Accelerate Your Machine Learning and Artificial Intelligence Journey Using These DevOps Best Practices

Published 12 November 2019 - ID G00463803 - 18 min read

By Analysts Arun Chandrasekaran, FARHAN CHOUDHARY

Artificial intelligence and machine learning initiatives are maturing across organizations, but EA and technology innovation leaders continue to face significant challenges in moving them to production. We provide best practices on how and where DevOps can help in accelerating operationalization.

Overview

Key Challenges

- There is a huge knowledge gap in understanding how to foster cohesive collaboration between data science teams and IT.
- Most enterprises view artificial intelligence (AI) projects mostly as a machine learning (ML) model development exercise. Hence, far less time and less budget are spent on data wrangling, model operationalization and integration with core enterprise infrastructure and applications.
- While traditional DevOps processes mostly deal with code changes, the dynamic nature of code, model and data changes, which are common in ML projects, pose significant operational challenges for innovation leaders.
- Technology innovation leaders are keen to apply DevOps principles for AI and ML projects, but they often struggle with architecting a solution for automating end-to-end

ML pipelines across data preparation, model building, deployment and production due to lack of process and tooling know-how.

Recommendations

Enterprise architecture (EA) and technology innovation leaders driving business transformation through technology innovation should:

- Reduce the friction around data management by enabling a DataOps culture that can automate the design, deployment and management of data pipelines for ML with lean methodologies and appropriate levels of metadata and governance.
- Implement a model management system along with a DevOps toolchain to automate and streamline model development, testing and deployment, and to enable a hybrid cloud strategy.
- Institute an MLOps practice to improve model tracking, versioning and monitoring, and to constantly reevaluate model accuracy and potential business value.
- Drive end-to-end automation, enable collaboration between data scientists and the IT team, and create shared KPIs and operational SLAs between them to deliver the true business value of AI and ML.

Strategic Planning Assumptions

Through 2022, more than 80% of enterprise organizations piloting AI projects will face delays in moving them to production due to lack of organizational collaboration and IT process immaturity.

By 2022, at least 50% of machine learning projects will not be fully deployed in production.

Introduction

According to the 2019 Gartner CIO Survey, AI and ML continue to be viewed as the No. 1 game-changing technology by CIOs (see "2019 CIO Survey: CIOs Have Awoken to the Importance of AI"). However, most organizations underestimate how long it will take to move AI and ML projects into production. Among the key challenges that organizations face with operationalizing ML are security/privacy concerns, complexity in integrating AI workloads with their core infrastructure, building an automation pipeline and ensuring

cohesive collaboration between data science teams and IT. While developing ML models has become much easier, seamless data preparation and model deployment and operations continue to be elusive for most organizations. These are the most common reasons why ML operations (MLOps) is challenging:

- Data scientists are usually focused on model building, training and validation processes, but they do not take into consideration how the model will be deployed and monitored in a production environment. This leads to a significant gap between pilot and production environments.
- Organizations lacking a DevOps culture either fail to create distinct roles such as data engineers, data scientists, AI (or ML) architects and AI (or ML) engineers responsible for end-to-end pipeline management or these teams end up operating in silos.
- The heterogeneity of data, models and tools is overwhelming the scarce resources allocated to the AI and ML project, which exacerbates operational excellence.
- There is a lack of know-how on best practices for moving ML models from pilot to production.

What Is DevOps?

DevOps is a customer-value-driven approach to delivering solutions using agile methods, collaboration and automation. DevOps emphasizes people, culture and collaboration between development, data science, operations and other stakeholders to improve the delivery of customer value. DevOps implementations seek to continually improve the flow of work by removing constraints with the intent of improving the delivery of customer value as a result.

Applying DevOps principles for machine learning could potentially solve a number of these challenges and enable a more collaborative and highly automated development life cycle for ML. While DevOps as we know it mostly deals with high-velocity code changes, machine learning projects have highly dynamic changes in code, models and data, which need to be accounted for. Hence, DevOps principles need to be applied across the data pipeline (DataOps) and ML model pipeline (MLOps) stages for organizations to reap the most benefit.

Figure 1 illustrates the different steps in an ML workflow:

Figure 1. Machine Learning Workflow Phases

Analysis

Create a DataOps Culture

Businesses nowadays run at a fast pace, and data management should keep up with this pace. With the increasing proliferation of data sources and the realization that data can be a competitive advantage, if the data is not agile enough, it will be dropped from the decision-making process. Data teams are now facing a challenge like application developers once did — now, instead of developers writing code, teams are solely reliant on data scientists to prepare data, explore data and extract insights from data. The problem here is that no matter how innovative data scientists get, they won't be able to help the business if they don't have access to the right data at the right time.

The DevOps culture transformed the agility in code development, and a similar culture is required on the data side as well. Therefore, creating a DataOps culture has become a critical discipline for any organization that wants to have a competitive edge and survive in this data-driven market.

DataOps is an automated, process-driven methodology used by data teams to improve quality and reduce the time to deployment of data management and analytics. It can be applied to the entire data life cycle, from ingestion to storage, encoding, preparation and delivery. It recognizes the interconnected nature of the analytics and operations teams. DataOps is a collaboration across data engineering, data quality, data integration and data security, revolving around the data engineer.

The prerequisite to creating a DataOps culture is to first create a data-driven organization. A data-driven organization is one that realizes the importance of data and its accessibility, and enables all the users involved in the data initiative to have seamless collaboration. Most data-driven organizations have a cohesive data strategy and best practices for supporting operational analytics. EA and technology innovation leaders need to borrow/elevate the same best practices for the data pipelines built for ML and AI, thereby crystalizing (defining) the DataOps culture. An enterprise deals with a plethora of internal and external data sources that come from business applications, product applications, public and private customer touchpoints, IoT or monitoring systems, third parties, and many others. While the data is constantly collected, analytics always becomes an afterthought.

Gartner recommends taking these steps to create a DataOps culture:

1. **Examine, evaluate and consolidate all data sources.** The first step toward achieving and enabling a DataOps culture is to evolve toward a unified data repository. Creating a common data infrastructure across the organization will allow you to not only capture the right data but also create the necessary management of metadata and data lineage (see "A Guidance Framework for Operationalizing Machine Learning for AI"). Metadata management will thereafter assist teams in finding data for their various tasks. The common repository of data helps everyone involved in the data journey to collaborate around data and reduces data silos. With the standardization of data across the organization, data engineers can easily work to create applications, and the business users can use the same data to arrive at their decisions. With a consolidated repository of data available to all users, the users can not only use the data at a granular level but also use it in a holistic way, which would help them understand the bigger picture.

2. **Enable a data-driven culture by filling the skills gaps at the management level.** While organizations have data engineers, data analysts and data scientists for wrangling, munging, exploring and modelling, the top management lacks the skills to understand the value of their data. It's important for management and different stakeholders to understand the motivation behind allowing data-driven decisions to take place by liaising with data users. Understanding the importance of creating this culture would also help with executive buy-in while making strategic decisions.

3. **Convene a community of data users for idea exchange.** All users involved at any stage of the data journey should be allowed to freely exchange ideas among themselves. This can be achieved by creating a community either online or offline within the organization. It would allow the data users to come up with creative ways data can be used, create transparency and trust by cross-functional collaboration and, in some instances, provide a fresh perspective to an ongoing challenge. With the creation of this community, every decision will be backed up with data, and data users will have a strong case to present to the management in case their finding contradicts the understanding or executive decision. By creating a DataOps culture, organizations must also look at creating other roles that can assist in data exploration, reporting and dashboarding (see "Build a Comprehensive Ecosystem for Citizen Data Science to Drive Impactful Analytics").

4. **Identify the right tools to get started.** Organizations have a limited number of data engineers and data scientists who can derive predictive insights. Even if your data is readily accessible in a unified repository, most of your employees lack the necessary

skills or tools that can enable them to explore this data. This problem can be solved by identifying and investing in the right tools. At minimum, the data tools should support easier data access and management (connection to various data formats, ETL operations, metadata and lineage management, and data synchronization), data exploration and visualization (search, metadata analysis, visualization, and data transformation and modeling). You can pick the appropriate tools depending on your vision, strategy, how far off are you in your analytics journey and ease of integration. This step requires active training to be imparted to your employees and then assessing the capabilities of each employee in terms of their understanding of statistics, data exploration and business context.

5. **Align all processes with business goals and allow everyone to contribute.** It is imperative to understand that data belongs to everyone, not just the IT, data analytics or data science teams. Therefore, every team should be allowed to create its own analysis of the data following the right procedure. It is understandable that with more data access it will become difficult to keep track of changes. Hence, necessary guard rails must be put in place to protect the data by assigning the role of a data officer within each team who would be responsible for all analysis carried out by the team members in a given business function or process. Audit trails must be established to ensure the integrity of the data, and the analysis should be met with swift actions from the management, acceptance and suggestions for improvement — only then can a holistic DataOps culture be established.

6. **Utilize quality data.** There's a common misconception that building machine learning models requires huge quantities of data. While that is partly true, quality data is much more important for the success of machine learning models. If poor quality data is ingested in the models, this data can become the primary cause of project failure. Assessing the quality of data and improving it should be the very first steps of any machine learning project. This would typically include checking data for consistency, accuracy and inherent biases (and identifying them early on), completeness, duplicity, missing values, data corruption, and compatibility. Otherwise, ensuring data quality near the operationalization/production phases or while scaling up is a near impossibility.

Establish MLOps Practices for End-to-End ML Life Cycle Management

Machine learning operations (MLOps) aims at streamlining the development, deployment and operationalization of ML models. It supports the build, test, release, monitoring, performance tracking, reuse, maintenance and governance of ML models.

Agile methodology should be used during the development and deployment of machine learning projects. Agile usually involves collaboration across multiple layers of stakeholders along with cyclical testing and rapid prototyping. Agile aids in establishing a cleaner communication channel and helps in creating teams with higher levels of trust. This leads to efficient project management, allowing ideas, features and feedback to be introduced in the loop at any time.

Similar to a DevOps process where we tightly rein in the code development, testing, delivery and management of applications, in a ML process it is important to manage model code, boot scripts, parameters and the trained model. It is important to build an abstraction in the model build phase that will decouple it from the deployment environment so that models can be reused and are independent of the target environment.

While ML workloads can be deployed on any virtualization technology, containers are a natural fit in this context and can help in bringing machine learning models into production faster with consistent packaging and automated code pipeline. Container-specific orchestrators such as Kubernetes can simplify the deployment, scaling and availability of AI workloads across multiple target environments. In addition, projects such as Kubeflow create a synchronized data science workflow and allow you to train machine learning models built in different frameworks in a distributed manner, obviating the need for manual configuration of training infrastructure. Integrating this with CI/CD pipelines can automate the model deployment process.

Only half of the models created under various machine learning projects make it to a production stage. In order to push the model into production, data scientists typically hand over their model to data and machine learning engineers. The two practices, however, operate in silos, making it challenging for teams to work collaboratively to achieve their common goal.

Deploying machine learning projects is particularly challenging because it poses some unique hurdles:

- **Programming language mismatch during model coding and deployment.** While data scientists work with open-source coding languages such as R or Python for building their machine learning models, porting these scripts into a more production-friendly environment like C++ or Java becomes complicated. This often results in reduced performance in terms of the speed and accuracy of the machine learning models.

- **Tracking model changes postdeployment.** Machine learning models churn outputs based on the features in the model; changing any feature in the model affects the model accuracy. Since models are subject to small changes to improve accuracy, tracking configuration updates while also maintaining the clarity of configuration becomes an additional burden. No clear roles are defined for this specific task, diluting the efforts made by data scientists and ML engineers.

- **Concept drift.** Inference output with ML models also keeps degrading in performance over time; these changes can either be abrupt or seasonal. Tracking changes with the data and monitoring models becomes paramount, especially if the new data is being constantly fed back into the model (i.e., if the model is not static). Machine learning models need to be updated much more frequently than traditional software applications.

- **Usage patterns of machine learning models.** Machine learning models tend to work in batches, which causes spikes in the computational usage. That is, some models will experience higher traffic during certain times and some models, on the other hand, might not even be used for days. Automating the scalability of the ML platform environment is extremely critical but is hard for most organizations.

- **Lack of heterogeneity in teams.** The role of any individual involved in the release, activation, monitoring, performance tracking, management, reuse, maintenance and governance of ML models requires a fair bit of research. However, organizations tend to consider researchers as individual contributors, which often leads to their isolation from the feedback cycle obtained from customers, product owners and developers.

To overcome these hurdles, enterprise architecture and technology innovation leaders can use the following methods:

- **Overcome language barrier.** While data scientists build machine learning models with different programming languages, containerization can solve incompatibility or portability issues. Machine learning teams can use tools or platforms such as Simplified Wrapper and Interface Generator (SWIG) that facilitate interfacing or translating code from one language to another.

- **Address reproducibility, traceability, integrity and integrability.** Data scientists build many versions of their machine learning models (coupled with codes written in different languages) using different libraries, or different versions of libraries, and changing the data transformation or model approach itself. It becomes particularly

challenging to keep a track of these dependencies manually. Using model management systems and ML life cycle tools with automatic logging capabilities can help overcome the issues of:

- Reproducibility, by providing references to the model code in previous and current releases along with the data that was used to train the model.
- Traceability, by maintaining references to all data engineering scripts that were used to explore, enrich and transform data for model consumption. Adopt a git-based approach to commit changes and track them.
- Integrity, by providing references to the outputs and logs related to model bias or drift.
- Integrability, by capturing the metadata in all steps involved in the operationalization phase.

- **Create a heterogeneous team.** Operationalizing machine learning models is an iterative process. There are researchers who work on creating new features and developers that focus on pushing the features into production. Creating a team with a good mix of researchers and developers introduces an overlap that assists in collaboration and cross-role knowledge sharing.
- **Focus on retrospective and continuous improvement.** Working in heterogeneous teams requires teams to not only discuss what work is being done but also how it is being done. Adapting a methodology that works according to the needs of the individuals in the team drives the team to be self-empowered and free in their thinking. Adopting agile in research-intensive environments focuses more on improvement iterations instead of retrospection; therefore, it is all the more important to keep revisiting work methods and goals, and defining clear outcomes.

In addition, adequate care must be taken when releasing new models into production. While there will be a need to update ML models frequently, EA and technology innovation leaders need to automate the continuous delivery pipeline to reduce errors and to accelerate time to value.

There are three common ways to release ML models into production:

- **Canary release.** The canary release deploys the new version to a subset of users before rolling it out to all users. You can then monitor your customers' responses to see how aggressively you want to proceed with the release across the entire environment. Canary releases let you mitigate the cost of errors without spending an enormous amount of time and money on preventive testing. However, canary deployments are much slower due to the selective release cycle.

- **Blue/green release.** In a blue/green release approach, you have two production environments, as identical as possible. As you prepare a new release of your software, you do your final stage of testing in the green environment. Once the software is working in the green environment, you switch the traffic so that all incoming requests go to the green environment. If a problem is detected, you switch back to the original blue environment. While blue/green releases can provide instant model rollout, they are expensive due to the need for redundant resources and the need to do extensive testing when compared to canary release.

- **Challenger/champion deployment.** A challenger/champion deployment model is well suited to the task of testing new and multiple competing ML and AI models in a production environment. In this approach, multiple versions of a model "shadow" the primary production model (the champion). To determine which of the competing models is the best, you send all to production and track the business outcomes over time to determine which one is the best. While this approach provides "survival of the fittest," fair testing, patience and being decisive are critical to achieving success.

Recommendations:

- Unlock the potential benefits of using machine learning models by using an agile approach to take insights, validate models, optimize core resources and accelerate decision making and information usage from all domains to create a more transparent solution.

- Deploy a container-/Kubernetes-based architecture to bring all the tools and methodologies together to solve your heterogeneity and infrastructure challenges.

- Monitor all models with logs, inputs, outputs and exceptions using different performance visualization techniques to keep a close eye on model performance.

- Assess the pros and cons of automated ML model release options, and determine the right fit for your organization.

Evidence

This research is based on more than 50 phone inquiries and face-to-face interactions with Gartner clients.

"Why 60 Percent of Machine Learning Projects Are Never Implemented," Grit Daily.

"The Rise of the Term 'MLOps'," Medium.

Appendix

Sample List of DataOps Vendors (Not Exhaustive)

- Databricks
- DataKitchen
- Immuta
- Pachyderm
- Palantir Foundry
- Qubole
- Quilt Data
- StreamSets
- Tamr

Sample List of Augmented Analytics and MLOps Vendors (Not Exhaustive)

- Algorithmia
- Alteryx
- Amazon Web Services (AWS)
- DataRobot
- Datatron
- Google (Google Cloud Platform)

- H2O.ai
- Hydrosphere.io
- IBM
- Microsoft Azure
- ModelOp
- Oracle
- Paperspace
- Salesforce
- SAP
- SAS
- Seldon

Recommended by the Authors

How to Build Machine Learning and Artificial Intelligence Into Production Applications

A Guidance Framework for Operationalizing Machine Learning for AI

Recommended For You

Control Bias and Eliminate Blind Spots in Machine Learning and Artificial Intelligence

Enabling Data Quality for Machine Learning and Artificial Intelligence

Accelerate Your DevOps Initiative With Scrumban

How to Build Machine Learning and Artificial Intelligence Into Production Applications

Transform Enterprise Search and Insight With Machine Learning and Artificial Intelligence