Riya Simon

CS 410

Nov 4, 2021

Technological Review

Introduction

The purpose of this technological review is to identify the main features of two widely used cloud-managed services that provide an end-to-end solution for deploying and training production-level machine learning models, as well as identify their shortcomings with respect to one another. I chose this topic because I often use AWS in my job and I felt it would be interesting to learn how these two services approach the problem of building models and what is the user's experience when using both these tools. As expected, both these services have a multitude of tools that address each part of the workflow - cleaning data, processing data, training models, deploying models, and evaluating models.

Overview of Sagemaker and Azure ML

AWS's offering, Sagemaker, includes several tools, one of which is a Jupyter notebook compute instance which allows the user to write code and explore the dataset. AWS installs all the packages included in the Anaconda installer and the TensorFlow and Apache MXNet deep learning libraries. In addition, it allows the user to customize the memory and CPU of the compute instance depending on the use case. For training the model, AWS offers a way to store training data in S3 (object storage service), to train the model using built-in algorithms, save the output of the job in S3 and store the code in the form of a docker image on Elastic container registry. The image is helpful in order to repeat the process with different datasets. One of the available out-of-the-box algorithms related to textual analysis is Neural Topic Algorithm, used to organize a corpus of documents into topics that contain word groupings based on their statistical distribution. The user configures "Experiments" which allows them to group and organize their machine learning iterations. The experiment consists of several trials, each of which contains trial components. Each trial component can include a combination of inputs such as datasets, algorithms, and parameters, and produce specific outputs such as models, metrics, datasets, and checkpoints. Using this setup, the user can explore several combinations of features and model heuristics to obtain the best model. The last step is to deploy the model for future predictions. SageMaker provides model hosting services for model deployment and provides an HTTPS endpoint where the user can pass data in the body and get back the results from the stored model. This is one typical workflow for building an application but there are different customizations available for each step, for example, the user can use Apache Spark or custom code for training the model.

Microsoft's offering Azure Machine Learning has a similar approach to this problem and has many of the same components that Sagemaker has. One major difference that Azure ML provides is the ML Studio, which is a web portal that provides a no-code experience when building a pipeline. The service still provides Jupyter Notebook servers to developers who are comfortable with coding. However, people with less experience can take advantage of the studio to organize the workflow using the drag and drop interface. There is a palette of datasets and components for the user to choose from. Some of the data processing components include "Cleaning missing data", "Selecting data", "Split Data". There are components to select from a variety of ML algorithms to train the model. The final component is "Score the model" which allows the user to run the model on the test dataset and view the results. In terms of deployment, Azure has a similar solution which is to deploy the model to an Azure Kubernetes cluster and register an endpoint, which can be used to post data and retrieve the results. Apart from the studio, Azure ML has a lot of tools with a similar purpose to the Sagemaker's offerings. There is a container registry for storing artifacts, a data storage for datasets, and a key vault for storing sensitive data used by the code. There are configurable compute instances for training and inference jobs similar to Sagmaker's resources.

Comparison

After investigating the benefits and unique features of both these services, there are some features that are more beneficial depending on the user's level of expertise. One of the standout features is Azure's drag and drop interface vs the more technical interface of Sagemaker. Azure caters to a larger proportion of the population which Sagemaker targets those who are comfortable with the technical aspects of data engineering. Sagemaker provides more freedom and flexibility in creating models. It is a good choice if the problem space is complex and needs customized solutions which need to be designed in an iterative fashion. There is a larger variety of selection of supported algorithms as well for NLP and Computer Vision.

Conclusion

Both platforms provide a well-developed and mature solution for this problem but they differ in how they provide the user experience. One caters to users who might be interested in exploring the data and gaining actionable insights while the other is designed to provide a more configurable platform to accommodate various complex use cases. Nevertheless, both platforms provide an end-to-end solution for professionals seeking a production-ready service for their ML needs.

References

Sdgilley. (n.d.). *Architecture & Key Concepts - Azure Machine Learning*. Architecture & key concepts - Azure Machine Learning | Microsoft Docs. Retrieved November 6, 2021, from https://docs.microsoft.com/en-us/azure/machine-learning/concept-azure-machine-learning-architecture.

Mishra, Abhishek. “Machine Learning in the AWS Cloud: Add Intellegence to Applications with Amazon Sagemaker and Amazon Rekognition.” *Amazon*, Sybex, a Wiley Brand, 2019, https://docs.aws.amazon.com/sagemaker/latest/dg/how-it-works-training.html.

Engdahl, Sylvia. “Blogs.” *Amazon*, Greenhaven Press/Gale, 2008, https://aws.amazon.com/blogs/machine-learning/building-a-customized-recommender-system-in-amazon-sagemaker/.

Peter Lu. “Tutorial: Designer - Train a No-Code Regression Model - Azure Machine Learning.” *Azure Machine Learning | Microsoft Docs*, https://docs.microsoft.com/en-us/azure/machine-learning/tutorial-designer-automobile-price-train-score.