# Azure Machine Learning Services and Azure Databricks

As a consultant working almost exclusively in Microsoft Azure, developing and deploying artificial intelligent (AI) solutions to suit our client's needs is at the core of our business. Predictive solutions need to be easy to implement and must scale as it becomes business critical. Most organizations have existing applications and processes that they wish to infuse with AI. When deploying intelligence to integrate with existing applications it needs to be a microservice type feature that is easy to consume by the application. After trial and error I have grown to love implementing new features using both the Azure Machine Learning Service (AML Service) and Azure Databricks.

Azure Machine Learning Service is a platform that allows data scientists and data engineers to train, deploy, automate, and manage machine learning models at scale and in the cloud. Developers can build intelligent algorithms into applications and workflows using Python-based libraries. The AML Service is a framework that allows developers to train where ever they choose, then wrap their model as a web service in a docker container and deploy to any container orchestrator they wish!

Azure Databricks is a an optimized Apache Spark Platform for heavy analytics workloads. It was designed with the founders of Apache Spark, allowing for a natural integration with Azure services. Databricks makes the setup of Spark as easy as a few clicks allowing organizations to streamline development and provides an interactive workspace for collaboration between data scientists, data engineers, and business analysts. Developers can enable their business with familiar tools and a distributed processing platform to unlock their data's secrets.

While Azure Databricks is a great platform to deploy AI Solutions (batch and streaming), I will often use it as the compute for training machine learning models before deploying with the AML Service (web service).

## Ways to Implement AI

The most common ways to deploy a machine learning solution are as a :

* Consumable web service
* Scheduled batch process
* Continuously streaming predictions

Many organizations will start with smaller batch processes to support reporting needs, then as the need for application integration and near real-time predictions grow the solution turns into streaming or a web service.

### Web Service Implementation

A web service is simply code that can be invoked remotely to execute a specific task. In machine learning solutions, web services are a great way to deploy a predictive model that needs to be consumed by one or more applications. Web services allow for simply integration into new and existing applications.

A major advantage to deploying web services over both batch and streaming solutions is the ability to add near real-time intelligence without changing infrastructure or architecture. Web services allow developers to simply add a feature to their code without having to do a massive overhaul of the current processes because they simply need to add a new API call to bring those predictions to consumption.

One disadvantage is that predictions can only be made by calling the web service, therefore, if a developer wishes to have predictions made on a scheduled basis or continuously there needs to be an outside process to call that web service. However, if an individual is simply trying to make scheduled batch calls, I would recommend using Azure Databricks.

### Batch Processing

Batch processing is a technique to transform a dataset at one time, as opposed to individual data points. Typically this is a large amount of data that has been aggregated over a period of time. The main goal of batch processing is to efficiently work on a bigger window of data that consists of files or records. These processes are usually ran in "off" hours so that it does not impact business critical systems.

Batch processing is extremely effective at unlocking *\*deep insights\** in your data. It allows users to process a large window of data to analyze trends over time and really allow engineers to manipulate and transform data to solve business problems.

As common as batch processing is, there are a few disadvantages to implementing a batch process. Maintaining and debugging a batch process can sometimes be difficult. For anyone who has tried to debug a complex stored procedure in a Microsoft SQL Server will understand this difficulty. Another issue that can arise in today's cloud first world is the cost of implementing a solution. Batch solutions are great at saving money because the infrastructure required can spin up and shut down automatically since it only needs to be on when the process is running. However, the implementation and knowledge transfer of the solution can often be the first hurdle faced.

By thoughtfully designing and documenting these batch processes, organizations should be able to avoid any issues with these types of solutions.

### Stream Processing

Stream processing is the ability to analyze data as it flows from the data source (application, devices etc.) to a storage location (relational databases, data lakes etc.). Due to the continuous nature of these systems, large amounts of data is not required to be stored at one time and are focused on finding insights in small windows of time. Stream processing is ideal when you wish to track or detect events that are close in time and occur frequently.

The hardest part of implementing a streaming data solution is the ability to keep up with the input data rate. Meaning that the solution must be able to process data as fast or faster than the rate at which the data sources generate data. If the solution is unable to achieve this then it will lead to a never ending backlog of data and may run into storage or memory issues. Having a plan to access data after the stream is operated on and reduce the number of copies to optimize storage can be difficult.

While there are difficulties with a streaming data architecture, it enables engineers to unlock insights as they occur. Meaning, organizations can detect or predict if there is a problem faster than any other method of data processing. Streaming solutions truly enable predictive agility within an organization.

## Check out the Walkthrough

Implementing a machine learning solution with Azure Databricks and Azure Machine Learning allows data scientists to easily deploy the same model in several different environments. Azure Databricks is capable of making streaming predictions as data enters the system, as well as large batch processes. While these two ways are great for unlocking insights from your data, often the best way to incorporate intelligence into an application is by calling a web service. Azure Machine Learning service allows a data scientist to wrap up their model and easily deploy it to Azure Container Instance. From my experience this is the best and easiest way to integrate intelligence into existing applications and processes!

Check out the [walkthrough](https://github.com/ryanchynoweth44/AzureMachineLearningWithDatabricks) I created that shows engineers how to train a model on the Databricks platform and deploys that model to AML Service.