## Auto Machine Learning with Azure Machine Learning

I recently wrote a blog discussing automated machine learning (AutoML) libraries when developing predictive solutions. If you have not read it you can check it out [here](https://github.com/ryanchynoweth44/AutoMLExamples/blogs/AutoMachineLearning.md). With there being a surplus of AutoML libraries in the marketplace my goal is to provide quick overviews and demo of libraries that I use to develop solutions. In this blog I will focus on the benefits of the Azure Machine Learning Service (AML Service) and the AutoML capabilities it provides.

### Azure Machine Learning Service

An Azure Machine Learning Workspace (AML Workspace)is the foundation of developing and deploying predictive python-based web services in Azure. The AML Workspace allows data scientists to track their experiments, train and retrain their machine learning models, and deploy machine learning solutions as a containerized web service. When an engineer provisions an Azure Machine Learning Workspace the resources below are also created within the same resource group, and are the backbone to Azure Machine Learning.

* Azure Container Registry
* Azure Storage
* Azure Application Insights
* Azure Key Vault

The Azure Container Registry gives developer easy integration with creating, storing, and deploying our Python web services as Docker Containers. One added future is the easy and automatic tagging to describe your container and associate the container with specific machine learning models.

An Azure Storage account enables for fast dynamic storing of information from our experiments i.e. models, outputs. After training an initial model using the service, I would recommend manually navigating through the folders. Doing this will give you deeper insight into how the AML Workspace functions. But simply and automatically capture metadata and outputs from our training procedures is crucial to visibility and performance over time.

When we deploy a web service using the AML Service, we allow the Azure Machine Learning resource to handle all authentication and key generation code. This allows data scientists to focus on developing models instead of writing authentication code. Using Azure Key Vault, the AML Service allows for extremely secure web services that you can expose to external and internal customers.

Once your secure web service is deployed. Azure Machine Learning integrates seamlessly with Application Insights for all code logging and web service traffic giving users the ability to monitor the health of the deployed solution.

A key future to allowing data scientists to scale their solutions is offering remote compute targets. Remote compute gives developers the ability easily get their solution off their laptop and into Azure with a familiar IDE and workflow. The remote targets allow developers to only pay for the run time of the experiment, making it a low cost for entry in the cloud analytics space.

Azure Machine Learning provides data prep capabilities in the form of a "dprep" file allowing users to package up their data transforms into a single line of code. I am not a huge fan of the dprep but it is capability that makes it easier to handle the required data transformations to score new data in production. Like most platforms, the AML Service offers specialized "pipeline" capabilities to connect various machine learning phases with each other like data acquisition, data preparation, and model training.

In addition to remote compute, Azure Machine Learning enables users to deploy anywhere they can run docker. Theoretically, one could train a model locally and deploy a model local (or another cloud), and only simply use Azure to track their experiments for a cheap monthly rate. However, I would suggest tracking advantage of Azure Kubernetes Service for auto scaling of your web service to handle up ticks in traffic, or to a more consistent compute target in Azure Container Instance.

### Using Azure Machine Learning’s AutoML

Now its time to get to the actual point of this blog. Azure Machine Learning's AutoML capabilities. In order to use Azure Machine Learning's AutoML capabilities you will need to pip install “azureml-sdk”. This is the same Python library used to simply track your experiments in the cloud.

As with any data science project, it starts with data acquisition and exploration. In this phase of developing we are exploring our dataset and identifying desired feature columns to use to make predictions. Our goal here is to create a machine learning dataset to predict our label column.

Once we have created our machine learning dataset and identified if we going to implement a classification or a regression solution, we can let Azure Machine Learning do the rest of the work to identify the best feature column combination, algorithm, and hyper-parameters.

To automatically train a machine learning model using Azure ML the developer will need to: define the settings for the experiment then submit the experiment for model tuning. Once submitted, the library will iterate through different machine learning algorithms and hyperparameter settings, following your defined constraints. It chooses the best-fit model by optimizing an accuracy metric. The parameters or setting available to auto train machine learning models are:

* **iteration\_timeout\_minutes**: time limit for each iteration. Total runtime is iterations multiplied by iteration\_timeout\_minutes.
* **iterations**: Number of iterations. Each iteration produces a machine learning model.
* **primary\_metric**: metric to optimize. We will choose the best model based on this value.
* **preprocess**: When `True` the experiment may auto preprocess the input data with basic data manipulations.
* **verbosity**: Logging level.
* **n\_cross\_validations**: Number of cross validation splits when the validation data is not specified.

The output of this process is a dataset containing metadata the training runs and their results. This dataset enables developers to easily choose the best model based off the metrics provided. Being able to choose the best model out of many training iterations with different algorithms and feature columns automatically is that it enables us to easily automate the model selection process for \*each\* model deployment. With typical machine learning deployments, engineers typically deploy the same algorithm with the same feature columns each time. But with Auto Machine Learning solutions we are able to note only choose the best algorithm, feature combination, and hyper-parameters each time. That means, we can deploy a decision tree model trained on 4 columns one release, the deploy a logistic regression model trained on 5 columns another release without any code edits.

### My one compliant

My one compliant could be that I was just not doing it right. However, the documentation of this library notes that it works with Python 3.5.2 and up. I was unable to get the proper libraries installed and working correctly using a Python 3.6 interpreter. However, I simply created a Python 3.5.6 interpreter and it worked great!

Overall, I think Azure Machine Learning' Auto ML works great. It is not ground breaking or a game changer, but it does exactly as advertised which is huge in the current landscape of data where it seems as if many tools do not work as expected. Azure ML will run iterations over your dataset to figure out the best model possible, but in the end predictive solutions depend on the correlation between your data points. For a more detailed example of Azure Machine Learning's AutoML feature check out my walk through available [here](https://github.com/ryanchynoweth44/AutoMLExamples/AzureMLExample/walkthrough/01_EnvironmentSetup.md).