DZone
A DEVADA MEDIA PROPERTY

# Machine Learning Trend Report

Deploying Models at Scale

BROUGHT TO YOU IN PARTNERSHIP WITH

DZone
A DEVADA MEDIA PROPERTY

# Table of Contents


3 Highlights & Introduction
BY KARA PHELPS

5 Machine Learning in Microservices
BY TIM SPANN

8 How to Tackle the Challenges in Deploying Machine Learning Models
BY FRANCESCA LAZZERI, PHD AND AASHISH BHATEJA

12 Key Research Findings
BY JORDAN BAKER

16 Diving Deeper Into Machine Learning

17 Executive Insights Into Machine Learning
BY TOM SMITH



**PRODUCTION**
**Billy Davis** Production Coordinator
**Naomi Kromer** Sr. Campaign Specialist
**Avery Falkenstein** Campaign Specialist

**EDITORIAL**
**Mike Gates** Senior Editor
**Jordan Baker** Publications Associate
**Andre Lee-Moye** Content Coordinator
**Lauren Ferrell** Content Coordinator
**Lindsay Smith** Content Coordinator
**Charleigh Smith** Content Coordinator
**Peter Connolly** Content Coordinator

**MARKETING**
**Susan Wall** CMO
**Aaron Tull** Dir. of Demand Generation
**Waynette Tubbs** Dir. of Marketing Comm.
**Colin Bish** Member Marketing Spec.
**Suha Shim** Acquisition Marketing Mgr.
**Cathy Traugot** Content Marketing Mgr.
**Melissa Habit** Content Marketing Mgr.
**Kate Lloyd** Customer Marketing Mgr.
**Lisa Cronk** Graphic Design Specialist
**Blake Ethridge** Community Manager

**BUSINESS & PRODUCT**
**Terry Waters** CEO
**Aaron Bailey** General Manager, Media
**Jesse Davis** EVP, Technology
**Kellet Atkinson** Media Product Manager

**SALES**
**Tevano Green** Sr. Account Executive
**Alex Crafts** Key Account Manager
**Capel States** Key Account Manager

ORIGINAL RESEARCH

# Highlights & Introduction

**By Kara Phelps,** Editorial Project Manager at DZone

The terms "artificial intelligence" and "machine learning" are often treated as though they're interchangeable, but they shouldn't be. ML is AI, but AI isn't necessarily ML. Artificial intelligence, or AI, is simply and most broadly defined as "the use of computers to mimic the cognitive functions of humans." Machine learning, or ML, occupies a more narrow category within that definition — it involves machines training with models and "learning" from existing data in order to make predictions. In order for a machine to learn, in other words, algorithms allow it to explore data and extrapolate based on specific objectives.

Intertwined with the field of data science and data analytics, machine learning is constantly expanding in its scope and sophistication. ML has evolved from a simple application programmed by Arthur Samuel at IBM in 1952, which taught a computer to play a game of checkers, to today's complex algorithms behind the decision-making in self-driving cars.

**TREND PREDICTIONS**

- Machine learning and its models will become even more enmeshed with data analytics as enterprises demand more real-time access to data.
- The roles of the developer and the data scientist will begin to overlap and will eventually merge.

AI and ML topics have been popular among DZone readers over the last several years — so much so, in fact, that we decided to launch the AI Zone in 2017. DZone has also released several publications covering artificial intelligence in recent years, but this is the first Trend Report to focus exclusively on machine learning and its applications in the world of software development. Our Trend Reports are meant to drill down and examine trends within trends, so we decided to hone this year's AI-oriented topic into machine learning for developers.

Tim Spann discusses how developers can best architect their microservices to handle machine learning in his article, "Machine Learning in Microservices." He reviews options for building and running machine learning at scale within real-time data streams and flows.

In “How to Tackle the Challenges in Deploying Machine Learning Models,” Francesca Lazzeri and Aashish Bhateja explore why successful ML model deployment is essential for AI-driven companies, why they frequently struggle with it, and how to choose the best tools for your use case.

## Intertwined with the field of data science and data analytics, machine learning is constantly expanding in its scope and sophistication

Finally, Tom Smith shares the thoughts of executives from 16 different tech organizations in his Executive Insights on the state of machine learning in software development. These executives sat down for interviews with DZone to offer their advice on best practices in ML, as well as their predictions on what the future of ML will look like.

We appreciate your interest in the future of machine learning, and we’d like to thank you for downloading this report. Happy reading!

OPINION

# Machine Learning in Microservices

**By Tim Spann,** Field Engineer Data in Motion at Cloudera

Microservices have become the preferred methodology for delivering targeted functionality to various enterprise applications. As they are often hosted on various cloud, Kubernetes, or PaaS environments such as OpenShift, we've had to assume that machine learning would eventually join them there, as much of the algorithmic functionality is being enhanced or augmented by machine learning and deep learning. Let's investigate why this is happening, and how you can best architect your microservices for handling machine learning.

It is now unreasonable to wait for data to land in a final data store, such as HDFS or S3. Before it becomes available for retraining machine learning and deep learning models, it needs to be used for classification or model execution. The data must be available immediately, streaming at incredible velocities from devices, gateways, logs, web applications, mobile applications, blockchain nodes, smart meters, and containers in every cloud. This ever-changing, fast data requires more agile means of tapping into it. It demands more than just simple SQL analytics. As fast data becomes relevant for so many use cases — from IoT, cybersecurity, and log analytics to SQL change data capture and more — we will find the need to push machine learning execution out to every edge.

**TREND PREDICTIONS**

- The increasing speed of data means the use of machine learning with edge devices and microservices will increase.
- Organizations will increasingly deploy machine learning with microservices, as it allows them to target a specific functionality.

Every day, a Fortune 500 customer approaches me with questions in a meeting. No longer do they just ask how to obtain their data at scale, but how they can ingest at scale with simultaneous storage to multiple cloud endpoints and Hadoop — while providing data for retraining their machine learning models and also running those models against incoming data. Everyone needs sub-second responses and real-time streaming analytics. For some of this, SQL will be adequate, such as Apache Calcite queries inside Apache NiFi on Dataflows. Oftentimes, however, we require additional analytics, such as YOLO analysis on images or sentiment analysis on unstructured chunks of ingested text from calls and chats. Fortunately, there are options that can be handled in a properly built environment to allow developers to build microservices that add machine learning to this process.

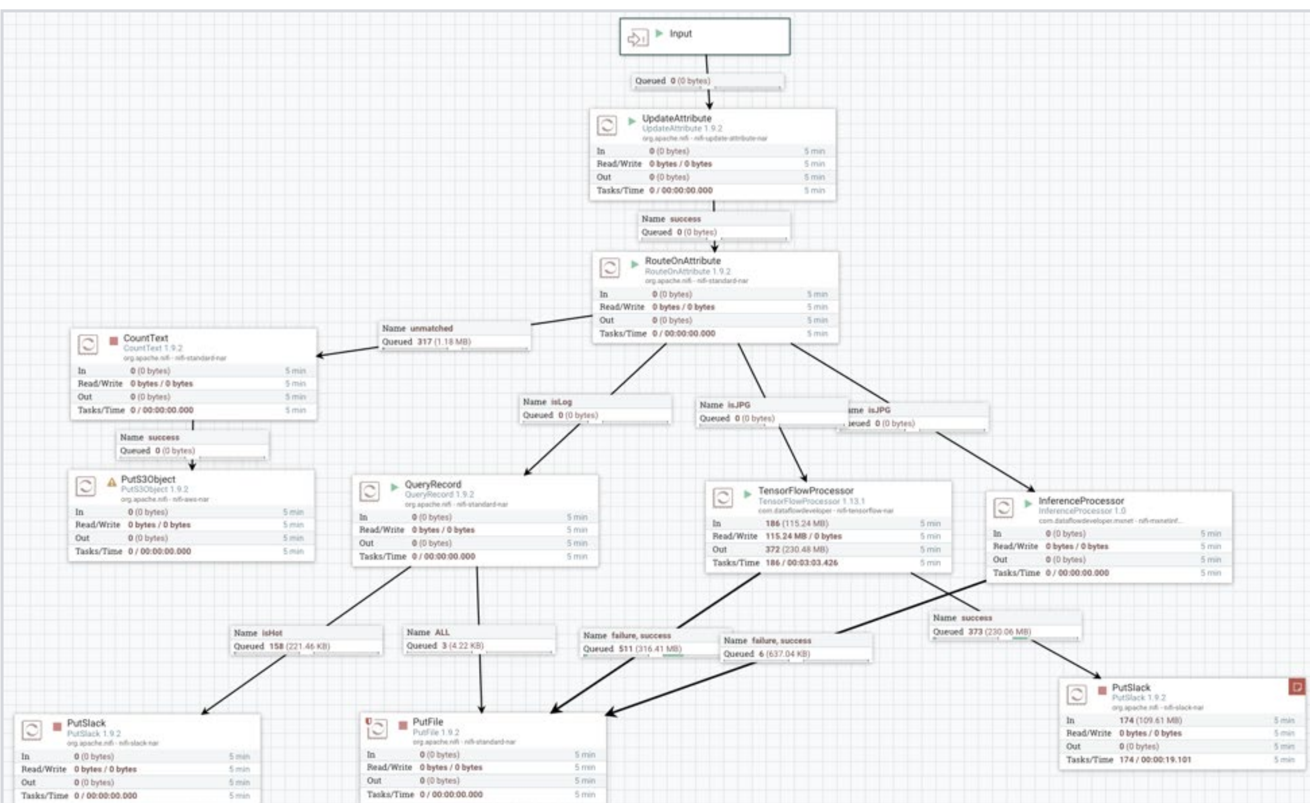
Unfortunately, like most things in enterprise IT, definitions can be nebulous or defined by many competing organizations, corporations, or media entities. I do not wish to join one of these factions. Having come from a microservices-enabling entity, Pivotal, I will just try to find a baseline that makes sense. You can add or subtract pieces from my definition as need be. We can also go further and make our services serverless. These are even lighter weight and may be more appropriate, depending on the use case. If you are evaluating such a paradigm, please check out the Open Source Apache NiFi Stateless container to see if fits your needs. I recommend that you also look at Apache OpenWhisk or the FNProject. There are many options of where to run these services, from native cloud to YARN to projects on Kubernetes. (However you decide to run these, take a look at this interesting universal scheduler, YUNIKORN.)

For now, let's stick to basic microservices, which I will define as maintainable, testable, loosely coupled services that can be independently deployed with one real use case/function. (See: https://martinfowler.com/articles/microservices.html ) This definition makes it a little simpler and keeps us from haggling over whether or not something is a microservice if it does not have all the factors of a 12-factor app.

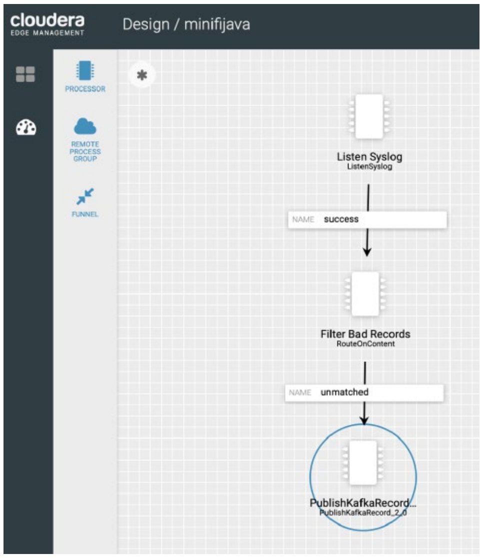
So, for the purpose of running machine learning at scale as part of real-time data streams and flows, I suggest some of the following options: Apache Kafka Streams, Spring Boot, Apache Flink, Apache NiFi Flows, and CSDW Models. To allow you to run all of these models in the same environment — which I have done — I recommend having Apache Kafka as the decoupling agent. Having this securely managed, enterprise streaming platform as our distribution channel provides much of the infrastructure, security, scalability, decoupling, and monitoring that is needed in such a system. Using specialized tools, you can obtain full insight into what's happening with these complex machine learning pipelines.

Let's look at a few ways of using microservices for machine learning and deep learning.

Our first case is the easiest. I want to run YOLO SSD processing on an image as it comes off an IoT camera. In this use case, as the image is loaded by MiNiFi, it is sent to Apache NiFi. Apache NiFi can run PMML, ONNX, TensorFlow Lite, or other models via custom processors, such as the ones I have written in Java for Apache MXNet and TensorFlow. This method is very quick, but requires that at least one part of your machine learning pipeline live directly inside of Apache NiFi. We can also push the model all the way to the edge and have the MiNiFi agent run the model right at the edge device or gateway.

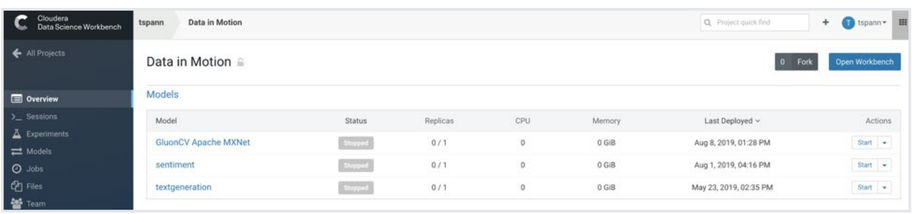


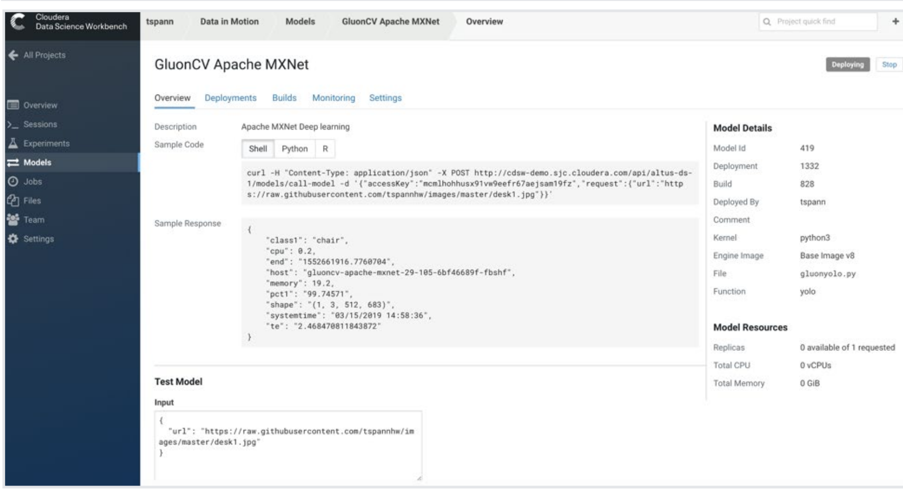

A second option is to have a very slim process, such as a minimal MiNiFi flow, which captures the source data — whether it's an image from a camera, text from a log, or time series data from a sensor — and sends it directly to a Kafka topic for consuming and pro-




cessing by a Kafka Stream or Flink microservice that can execute your machine learning model. You can run your model via Flink ML or have Flink call an existing PMML model. You can also execute an ONNX model from Flink or KafkaStreams. There are many options for models, including TensorFlow Lite, ONNX, PMML, DeepLearning4Java, H2O, and Pickle. Most of these model runners can be distributed and used by most of the microservices options. Again, you need to decide which language and stream processing engine makes the most sense for your use cases.

A third option is to build machine learning microservices in Python, R, or Scala. You can host your microservices in an enterprise environment with full security running on Kubernetes. As shown in the screenshots below, it is very easy to host your model and provide a REST API for integration. This option is great for easy integration of your machine learning and deep learning models with all your other microservices. It also lets data scientists work in an agile collaborative environment, allowing them to easily deploy evolving models. If you need to use asynchronous Kafka to decouple this from other processes, we can use Apache NiFi to convert Kafka messages into calls to the REST microservice, and then package the results in a second Kafka message. Apache NiFi will act as your universal router and gateway for microservices and messages at scale.




Another use case is the processing of streaming textual data, such as social media, as it arrives. We can use this text data to expand our machine learning corpus for our model. Using Apache Kafka as a universal messaging gateway, for this last use case, we will have Apache NiFi ingest a social media feed — say, real-time tweets — and push these tweets as AVRO options with schemas with a standard schema we have built. We use the Cloudera Schema Registry as a universal service that anyone can interface with using a simple REST API. More decoupling of our microservices makes it easy to send these messages in AVRO to many subscribers via our Kafka distribution channel. One such consumer of the Kafka messages could be a custom Java microservice written in Apache Flink to apply custom NLP machine learning to analyze any given tweet for content and sentiment. The results of this would be sent to another topic in Kafka, for either processing by Apache NiFi in production (Big Data stores such as Kudu, HBase, Hive, or Impala) or possibly straight to an object store such as Amazon S3.

OPINION

# How to Tackle the Challenges in Deploying Machine Learning Models

**By Francesca Lazzeri, PhD,** Senior Machine Learning Scientist
**and Aashish Bhateja,** Senior Program Manager, Azure Machine Learning

## Introduction

Artificial intelligence offers companies the potential to transform their operations: from applications that predict and schedule equipment maintenance, to intelligent R&D systems that estimate the cost of new drug development, to AI-powered tools for HR that enhance the hiring process and employee retention strategy. However, in order to be able to leverage this opportunity, companies must learn how to successfully build, train, test, and push hundreds of machine learning models to production. They also need to move models from development to their production environment in ways that are robust, fast, and repeatable.

Today's data scientists and developers have a much easier experience when building AI-based solutions through the availability and accessibility of data and open source machine learning frameworks. This process becomes a lot more complex, however, when they need to think about model deployment and pick the best strategy to scale up to a production-grade system.

**TREND PREDICTIONS**

- By deploying machine learning models, other teams in your company can use them, send data to them, and get their predictions, which are in turn populated back into the company systems to increase training data quality and quantity.
- Once this process is initiated, companies will start building and deploying higher numbers of machine learning models in production. They will master robust and repeatable ways to move models from development environments into business operations systems.

In this article, we will introduce some common challenges of machine learning model deployment. We will also discuss the following points that may enable you to tackle some of those challenges:

1. Why successful model deployment is fundamental for AI-driven companies.
2. Why companies struggle with model deployment.
3. How to select the right tools to succeed with model deployment.

## Why Successful Model Deployment Is Fundamental for AI-Driven Companies

Machine learning model deployment is the process by which a machine learning algorithm is converted into a web service. We refer to this conversion process as *operationalization*: to operationalize a machine learning model means to transform it into a consumable service and embed it into an existing production environment.

Model deployment is a fundamental step of the Machine Learning Model Workflow (Figure 1). Through machine learning model deployment, companies can begin to take full advantage of the predictive and intelligent models they build, develop business practices based on their model results, and, therefore, transform themselves into actual AI-driven businesses.

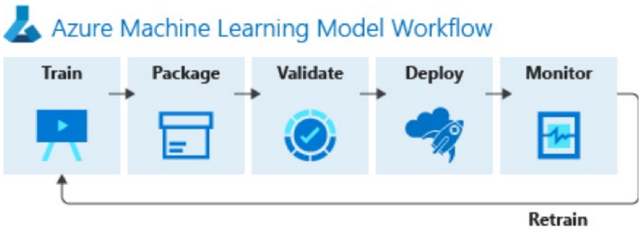



*Figure 1 — Azure Machine Learning Model Workflow. Source: www.aka.ms/AzureMLDoc*

When we think about AI, we focus our attention on key components of the machine learning workflow, such as data sources and ingestion, data pipelines, machine learning model training and testing, how to engineer new features, and which variables to use to make the models more accurate. All these steps are important; however, thinking about how we are going to consume those models and data over time is also a critical step in every machine learning pipeline. We can only begin extracting real value and business benefits from a model's predictions when it has been deployed and operationalized.

We believe that successful model deployment is fundamental for AI-driven enterprises for the following key reasons:

- Deployment of machine learning models means making models available to external customers and/or other teams and stakeholders in your company.
- By deploying models, other teams in your company can use them, send data to them, and get their predictions, which are in turn populated back into the company systems to increase training data quality and quantity.
- Once this process is initiated, companies will start building and deploying higher numbers of machine learning models in production and master robust and repeatable ways to move models from development environments into business operations systems.

## Why Companies Struggle With Model Deployment

Many companies see AI-enablement effort as a technical practice. However, it is more of a business-driven initiative that starts within the company; in order to become an AI-driven company, it is important that the people who currently operate and understand the business begin to collaborate closely with those teams who are responsible for the machine learning deployment workflow.

As illustrated in Figure 2 below, each step of a machine learning deployment workflow is based on specific decisions about the different tools and services that need to be used in order to make the deployment successful, from model training and registration to model deployment and monitoring:

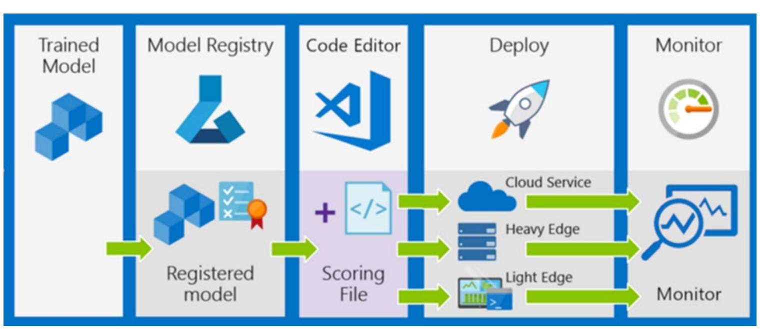



*Figure 2 — Machine Learning Deployment Workflow. Source: www.aka.ms/AzureMLDoc*

Right from the first day of the AI application development process, machine learning teams should interact with business counterparts. It is essential to maintain constant interaction to understand the model *experimentation* process parallel to the model *deployment* and *consumption* steps. Most organizations struggle to unlock machine learning's potential to optimize their operational processes and get data scientists, analysts, and business teams speaking the same language.

Moreover, machine learning models must be trained on historical data. This demands the creation of a prediction data pipeline — an activity requiring multiple tasks, including data processing, feature engineering, and tuning. Each task — down to versions of libraries and handling of missing values — must be exactly duplicated from the development to the production environment. Sometimes, differences in technology used in development and in production contribute to difficulties in deploying machine learning models.

Companies can use machine learning pipelines to create and manage workflows that stitch together machine learning phases. For example, a pipeline might include data preparation, model training, model deployment, and inference/scoring phases. Each phase can encompass multiple steps, each of which can run unattended in various compute targets. Pipeline steps are reusable and can be run without rerunning subsequent steps if the output of that step hasn't changed. Pipelines also allow data scientists to collaborate while working on separate areas of a machine learning workflow.

## How to Select the Right Tools to Succeed With Model Deployment

Building, training, testing, and finally deploying machine learning models is often a tedious process for companies that are looking at transforming their operations through AI. Moreover — even after months of development that delivers a machine learning model based on a single algorithm — the management team has little means of knowing whether their data scientists have created a great model, or how to scale and operationalize it.

Below we share a few guidelines on how a company can select the right tools to succeed with model deployment. We will illustrate this workflow using Azure Machine Learning Service, but it can be also used with any machine learning product of your choice.

The model deployment workflow should be based on the following three simple steps:

1. Register the model.
2. Prepare to deploy (specify assets, usage, compute target).
3. Deploy the model to the compute target.

### Register Your Model

A registered model is a logical container for one or more files that make up your model. For example, if you have a model that is stored in multiple files, you can register them as a single model in the workspace. After registration, you can then download or deploy the registered model and receive all the files that were registered.

Machine learning models are registered when you create an Azure Machine Learning workspace. The model can come from Azure Machine Learning or can come from somewhere else.

### Prepare to Deploy

To deploy a model as a web service, you must create an inference configuration (InferenceConfig) and a deployment configuration. Inference, or model scoring, is the phase where the deployed model is used for prediction, most commonly on production data. In the inference config, you specify the scripts and dependencies needed to serve your model. In the deployment config, you specify details of how to serve the model on the compute target.

The entry script receives data submitted to a deployed web service and passes it to the model. It then takes the response returned by the model and returns that to the client. The script is specific to your model; it must understand the data that the model expects and returns.

The script contains two functions that load and run the model:

- init(): Typically, this function loads the model into a global object. This function is run only once when the Docker container for your web service is started.
- run(input_data): This function uses the model to predict a value based on the input data. Inputs and outputs to be run typically use JSON for serialization and de-serialization. You can also work with raw binary data. You can transform the data before sending it to the model, or before returning it to the client.

When you register a model, you provide a model name used for managing the model in the registry. You use this name with Model.get_model_path() to retrieve the path of the model file(s) on the local file system. If you register a folder or a collection of files, this API returns the path to the directory that contains those files.

### Deploy to Target

Finally, before deploying, you must define the deployment configuration. The deployment configuration is specific to the compute target that will host the web service. For example, when deploying locally, you must specify the port where the service accepts requests. The following compute targets, or compute resources, can be used to host your web service deployment:

| Compute target | Usage | Description |
|---|---|---|
| Local web service | Testing/debug | Good for limited testing and troubleshooting. Hardware acceleration depends on using libraries in the local system. |
| Notebook VM web service | Testing/debug | Good for limited testing and troubleshooting. |
| Azure Kubernetes Service (AKS) | Real-time inference | Good for high-scale production deployments. Provides fast response time and autoscaling of the deployed service. Cluster autoscaling is not supported through the Azure Machine Learning SDK. To change the nodes in the AKS cluster, use the UI for your AKS cluster in the Azure portal. |
| Azure Container Instances (ACI) | Testing or dev | Good for low scale, CPU-based workloads requiring <48 GB RAM. |
| Azure Machine Learning Compute | Batch inference | Run batch scoring on serverless compute. Supports normal and low-priority VMs. |
| Azure IoT Edge | IoT module | Deploy and serve ML models on IoT devices. |
| Azure Data Box Edge | via IoT Edge | Deploy and serve ML models on IoT devices. |

Finally, below we show a few examples of creating a deployment configuration for some of the compute targets:

| Compute target | Deployment configuration example |
|---|---|
| Local | `deployment_config = LocalWebservice.deploy_configuration(port=8890)` |
| Azure Container Instance | `deployment_config = AciWebservice.deploy_configuration(cpu_cores = 1, memory_gb = 1)` |
| Azure Kubernetes Service | `deployment_config = AksWebservice.deploy_configuration(cpu_cores = 1, memory_gb = 1)` |

## Conclusion

In this article, we introduced some common challenges of machine learning model deployment. We also discussed why successful model deployment is fundamental to unlock the full potential of AI, why companies struggle with model deployment, and how to select the right tools to succeed with model deployment. If you want to learn more about machine learning and model deployment, visit the following pages:

- Azure Machine Learning Service
- Azure Machine Learning for Visual Studio Code
- Get Started with Azure Machine Learning
- Deploy Models with Azure Machine Learning
- Azure Machine Learning Notebooks

ORIGINAL RESEARCH

# Key Research Findings

**By Jordan Baker,** Publications Associate at DZone

## Demographics

For this year's Machine Learning survey, we asked respondents how both they and their organization use machine learning, what their organizations are doing to further their use of machine learning, and the tools, languages, and frameworks popular among those working with machine learning.

But before we dive into the analysis of these topics, let's quickly take a look at the demographic information for this survey's respondents.

- A majority of respondents work for enterprise-level organizations.
  - 24% work for organizations sized 1,000-9,999.
  - 20% work for organizations sized 100-999.
  - 19% work for organizations sized 10,000+.
- Respondents tend to work on one of three types of applications.
  - 73% work on web applications/services.
  - 51% work on enterprise business applications.
  - 36% work on native mobile applications.
- Respondents usually fill one of three main roles.
  - 34% work as developers/engineers.
  - 20% work as developer team leads.
  - 18% work as software architects.
- The organizations for whom our respondents work tend to use three main programming language ecosystems.
  - 74% use the Java ecosystem.
  - 66% use the Python ecosystem.
  - 64% use the client-side JavaScript ecosystem.

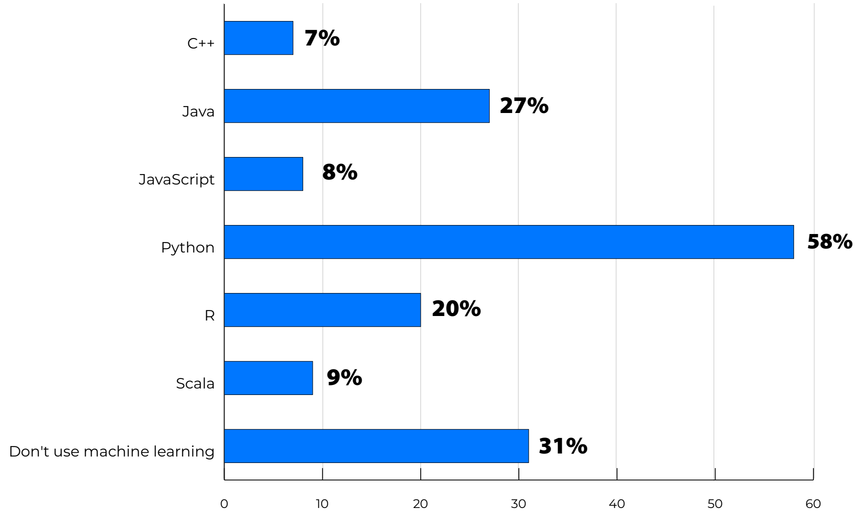
**TREND PREDICTIONS**

- Python will continue to grow in popularity for ML development.
- Oganizations will continues to ramp up their efforts around educating employees on ML.
- Expect to see the perecent of organizations developing ML funcationality increase.

## Programming Languages, Frameworks, and Tools

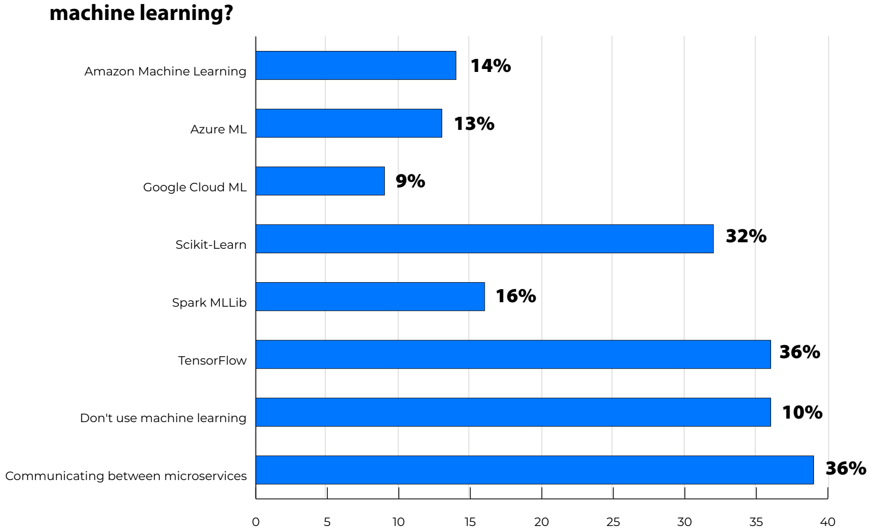
When developers and data scientists discuss machine learning, the Python language typically shows up somewhere in the conversation. With its easy to learn syntax and ecosystem full of powerful mathematical, scientific, and statistical libraries, Python has become the go-to language for machine learning. In our 2018 AI and Machine Learning Survey, 45% of respondents told us they used Python for machine learning development and 35% reported using Java. In this year's survey, when we asked respondents which languages their organization uses for machine learning development, Java usage rates fell to 27% while Python's climbed to 58%. In previous years, Python had a decent market share in the area of languages used for machine learning development. Increasingly, however, it seems Python is taking control of the field, becoming *the* language for machine learning development.

**Figure 1: Which languages does your organization use for machine learning development?**




Due to the quickly increasing dominance of Python in the field of machine learning and the development of ML software, libraries and frameworks that support Python have become popular among ML devs. While there are several paid options out there that have proven somewhat popular, two open source options have come to dominate: TensorFlow and scikit-learn. In 2018, 27% of respondents told us they used TensorFlow in their ML development efforts; in 2019, that number rose to 36%. Similarly, in 2018, 15% of respondents reported using scikit-learn; this year, we saw this usage rate more than double, with 32% of respondents telling us they use the scikit-learn library. This increase in scikit-learn usage rates is undoubtedly linked to the fact that it is a Python library. Thus, as Python goes, so goes scikit-learn. TensorFlow, however, is a different case. A massive platform for creating ML solutions in almost any language, Tensorflow's adoption and usage rates are more indicative of the general growth of machine learning in the industry.

**Figure 2: Which libraries or frameworks does your organization use for machine learning?**

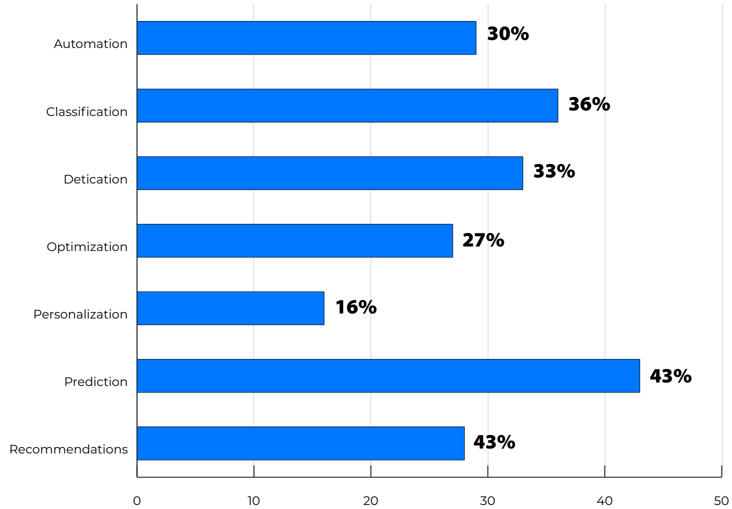



While machine learning SaaS platforms have yet to truly take off in the developer community, we did find some interesting trends around this technology. In this year's survey, 13% of respondents told us they use Google Prediction APIs, 11% said they use IBM Watson, and 10% reported using Microsoft Cognitive Services. In 2018, we also asked respondents which ML SaaS platform their organization used. 18% reported Google Prediction APIs, 10% said IBM Watson, and 8% said Microsoft Cognitive Serivces. While Google's ML SaaS has held steady, Microsoft's has nearly doubled its usage rates. As for the future of ML SaaS platforms, 17% of our 2019 survey respondents said their organization plans on using Google Prediction APIs over the next 12 months, 15% said they plan on using Microsoft Cognitive Services, and 13% plan to use IBM Watson.

## Machine Learning and the Enterprise

Even with the popularity of the tools and languages discussed above, for organizations to invest their time and money in machine learning it has to make business sense. To that end, we asked respondents why their organization uses machine learning. Chosen by 43% of respondents, prediction came in as the most popular answer. Predictive analyses using machine learning technology could cover anything from economic forecasts to fraud detection to predicting illness patterns in human populations. Thus, for the enterprise-level organizations for whom our respondents work, predictive analysis can be a great way to stay ahead of the competition or set trends in their field.

**Figure 3: Why does your organization use machine learning?**

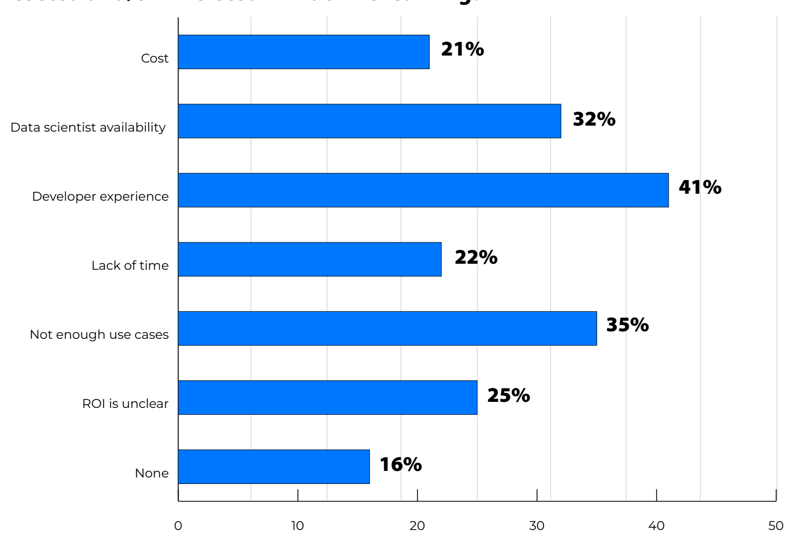



The second most popular response, classification, garnered 36% of responses. Classification is the process of identifying the class of a given piece of data, which, once identified, allows the software to process that data in an automated fashion. Lastly, detection was the other option chosen by over one-third of respondents (33%). In machine learning, detection is the software's ability to detect and classify physical objects around it. A few examples of how this technology is currently being put to use are self-driving cars and face detection in cameras. And, interestingly, of the three most popular uses for machine learning in respondents' organizations, detection exhibited the largest year-over-year growth, increasing from 23% in 2018.

To build the software for the above three use cases, organizations tend to rely fairly evenly on custom built ML functionality and machine learning APIs. 22% of respondents told us they generally use APIs, 21% reported they generally build custom ML software, and 19% said they use both about as equally as often. Given the difficulty in developing this type of software, and the data and AI specialists that need to be employed, it's interesting that the number of organizations using custom built machine learning is about equal to those that use API solutions.

Despite the apparently large number of organizations using machine learning, whether through custom built or API-based solutions, only 37% of respondents said that their organization is actively invested and engaged in one or more machine learning projects. Interestingly, the percentage of organizations engaged in ML projects grew 6% over last year, with 31% of respondents in 2018 telling us the organization invested in ML. In a similar vein, the percent of respondents who reported that their organizations are interested in adopting machine learning, but have yet to do so, dropped by 6%. In 2018, 33% of respondents said their organization fell into this category; this year, 27% reported thusly.

**Figure 4: What issues, if any, prevent your organization from being more intereseted and/or inversted in machine learning?**



## Addressing the Skill Gap

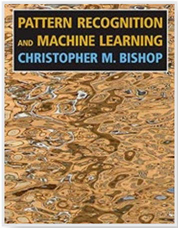
While it's clear that organizations currently have machine learning projects under development, or have plans to implement machine learning, many still find themselves facing a skill gap. When we asked survey takers what issues, if any, prevent their organizations from being more interested/invested in ML, 41% said developer experience. Additionally, when we asked what challenges their organizations face with regard to implementing machine learning, 38% reported developer training and 30% said there aren't enough data scientists in their organization. A skill gap has clearly appeared, but what are organizations doing to address it?

As more data science-related jobs (including machine learning) exist than data science professionals, organizations really have only one choice: educate and invest in their people. And this, indeed, seems to be what is happening. In our survey, we asked respondents what their organization is currently doing to pursue machine learning. 'Training its developers' came in as the top response, chosen by 42% of survey takers. Additionally, 28% of respondents told us that their organization is educating upper management about the benefits of machine learning. This willingness to educate all levels of the organization on the facets of ML they need to know in order to integrate machine learning into their role is a truly encouraging sign.

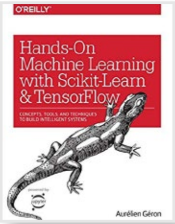
With a mind to the future, we also asked survey takers in which ways their organization plans on pursuing machine learning within the next 12 months. Again, their answers speak to the need to educate and invest in employees. 47% of respondents told us that over the next 12 months their organization plans on training its developers to better use machine learning. On top of that, 27% said their org will be educating upper management on ML over the coming 12 months.

# Diving Deeper Into Machine Learing

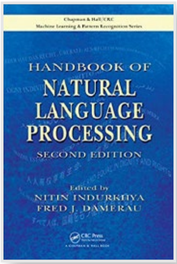
## Books




**Pattern Recognition and Machine Learning** This book presents statistical techniques for machine learning from graphical models to Bayesian methods.




**Hands-On Machine Learning With Scikit-Learn and TensorFlow** Get a practical understanding of everything you need to build and train intelligent systems.




**Handbook of Natural Language Processing** Learn how to develop and implement NLP in computational systems.

## Zones

**AI** The Artificial Intelligence (AI) Zone features all aspects of AI pertaining to machine learning, natural language processing, and cognitive computing. The AI Zone goes beyond the buzz and provides practical applications of chatbots, deep learning, knowledge engineering, and neural networks.

**Big Data** The Big Data Zone is a prime resource and community for big data professionals of all types. We're on top of all the best tips and news for Hadoop, R, and data visualization technologies. Not only that, but we also give you advice from data science experts on how to understand and present that data.

**IoT** The Internet of Things (IoT) Zone features all aspects of this multifaceted technology movement. Here you'll find information related to IoT, including Machine-to-Machine (M2M), real-time data, fog computing, haptics, open distributed computing, and other hot topics. The IoT Zone goes beyond home automation to include wearables, business-oriented technology, and more.

## Refcards

**Working With Time Series Data** Learn about the basics of time series data, how it can be used across industries, and using open source tools to manage it.

**Introduction to TensorFlow** This Refcard will help you understand how TensorFlow works, how to install it, and how to get started with in-depth examples.

**Getting Started With Scala** Covers creating a new Scala project, a tour of Scala's features, an introduction to classes and objects within Scala, and much more.

## Podcasts

**Talking Machines** Listen to experts get into deep (and exciting) conversations on emerging topics in AI and machine learning.

**Linear Digressions** Data scientists will enjoy this podcast on unusual applications for machine learning.

**Learning Machines 101** Get an introduction to a wide variety of ML and AI topics in a format that doesn't shy away from the highly technical.

ORIGINAL RESEARCH

# Machine Learning in the SDLC Research Report

By Tom Smith

To learn about the current and future state of machine learning (ML) in software development, we gathered insights from IT professionals from 16 different companies.

**TREND PREDICTIONS**

- Organizations will place greater emphasis on training and retaining machine learning specialists.
- Machine learning will increasingly be used to improve testing, automation, and software quality.
- We will see opportunities to build automated systems that analyze and use machine data to improve security, performance, and reliability.

Here's what we learned:

1. The most important elements of using ML to develop software are identifying the business problem you are trying to solve and ensuring ML is the appropriate solution. It's crucial to consider how ML can impact testing, automation, the importance of data, and the ability to improve time-to-value.

Understand what ML will bring you. You need the talent to define and understand how to solve the business problem you hope to address — as well as someone with the technical depth to determine the type of model you need and to be able to integrate it into DevOps processes for full implementation. Making a lot of models and determining their accuracy is a research project just in itself. If you are going to spend the time and money to deploy ML, you need to make sure the business outcomes are significant.

Automate deployment pipelines. Automate testing and improve test coverage. Decrease defects during release and optimize software performance. Use ML to detect failures and implement self-healing tests. Learn and understand what repetitive activities the developers are doing, and work to automate drudgery. This benefits anyone writing code, regardless of how it's written or operated.

The key theme with ML is time-to-value. The product must provide value to consumers, and it must be delivered on time with good quality. ML injects intelligent error handling, produces optimized algorithms, and improves deployment management — consequently improving developer productivity and improving time from the first code to release.

2. ML has improved testing, automation, and the quality of software developed. Automated test authoring, root cause analysis, greater coverage, and the ability to test on multiple devices and browsers have led to 100X better testing. Visual testing also enables the exposure of all the outcomes of functions. It takes a fraction of the time to offer and maintain new tests.

By implementing algorithms to self-correct code, ML significantly improves the identification and fixing of bugs, as well as improving and reducing the time required to fix and optimize code. Organizations can employ smart software development techniques to create outcome-driven processes. Such systems can auto-choreograph what needs to happen in the SDLC to meet objectives with a high degree of precision and accuracy.

3. ML tools and techniques most effective for software development involve the ability to version-control ML models, automate testing, and provide better feedback. Given the number of models you will be developing and testing, you need an ML data platform management system to track models with parameters and metrics.

Modern test automation tools enable self-healing tests, automated tests, automated crawlers that find bugs, and logging systems that find anomalies for security alerts. Tools simplify infrastructure and data engineering for developers.

With closed-loop functionality, smart agents can auto-detect and trigger re-learning and re-training processes to improve the performance and accuracy of models automatically. This leads to greater productivity, efficiency, and cost savings.

Tools simplify infrastructure and data engineering for developers.

4. There are many use cases for ML in software development — many more than the number of people who provided insights for this article. These use cases include anomaly detection, bug fixes, build monitoring, cash flow projection, data extraction, dynamic pricing, eye-tracking, and image recognition, to name a few. Likewise, there are a diversity of industries using ML in software, including automotive, financial services, insurance, logistics, professional services, and retail.

5. The most common issues with using ML in software development involve data, the transparency of the ML model, and the human skills needed. The most common issues center around poor data quality, having the right data, being able to access the data in a timely manner, and the time required to prepare the data. It typically takes a Fortune 500 company one month to get a data set to a data scientist. This is inefficient, and it hurts innovation.

Additionally, models tend to be a black box — not very transparent. A lack of transparency can make it difficult to provide definitive predictions on how well a model is going to perform in new environments.

Additionally, models tend to be a black box — not very transparent.

When building software with ML, companies need to realize it takes human capital with a specific skillset. It takes time to train people as well as to train a model. Retainment of data scientists is challenging, as well.

6. Like the issues now being seen, current major challenges revolve around data, bias, and skills. Data is not specific to software. It needs to be clean and governed to generate trusted insights. There are concerns around getting the right data to eliminate bias, taking organizational values into account. There are inherent biases in training set data, and the introduction of unintended bias into algorithms can skew outcomes. There is a need for trained and certified humans to train systems to pick the right algorithms with the right kind of gating measures.

7. The future is bright for the use of ML to develop higher quality software faster. We will see a lot of opportunities to build automated systems that analyze and act on machine data to improve the security, performance, and reliability of economically critical software services. There will be continued automation of bug fixes, testing, deployment, and code optimization. ML will continue to identify more tests that can be automated.

You will be able to deliver more, faster, with higher quality and less human involvement, by solving and automating small tasks done on a day-to-day basis to make intelligent decisions. The opportunity really lies in humans and machines working together intelligently, enabling developers to build new skills and freeing them to focus on what they are good at — while machines deal with the mundane.

8. Key for developers is to execute the fundamentals of software development, understand how ML differs from code and application development, and to always know the business problem to be solved. ML does not negate the need to adhere to traditional software design and development protocols. Maintain conscientious, diligent, and thoughtful software planning, design, development, and version control.

ML is the opposite of writing code. Models are constantly evolving based on training and data. Unlike code, you do not "set and forget" ML models. Understand ML models and their lack of predictability. Develop a close partnership with data scientists, and become sufficiently comfortable with data to understand what apps are doing with models.

## Over time, data scientists will become developers as they become more focused on scripting.

Developers and data scientists both need to understand the business problem to be solved. Over time, data scientists will become developers as they become more focused on scripting. Roles are getting merged, as statisticians learn Python and begin learning frameworks.

Figure out the guardrails to ensure you are getting to the right outcomes, ensure machine knowledge is transferable, build inclusivity into the outcomes, focus on optimization throughout the process, and know that feedback — collecting, interpreting, and incorporating it back into the data science — is where you will find real success.

Below is the list of executives and professionals who were kind enough to share their insights with us:

- Dipti Borkar, V.P. Products, Alluxio
- Adam Carmi, Co-founder and CTO, Applitools
- Dr. Oleg Sinyavskiy, Head of Research and Development, Brain Corp
- Eli Finkelshteyn, CEO & Co-founder, Constructor.io
- Senthil Kumar, VP of Software Engineering, FogHorn
- Ivaylo Bahtchevanov, Head of Data Science, ForgeRock
- John Seaton, Director of Data Science, Functionize
- Irina Farooq, Chief Product Officer, Kinetica
- Elif Tutuk, AVP Research, Qlik
- Shivani Govil, EVP Emerging Tech and Ecosystem, Sage
- Patrick Hubbard, Head Geek, SolarWinds
- Monte Zweben, CEO, Splice Machine
- Zach Bannor, Associate Consultant, SPR
- David Andrzejewski, Director of Engineering, Sumo Logic
- Oren Rubin, Evanglist, Testim.io
- Dan Rope, Director, Data Science and Michael O'Connell, Chief Analytics Officer, TIBCO