F%2Fazure.microsoft.com%2Fblog%2Fgetting-ai-ml-and-devops-working-better-
together%2F) [twitter] (https://twitter.com/share?
url=https%3A%2F%2Fazure.microsoft.com%2Fblog%2Fgetting-ai-ml-and-devops-working-better-
together%2F&text=Getting+AI%2FML+and+DevOps+working+better+together) [in]
(https://www.linkedin.com/shareArticle?
mini=true&url=https%3A%2F%2Fazure.microsoft.com%2Fblog%2Fgetting-ai-ml-and-devops-working-
better-together%2F)

Posted on 20 September, 2018

Kate Baroni
Software Architect, Microsoft Azure

Artificial Intelligence (AI) and machine learning (ML) technologies extend the capabilities of software
applications that are now found throughout our daily life: digital assistants, facial recognition, photo
captioning, banking services, and product recommendations. The difficult part about integrating AI or
ML into an application is not the technology, or the math, or the science or the algorithms. The
challenge is getting the model deployed into a production environment and keeping it operational
and supportable. Software development teams know how to deliver business applications and cloud
services. AI/ML teams know how to develop models that can transform a business. But when it comes
to putting the two together to implement an application pipeline specific to AI/ML — to automate it
and wrap it around good deployment practices — the process needs some effort to be successful.

The need for aligned development approaches

DevOps has become the de-facto development standard for cloud services. It places an emphasis on process, automation, and fosters a culture that encourages new ways of working together across teams. DevOps is an application-centric paradigm that focuses on the platform, instrumentation, and process to support applications: what is the infrastructure needed to support the application? What tools can be used to automate it? What is the release process for QA/production?

AI/ML projects have their own development methodologies including CRISP-DM (https://en.wikipedia.org/wiki/Cross-industry_standard_process_for_data_mining) and Microsoft Team Data Science Process (https://azure.microsoft.com/en-gb/documentation/learning-paths/data-science-process/) (TDSP). Like DevOps, these methodologies are grounded in principles and practices learned from real-world projects. AI/ML teams use an approach unique to data science projects where there are frequent, small iterations to refine the data features, the model, and the analytics question. It's a process intended to align a business problem with AI/ML model development. The release process is not a focus for CRISP-DM or TDSP and there is little interaction with an operations team. DevOps teams (today) are yet not familiar with the tools, languages, and artifacts of data science projects.

DevOps and AI/ML development are two independent methodologies with a common goal: to put an AI application into production. Today it takes the effort to bridge the gaps between the two approaches. AI/ML projects need to incorporate some of the operational and deployment practices that make DevOps effective and DevOps projects need to accommodate the AI/ML development process to automate the deployment and release process for AI/ML models.

Integrating AI/ML teams, process, and tools

Based on lessons learned from several Microsoft projects including the Mobile Bank Fraud Solution (https://download.microsoft.com/download/0/1/5/0150425C-14C7-41F4-97EA-3DE57B678C51/IndSG_FraudDetection.pdf), some suggestions for bridging the gap between DevOps

and AI/ML projects follow.

DevOps for AI/ML

DevOps for AI/ML has the potential to stabilize and streamline the model release process. It is often paired with the practice and toolset to support Continuous Integration/Continuous Deployment (CI/CD) (https://en.wikipedia.org/wiki/CI/CD). Here are some ways to consider CI/CD for AI/ML workstreams:

- The AI/ML process relies on experimentation and iteration of models and it can take hours or days for a model to train and test. Carve out a separate workflow to accommodate the timelines and artifacts for a model build and test cycle. Avoid gating time-sensitive application builds on AM/ML model builds.

- For AI/ML teams, think about models as having an expectation to deliver value over time rather than a one-time construction of the model. Adopt practices and processes that plan for and allow a model lifecycle and evolution.

- DevOps is often characterized as bringing together business, development, release, and operational expertise to deliver a solution. Ensure that AI/ML is represented on feature teams and is included throughout the design, development, and operational sessions.

Establish performance metrics and operational telemetry for AI/ML

Use metrics and telemetry to inform what models will be deployed and updated. Metrics can be standard performance measures like precision, recall, or F1 scores (https://en.wikipedia.org/wiki/Precision_and_recall). Or they can be scenario specific measures like the industry-standard fraud metrics developed to inform a fraud manager about a fraud model's performance. Here are some ways to integrate AI/ML metrics into an application solution:

- Define model accuracy metrics and track them through model training, validation, testing, and deployment.

- Define business metrics to capture the business impact of the model in operations. For an example see R notebook for fraud metrics (https://gist.github.com/jspoelstra/9577c07d3c9d1087072b87051abd3c59).

- Capture data metrics, like dataset sizes, volumes, update frequencies, distributions, categories, and data types. Model performance can change unexpectedly for many reasons and it's expedient to know if changes are due to data.

- Track operational telemetry about the model: how often is it called? By which applications or gateways? Are there problems? What are the accuracy and usage trends? How much compute or memory does the model consume?

- Create a model performance dashboard that tracks model versions, performance metrics, and data sets.

AI/ML models need to be updated periodically. Over time, and as new and different data becomes available — or customers or seasons or trends change — a model will need to be re-trained to continue to be effective. Use metrics and telemetry to help refine the update strategy and determine when a model needs to be re-trained.

Automate the end-to-end data and model pipeline

The AI/ML pipeline is an important concept because it connects the necessary tools, processes, and data elements to produce and operationalize an AI/ML model. It also introduces another dimension of complexity for a DevOps process. One of the foundational pillars of DevOps is automation, but automating an end-to-end data and model pipeline is a byzantine integration challenge.

Workstreams in an AI/ML pipeline are typically divided between different teams of experts where each step in the process can be very detailed and intricate. It may not be practical to automate across the entire pipeline because of the difference in requirements, tools, and languages. Identify the steps in the process that can be easily automated like the data transformation scripts, or data and model quality checks. Consider the following workstreams:

Workstream	Description	Automation
Data Analysis	Includes data acquisition and focusing on exploring, profiling, cleaning, and transforming. Also includes enriching, and staging data for modeling.	Develop scripts and tests to move and validate the data. Also create scripts to report on the data quality, changes, volume, and consistencies.
Experimentation	Includes feature engineering, model fitting, and model evaluation.	Develop scripts, tests, and documentation to reproduce the steps and capture model outputs and performance.
Release Process	Includes the process for deploying a model and data pipeline into production.	Integrate the AI/ML pipeline into the release process
Operationalization	Includes capturing operational and performance metrics.	Create operational instrumentation for the AI/ML pipeline. For subsequent model retraining cycles, capture and store model inputs, and outputs.
Model Re-training and Refinement	Determine a cadence for model re-training.	Instrument the AI/ML pipeline with alerts and notifications to trigger retraining.

Visualization	Develop an AI/ML dashboard to centralize information and metrics related to the model and data. Include accuracy, operational characteristics, business impact, history, and versions.	n/a

An automated end-to-end process for the AI/ML pipeline can accelerate development and drive reproducibility, consistency, and efficiency across AI/ML projects.

Versioning

Versioning is about keeping track of an application's artifacts and the changes to the artifacts.

In software development projects this includes code, scripts, documentation, and files. A similar practice is just as important for AI/ML projects because—typically—there are multiple components, each with separate release and versioning cycles. In AI/ML projects, the artifacts could include:

- Data: training data, inference data, data metrics, graphs, plots, data structures, schemas

- Models: trained models, scoring models, A/B testing models

- Model outputs: predictions, model metrics, business metrics

- Algorithms, code, notebooks

Versioning can help provide:

- Traceability for model changes from multiple collaborators

- Audit trails for project artifacts

- Information about which models are called from which applications

A practical example of the importance of versioning for the AI/ML team happens when the performance of a model changes unexpectedly, and the change has nothing to do with the model itself. The ability to easily trace back inputs, dependencies, model, and data set versions could save days or weeks of effort.

At a minimum, decide on a consistent naming convention and use it for the data files, folders, and AI/ML models. Several different teams will be involved in the modeling process and without naming conventions, there will be confusion over which data sets or model versions to use.

Consider container architectures

Container architectures have the potential to streamline and simplify model development, test, and deployment. And as a package-based interface, containers make it easy for software applications to connect. Containers create an abstraction layer between models and the underlying infrastructure. This lets the AI/ML team focus on model development and not worry about the platform. Containers can easily enable:

- A/B testing

- Deployment to multiple environments (IoT edge, local desktop, Azure infrastructure)

- Consistent environment configuration and setup to drive faster model development, test, and release cycles

- Model portability and scalability

Recommended next steps

The adoption of DevOps has been very effective at bringing together software development and operations teams to simplify and improve deployment and release processes. As AI and ML become increasingly more important components for applications, more pressure will exist to ensure they are part of an organization's DevOps model. The suggestions presented are examples of some steps to move towards an integration of two methodologies. To get started, please use some of the links below and please share your feedback and your experience!

- For an example of using an ML/AI solution that uses automation as a possible point of integration in the data pipeline, read the Mobile bank fraud solution guide (https://download.microsoft.com/download/0/1/5/0150425C-14C7-41F4-97EA-3DE57B678C51/IndSG_FraudDetection.pdf).

- Read Using the Team Data Science Process (TDSP) in Azure Machine Learning (https://blogs.technet.microsoft.com/machinelearning/2017/09/25/using-the-team-data-science-process-tdsp-in-azure-machine-learning/?WT.mc_id=devopsai-acomblog-kbaroni).

- Complete the tutorial DevOps for AI applications: Creating continuous integration pipeline on Azure using Docker and Kubernetes (https://docs.microsoft.com/azure/machine-learning/team-data-science-process/ci-cd-flask?WT.mc_id=devopsai-acomblog-kbaroni) to go through an example of setting up a Continuous Integration (CI)/Continuous Delivery (CD) pipeline for an AI application.

- Read Microservices and Docker containers: Architecture, Patterns and Development Guidance (https://blogs.msdn.microsoft.com/dotnet/2017/08/02/microservices-and-docker-containers-architecture-patterns-and-development-guidance/?WT.mc_id=devopsai-acomblog-kbaroni). This guide introduces Docker containers and provides an overview of microservices.

🔊 Subscribe (//azurecomcdn.azureedge.net/en-in/blog/feed/)

Explore

See where we are heading. Check out upcoming changes to Azure Products

Azure updates (/en-in/updates/)

Let us know what you think of Azure and what you would like to see in the future

Provide feedback (https://feedback.azure.com)

Topics

Announcements (/en-in/blog/topics/announcements/) (2222)

API Management (/en-in/blog/topics/api-management/) (33)

Artificial Intelligence (/en-in/blog/topics/artificial-intelligence/) (219)

Azure Maps (/en-in/blog/topics/azure-maps/) (25)

Azure Marketplace (/en-in/blog/topics/azure-marketplace/) (138)

Azure Stream Analytics (/en-in/blog/topics/azure-stream-analytics/) (31)

Big Data (/en-in/blog/topics/big-data/) (634)

Blockchain (/en-in/blog/topics/blockchain/) (88)

Business Intelligence (/en-in/blog/topics/business-intelligence/) (116)

Cloud Strategy (/en-in/blog/topics/cloud-strategy/) (626)

Cognitive Services (/en-in/blog/topics/cognitive-services/) (125)

Data Science (/en-in/blog/topics/datascience/) (110)

Data Warehouse (/en-in/blog/topics/data-warehouse/) (212)

Database (/en-in/blog/topics/database/) (595)

Developer (/en-in/blog/topics/developer/) (1174)

DevOps (/en-in/blog/topics/devops/) (77)

Events (/en-in/blog/topics/events/) (230)

Government (/en-in/blog/topics/government/) (67)

Hybrid (/en-in/blog/topics/hybrid/) (65)

Identity & Access Management (/en-in/blog/topics/identity-access-management/) (87)

Internet of Things (/en-in/blog/topics/internet-of-things/) (358)

IT Pro (/en-in/blog/topics/it-pro/) (592)

Last week in Azure (/en-in/blog/topics/last-week-in-azure/) (92)

Machine Learning (/en-in/blog/topics/machine-learning/) (36)

Management (/en-in/blog/topics/management/) (340)

Media Services & CDN (/en-in/blog/topics/media-services/) (206)

Migration (/en-in/blog/topics/migration/) (17)

Mobile (/en-in/blog/topics/mobile/) (157)

Monitoring (/en-in/blog/topics/monitor/) (138)

Networking (/en-in/blog/topics/networking/) (216)

Partner (/en-in/blog/topics/partner/) (113)

Security (/en-in/blog/topics/security/) (383)

Serverless (/en-in/blog/topics/serverless/) (66)

Storage, Backup & Recovery (/en-in/blog/topics/storage-backup-and-recovery/) (671)

Supportability (/en-in/blog/topics/supportability/) (45)

Updates (/en-in/blog/topics/updates/) (557)

Virtual Machines (/en-in/blog/topics/virtual-machines/) (689)

Web (/en-in/blog/topics/web/) (362)

Articles by date

July 2020 (/en-in/blog/2020/07/)

June 2020 (/en-in/blog/2020/06/)

May 2020 (/en-in/blog/2020/05/)

April 2020 (/en-in/blog/2020/04/)

March 2020 (/en-in/blog/2020/03/)

February 2020 (/en-in/blog/2020/02/)

Full archive (/en-in/blog/archives/)