MACHINE LEARNING AS A SERVICE (MLAAS) EXPLAINED



Machine learning (ML) is a valuable tool, but it's complicated to use. Machine learning requires:

- Time to understand how to build a model
- The artistry to create an accurate model
- Resources to build and serve the model to a user

If this feels like a large barrier to entry, you're not wrong. It's no surprise that boutique shops are appearing to provide very specific Machine Learning as a Service (MLaaS) options, bringing ML to people and companies who might not otherwise be able to use it.

Let's take a look.

What is MLaaS?

Machine Learning as a Service provides machine learning operations, such as labeling data and predicting outcomes, to a customer.

Its main advantages include:

- Ease of use and maintaining the code
- Scalability
- Ease of monitoring usage and costs

How MLaaS works

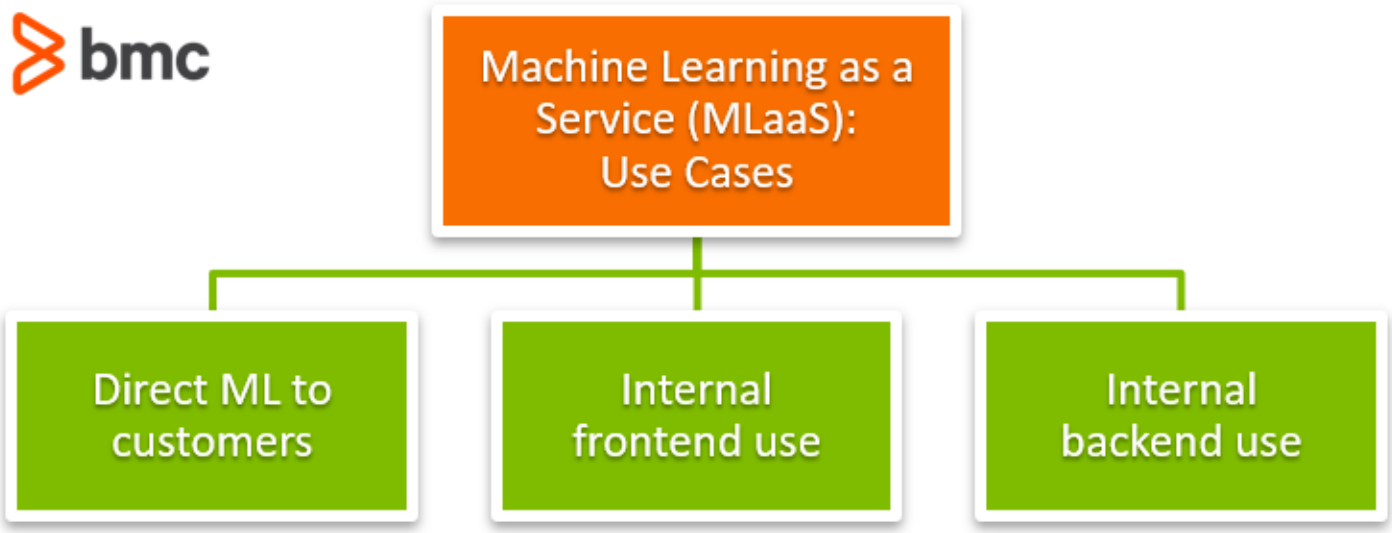
MLaaS is built on the same cloud architecture, comprised of containers and Kubernetes, that provides the foundation for other Functions as a Service (FaaS) and Software as a Service services (SaaS) solutions. Instead of offering a whole suite of tools as a single monolith of tools, a company may offer just one service: a finely tuned machine learning model.

MLaaS hangs out under the umbrella of microservices architecture, so customers use an API to access the machine learning model. The microservices architecture piecemeals services together, granting the company the capacity—the agility—to respond if one of their services becomes incredibly popular.

A whole suite of services doesn't need to move around the one. Instead, the single service can move independent of the others.

MLaaS use cases

MLaaS can be used to provide ML operations in three main areas:



Direct ML to customers

In this use case, customers can directly use the MLaaS. Most likely, the customer hopes some other company will do the hard work of creating the machine learning model. Then, the customer can use it for their purpose.

Internal frontend usage

Here, a frontend team could use an MLaaS model to power parts of an application.

For instance, if you've created a ride-sharing app, your team might rely on a ML model to predict the best route to take, given a traveler's location and destination. Your team might design the overall app, but not the ML model itself.

Internal backend usage

In this use case, teams can use MLaaS to understand their data more deeply.

ML vs MLaaS: which to use?

Machine learning isn't for every project. ML and the microservices architecture are fairly advanced, which could be overkill for some software projects. ML can open your app to offer many complex services, but its use is not always permitted.

Time, financial, or talent resources might come at too high a cost for you to implement the services. Finally, machine learning and microservices each have their own dependencies before they will be useful in your software ecosystem.

To know if traditional, robust machine learning is right for your software project, consider these must-haves for ML to have a chance at success.

ML requires lots of data

Your company must have access to lots (and lots) of data in order to even begin thinking about machine learning. A machine learning model trains on lots of examples of labeled data, so it can then make predictions.

If you have the data, you then have to deal with it: data needs to be stored, moved, and protected. If you're not ready for these tasks, ML might not be your best path forward.

ML requires talent

Beyond data, ML requires the ability of someone to:

1. Look at the data.
2. Ask good questions of it that can be of value to the user or customer.

For example, maybe there is a lot of user login data to know what times a user logs on. While that is information, and there can be plenty of data for it, the question needs to be asked,

"How can predicting the user's login behavior help create value to our service?"

Is the value internal, like knowing when to turn on servers? Or is it customer facing?

Timing for ML

Software starts as a monolith and gets split into a system of microservices. Adding MLaaS into your company's software will usually come later in the software's development.

Whether you choose to build using a microservice now or later probably depends on the project's time frame and its available resources.

The need for MLaaS

Creating a Machine Learning model is complicated. It requires resources and talent. Some companies know they want to get the benefits from using Machine Learning in their company but

might not want to have to deal with putting the resources together to build the team to make a model.

Instead, MLaaS can step in, trying to create models that could benefit a particular customer pool. These MLaaS companies can do the hard work of creating the model, training, and setting up the endpoint. Then, they can charge customers to use the model.

Creating a MLaaS

There are generally two paths to developing MLaaS for other users:

- As a company that provides the service
- On your own

Let's look at each.

Building a MLaaS company

Are you developing your own MLaaS? If you want to turn it into a business, it might work if your company has all these components:

- Plenty of financial resources
- Good talent
- Unique access to large datasets
- Datasets that can be beneficial to large amounts of business clients

Developing your own MLaaS

If you are a developer looking to build your own MLaaS, then your Machine Learning model can be hosted on both well-known and upstart companies.

MLaaS options & pricing

The prices and kinds of services fall in line with the variations that occur in all cloud computing services. You will pay different prices per minute per processor, GPUs, GBs stored, and region. Your region will determine available resources, latency speeds, accessibility, and market prices.

These are popular MLaaS options:

- **Amazon MLaaS** has a few options. SageMaker, for instance, has both a free tier and then varied prices for training and prediction.
- **Cloud Machine Learning Engine from Google** offers varied pricing
- **Azure ML from Microsoft** offers a free trial and varied pricing
- **BigML**, who aims to simplify ML, offers free usage as well as simple hourly plans.
- **PredictionIO**, under the Apache umbrella, is fully open source.

Each hosting platform will enable you to distribute your service through some kind of endpoint to your customers.

If you are the consumer of a MLaaS API, then all the hard work of modelling should be done for you, and you can stick to your operation and just figure out how to use the service through the service's

API portal. Connecting to the existing service is as easy as connecting to an API.

If machine learning isn't in your budget or your talent pool, MLaaS is a more affordable and accessible alternative.

Related reading

- BMC Machine Learning & Big Data Blog
- BMC Guides, where we offer tutorials on a variety of big data and data visualization tools
- Top Machine Learning Algorithms & How To Get Started
- What Is Machine Learning Operations? MLOps Explained