Deploying a Collaborative Framework for Crowd Sourcing the Evaluation of AI Model Effectiveness

Sarah Packowski
spackows@ca.ibm.com
IBM

Joshua Allard
jmallard@us.ibm.com
IBM

ABSTRACT

Evaluating the effectiveness of a binary classification model can be as simple as calculating the percent of inputs that are correctly classified by the model.

But when it comes to evaluating the effectiveness of a speech to text model or natural language understanding model, simply counting the number of incorrect words (for example) doesn't capture the nuances required to understand if the model is effective enough to do the job you need it to do. One way to tackle this problem is to experiment with multiple evaluation methods to see what fits your needs best.

In this workshop, participants deployed a sample Python Flask app in IBM Cloud that enables teams to dynamically collect and apply multiple model evaluation methods contributed by collaborators.

In this workshop, participants discussed common challenges with developing collaborative AI or data science solutions, including:

- Streamlining the workflow so specialists can focus on their area of expertise
- Making it easy for all team members to understand all solution components
- Scaling out the assembled solution
- Turning the solution into a platform by exposing endpoints

CCS CONCEPTS

• **Software and its engineering → Programming teams**; **Software prototyping**; • **Computing methodologies → Natural language processing**.

KEYWORDS

collaboration, prototyping, continuous delivery, AI, effectiveness

ACM Reference Format:
Sarah Packowski and Joshua Allard. 2020. Deploying a Collaborative Framework for Crowd Sourcing the Evaluation of AI Model Effectiveness. In *Proceedings of CASCON '20: Proceedings of 29th Annual International Conference on Computer Science and Software Engineering, Markham, (CASCON '20).* ACM, New York, NY, USA, 1 page.

CASCON '20, November, 2020, Toronto, On

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00

1 RATIONALE

Imagine two speech to text models produce the following output for the spoken phrase *the quick, brown fox jumped over the lazy dog*:

Model A result
`the quick brown fox jumped under the lazy dog`
Model B result
`the quick brown fox jumped over the lazy frog`

Both **Model A** and **Model B** got one word wrong. So, how would you decide which model performed better? The answer would vary, depending on your use case:

- If you are analyzing the transcript to determine *what actions happened*, the result from **Model A** would be more "wrong" (jumping under instead of jumping over)
- If you are analyzing the transcript to determine *who was involved*, the result from **Model B** would be more "wrong" (frog instead of dog)

To assess AI model performance for different use cases like this, our team developed a simple framework that applied multiple evaluation methods simultaneously so we could easily see which methods fit our needs best. The framework was deployed to IBM Cloud using a GitHub-integrated continuous delivery pipeline. Team members implemented each evaluation method in its own Python file; and then when team members uploaded their files to GitHub, the framework was automatically redeployed with their new methods included.

Using this simple framework and continuous delivery pipeline has had several advantages:

- The simplicity of the framework made it easy to get started
- Each team member could focus on their own implementation, without worrying about deployment or app management
- We have reused the framework for multiple, different sorts of projects with only minor adjustments

2 WORKSHOP FORMAT

In this workshop, participants gained hands-on experience deploying a Python Flask app to IBM Cloud, using a continuous delivery pipeline with GitHub.

The direct experience in this workshop, as well as the group discussions about how to enable both generalist and specialist team members to contribute to a solution, gave participants strategies for future collaborative AI or data science projects.