Model Evaluation Service: Improving Machine Learning Model Development Efficiency

CS 4991 Capstone Report, 2022

Alex Kwong
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
ask8kb@virginia.edu

Abstract

The Softlines Fit team within the Amazon Fashion organization leverages machine learning models and existing customer data to provide accurate size recommendations to customers buying clothing and apparel on Amazon.com. However, machine learning engineers are currently evaluating their models manually and individually which is inefficient. The proposed solution to is the Model Evaluation Service (MES) with the goal of giving engineers a standardized and more efficient way of evaluating models.

As an intern, I used the existing process of model evaluation as a baseline of what needed to be standardized and automated. I then automated the process by using Apache Spark for data processing and Amazon Quicksight for visualizations and analyses. I was able to resolve a longstanding cyclic dependency when programmatically creating Quicksight analyses. I also discovered limitations of Quicksight, such as the inability to customize visualizations and fully automate analysis creation.

The most urgent change needed is switching off Quicksight for analysis and visualization generation. Another feature for future development would be a user interface or web application for the Model Evaluation Service, enabling users to more easily access its features.

1. Introduction

Machine learning is integral to the operations of Amazon from catalog recommendations on Amazon.com to natural language processing in Alexa. My team required machine learning to provide customers with accurate size recommendations for clothing and apparel. In order to provide and improve these recommendations, machine learning engineers must tune models using existing customer data and evaluate their performance.

The issue is that machine learning engineers are evaluating their models individually because there is no standardized or centralized system for model evaluation. Currently, if a model needs to be evaluated, an engineer is required to run the model on their own local instance. This requires access to significant amounts of computational resources which are available in a system that leverages cloud computing services. Additionally, if multiple engineers are working on the same model, they may redundantly evaluate the model unknowingly because they lack a centralized system.

By implementing a centralized system to evaluate models, machine learning engineers

are provided with a standardized way of evaluating their models. This reduces the number of times models are evaluated and increases development efficiency overall.

2. Related Works

Due to the nature of the research problem being implementation-heavy rather than research heavy, related work is less prevalent for this issue.

Hart et al. (2011) [1] discusses a similar problem and solution. Instead of working with models which provide recommendations, they are working with climate models. However, the issue is the same in both cases where local evaluation of models is highly inefficient in which the solution is to leverage cloud services for large scale data processing and provide a centralized way to evaluate models. They describe a software architectural approach that leverages cloud computing to improve model evaluation.

Zaharia et al. (2016) [2] discusses Apache Spark, an engine used for large-scale data analytics, which was used in the Model Evaluation Service. Apache Spark leverages distributed data processing in order to handle large amounts of data. This is required by the Model Evaluation Service as millions of data points must be processed in order to evaluate models.

3. System Design

The Model Evaluation Service leverages multiple AWS cloud services to automate the process of model evaluation. The AWS services included in the design are S3, Glue, Lambda, and Quicksight.

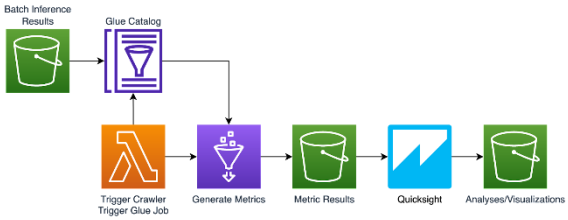

Figure 1: Model Evaluation Service Design

AWS S3 is a storage service which provides the MES a place to save and retrieve data to and from. The MES pipeline begins with an S3 bucket which stores batch inference results from evaluated models.

AWS Glue is a data integration service which enables data to be prepared, processed, and more for analytics, machine learning, and application development. The batch inference data from S3 is imported into a Glue catalog which allows for the data to be processed.

AWS Lambda is an event-driven computing platform. A Lambda layer is utilized to trigger a Glue job which generates the model performance metrics. Depending on user input, this layer selects which metrics to generate, such as accuracy, coverage, and distance from ground truth. Once the Glue job has completed, the results are saved back to S3.

Amazon QuickSight is a business analytics service that builds visualizations and performs analyses on demand. The metrics generated by the Glue job are imported into QuickSight which then automatically generates the visualizations engineers were previously generating manually. Once generated, the visualizations are saved and exported to S3.

4. Results

The end of the internship resulted in a basic implementation of the MES. The system is able to automatically import batch inference data into a Glue catalog, generate accuracy and coverage metrics, and update QuickSight visualizations to reflect the new results. However, the work done so far is only the groundwork for the service and many changes will need to be made because of limitations discovered during the internship.

For example, Amazon QuickSight was originally chosen for data analysis and visualization because that was the standard method being used by engineers. However, we discovered that using the QuickSight API to automate certain processes worked differently from how they would be done manually. For example, new analyses cannot be created on an ad-hoc basis because of a cyclic dependency. Also, visualizations cannot be created or adjusted programmatically which results in the need for a one-time setup for each visualization. As a result, we concluded that an alternative to QuickSight should be used.

5. Conclusion

The Model Evaluation Service provides a way to more efficiently evaluate models, thus aiding the development of machine learning models for the Softlines Fit team. It does so by leveraging Amazon cloud services to create a centralized and standardized platform for engineers to evaluate and compare models. The MES has potential for company-wide adoption as it has utilization anywhere that leverages machine learning.

6. Future Work

The end of the internship revealed the drawbacks of using Amazon QuickSight for data analyses and visualization. The alternative should allow for on-demand analysis and visualization creation as well as increased options for customization.

For the purposes of creating a working demo, only default metrics such as accuracy and coverage were provided. However, users should have the ability to onboard their own metric generators, allowing any metric to be visualized rather than the default metrics provided.

Another future goal is to implement a user interface or web application to enable users to interact with the MES more easily. This application will not only allow users to start and generate evaluation metrics but also provide a place to view past results to compare performance between models.

References

[1] Hart, A., Goodale, C., Mattmann, C., Zimdars, P., Crichton, D., Lean, P., Kim, J. and Waliser, D. 2011. A cloud-enabled regional climate model evaluation system. In Proceedings of the 2nd International Workshop on Software Engineering for Cloud Computing (SECLOUD '11). Association for Computing Machinery, New York, NY, USA, 43–49. https://doi-org.proxy01.its.virginia.edu/10.1145/1985500.1985508

[2] Zaharia, M., Xin, R., Wendell, P., Das, T., Armbrust, M., Dave, A, Meng, X. Rosen, J., Venkataraman, S., Franklin, M., Ghodsi, A., Gonzalez, J., Shenker, S. and Stoica, I. 2016. Apache Spark: a unified engine for big data processing. Commun. ACM 59, 11 (November 2016), 56–65. https://doi-org.proxy01.its.virginia.edu/10.1145/2934664