Post-Training Evaluation with Binder

Jessica Forde, Chris Holdgraf, Yuvi Panda, Aaron Culich, Matthias Bussonnier, Min Ragan-Kelley, M Pacer, Carol Willing, Tim Head, Fernando Perez, Brian Granger, Project Jupyter Contributors[1]

ABSTRACT

'Black box' models are increasingly prevalent in our world and have important societal impacts, but are often difficult to scrutinize or evaluate for bias. Binder provides anyone in the community the opportunity to examine a machine learning pipeline, promoting fairness, accountability, and transparency. Binder is used to create custom computing environments that can be shared and used by many remote users, enabling the user to build and register a Docker image from a repository and connect with JupyterHub. Users can select a specific branch name, commit, or tag to serve. Binder combines two projects: JupyterHub, which provides a scalable system for authenticating users and launching Jupyter Notebook servers, and repo2docker, which generates a Docker image from a Git repository. When connected with JupyterLab, users can navigate a repository on Binder with an IDE as if they were developing the project locally and can explore all underlying data (CSV, JSON, image, etc.). JupyterHub, repo2docker, and JupyterLab work together on Binder to allow a user to evaluate a machine learning pipeline with much greater transparency than a typical publication or GitHub page. Together, these three projects promote fairness, accountability, and transparency in machine learning.

1 INTRODUCTION

Trained models are based on computation. To fully understand a model and its predictions, we need access to the underlying code, data, and computational resources used to train that model. Assembling these elements is frequently challenging and requires considerable effort and technical expertise. Simply recreating the environment needed to run another researcher's analyses can take days.

Binder [1] allows researchers to rapidly recreate the computational environment needed to interact with research code and data shared online. It follows in the tradition of open science platforms like GitHub, the Open Science Framework [2], and CodaLab [3]. To interact with someone else's work, a user clicks on a URL and is taken directly to a live environment than can run the code in the cloud. When Binder was first announced in May 2016 [4], it connected GitHub repositories to universally accessible computational environments. The Binder service was initially launched out of Jeremy Freeman's lab at Janelia Farm, the HHMI Research Campus. It quickly gained traction in the scientific computing community [5] as a way to share interactive Jupyter notebooks, complementing the static rendering provided by nbviewer. Binder is now powered by BinderHub [6], which uses JupyterHub [7] technology to handle its services. This makes it more scalable, more stable, and more flexible. Binder also allows users to use JupyterLab [8], an IDE built as a follow-up of the Jupyter Notebook [9]. JupyterLab allows the user to navigate a repository in a more traditional developer environment, with improved tools for inspecting data and source code.

There have been calls for improved model transparency and accountability due to the increasing adoption of powerful 'black box' algorithms, which are being used to make important societal decisions [10] despite the difficulties of interpreting these models [11] and reproducing their results [12]. Binder can be used with JupyterLab to create an open source software stack that allows for greater scrutiny of machine learning pipelines and pre-trained models. We examine each component of the stack (JupyterHub, repo2docker [13], and JupyterLab) individually and discuss how each piece is used to encourage greater scientific communication through software. Together, we believe they provide machine learning researchers, data scientists, data journalists, and non-practitioners greater access to models, software and data that allow critiques of methods and evaluations of model fairness.

2 OPEN-SOURCE BUILDING BLOCKS

Project Jupyter provides open-source building blocks for interactive and exploratory computing that make science reproducible across over 40 programming languages (Python, Julia, R, etc.)[14]. Central to the project is the Jupyter Notebook, a web-based interactive computing platform that allows users to author "computational narratives" that combine live code, equations, narrative text, visualizations, interactive dashboards and other media. JupyterLab goes beyond the classic Jupyter

[1]Please contact the authors through Binder's issues page https://github.com/jupyterhub/binder/issues

Notebook and provides a flexible and highly extensible web-based IDE with powerful tools for data exploration and collaboration.

To fully understand a model, we need to inspect the data, run the underlying source code, and obtain model predictions. This is demonstrated by examining and reproducing the experiments of Ross et al. [15], which use a novel loss function as a competitor to LIME [16]. The corresponding binder can be accessed at https://mybinder.org/v2/gh/dtak/rrr/master?urlpath=lab. In this paper, the authors have provided a `rrr` folder which contains the source code of the model, a `data` folder for input data and model parameters , an `experiments` folder which contains Jupyter Notebooks implementing experiments presented in the paper on standard datasets, and a `bin` folder with Python scripts for experiments. Because of the IDE-like interface of JupyterLab, one can view the notebooks side by side with the imported source code to better understand the experiments. In addition, the user can modify the source code to examine or alter the experiment. For example, we modify the code in one of the notebooks to view a random sample of the data instead of the data immediately presented, as shown in Figure 1. Alternatively, we can replicate the "Toy Colors" experiment with 5 colors instead of 4 simply by changing a single line in the source code and running the "Toy Colors" notebook, as shown in Figure 2. Note that the figures show a modification to the binder performed online and that Binder does not save the changes to the files. To permanently change the files within a Binder instance, one would need to create a new commit to the repo and then rebuild the Binder image.

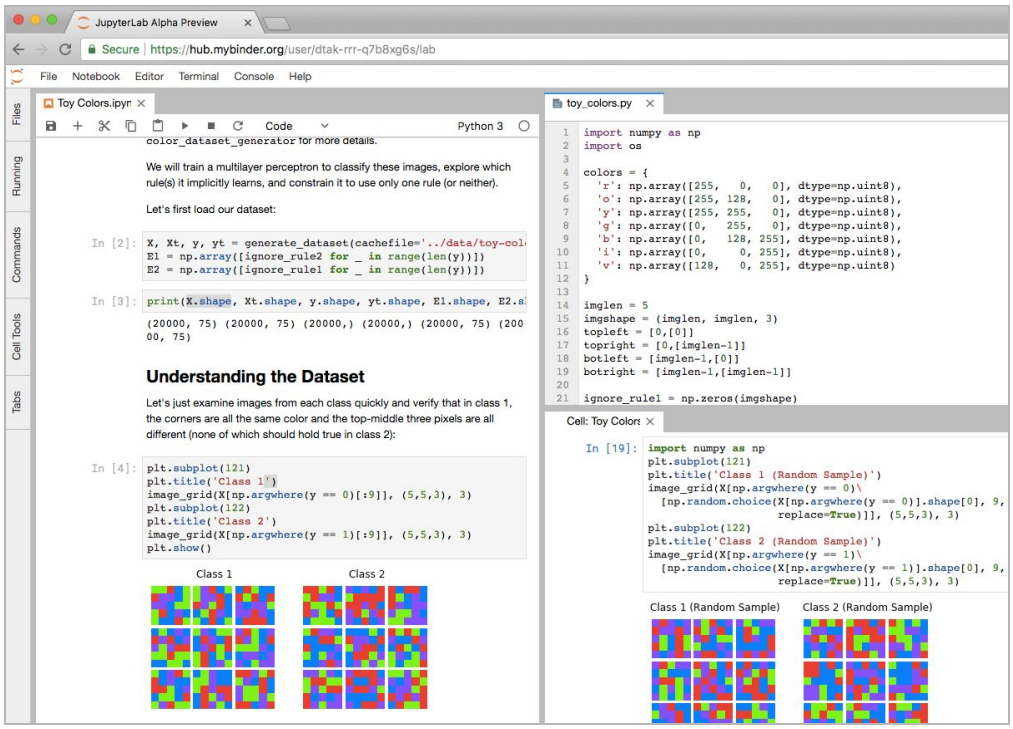

Figure 1: Examining image data from Ross et al. on Binder with JupyterLab. By dragging and dropping text files and notebooks, one can reference the source data on the top right when viewing the Jupyter Notebook on the left to understand how the toy images were generated. On the bottom right, we modify the authors' code to view a random sample of the input images.

JupyterHub is a multi-user hub that launches, manages, and proxies multiple instances of the single-user Jupyter notebook server. On Binder, we use JupyterHub to launch instances of JupyterLab or Jupyter Notebook with identical environment configuration and files, which allows users to have immediate access to a fully functional environment with zero local configuration. In the example from Ross et al., we are able to make these changes and run them immediately on Binder. In Figures 1 and 2, we see Binder's URL in place of a local host location as we modify the authors' experiment. Additionally, users of Binder have equal access to computational power; no user is prioritized over another.

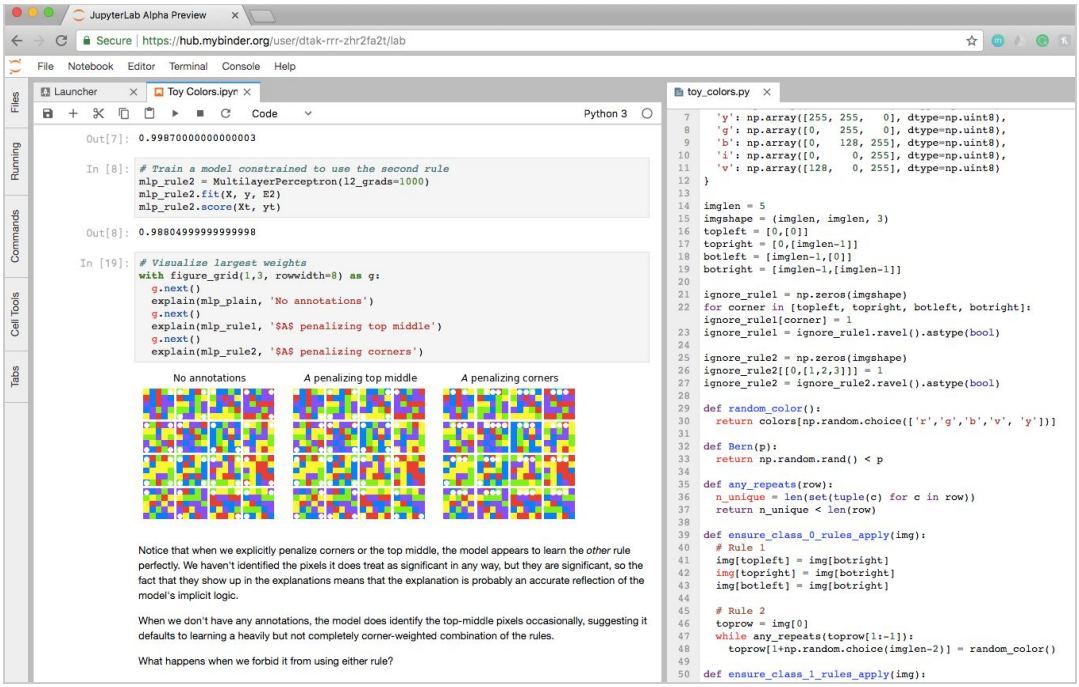

Figure 2: Modifying an experiment in the cloud with Binder. On the right panel on line 30, we modified the code to include yellow in addition to original colors used in the Toy Color notebook. The changes in the notebook are shown on the left. The address bar of the browser indicates that the these changes were made on the binder instance and not a local machine.

Binder uses repo2docker to build a Docker image from a repository based on configuration files such as `apt.txt`, `postBuild`, or `environment.yml`. This example repo uses an `apt.text` file to install LaTeX for plot figures and `environment.yml` to indicate which Python dependencies are available through pip and conda. In addition to configuring the JupyterHub instance, one can use repo2docker to replicate a GitHub repo's environment on another cloud service or cluster. Additional features that allow users run a single repo on a cloud instance with repo2docker are in development. Should a user want to configure their Binder instance to a particular repo branch or commit, the form on Binder's homepage allows the user to specify these details before launching the instance.

3 TECHNICAL REQUIREMENTS

To share machine learning pipelines with Binder, an environment configuration file such as `requirements.txt` is necessary. An example repo with a typical `environment.yml` can be found at: https://github.com/binder-examples/conda. Additional files must be included to facilitate model evaluation. First, a saved copy of the trained model allows users to easily predict from the model without retraining. Next, docstrings and other forms of documentation (readme, instructions for reproducing specific experiments) allow users to more easy understand code in the repo. Files describing the hardware configuration used to train the saved model inform users as to the hardware required to recreate results. The presence of these essential elements constitutes a metric for a repo's level of reproducibility [12]. At Project Jupyter, we have been evaluating the reproducibility of published work based on the presence of these elements in their repositories. We are populating our collection of reproduced scientific repos on Binder on https://github.com/binder-examples/ and are highlighting notable examples of repos that use Binder to share research on https://mybinder.readthedocs.io/en/latest/. These efforts dovetail with broader efforts in the machine learning community to create reproducible machine learning results [17,18]. While Binder requires diligence on the part of the researcher to carefully document their work for reproducibility, Binder users need only a computer with an internet connection and a browser to use the software of a repo. No other installation is necessary.

REFERENCES

1. Project Jupyter Contributors. Introducing Binder 2.0 — share your interactive research environment. In: eLife [Internet]. [cited 11 Dec 2017]. Available: https://elifesciences.org/labs/8653a61d/introducing-binder-2-0-share-your-interactive-research-environment

2. Erin D. Foster AD. Open Science Framework (OSF). J Med Libr Assoc. Medical Library Association; 2017;105: 203.

3. Liang P, Viegas E. CodaLab Worksheets for Reproducible, Executable Papers [Internet]. NIPS 2015, Demonstrations Track; 2015 Dec 9; Montreal, Quebec Canada. Available: https://nips.cc/Conferences/2015/Schedule?showEvent=5779

4. Freeman J, Osheroff A. Toward publishing reproducible computation with Binder. In: eLife [Internet]. 13 May 2016 [cited 11 Dec 2017]. Available: https://elifesciences.org/labs/a7d53a88/toward-publishing-reproducible-computation-with-binder

5. LIGO Scientific Collaboration. LIGO Open Science Center. In: LIGO [Internet]. [cited 12 Dec 2017]. Available: https://losc.ligo.org/tutorials/

6. Project Jupyter Contributors. BinderHub [Internet]. Available: https://github.com/jupyterhub/binderhub

7. Project Jupyter Contributors. JupyterHub [Internet]. Available: https://github.com/jupyterhub/jupyterhub

8. Project Jupyter Contributors. JupyterLab [Internet]. Available: https://github.com/jupyterlab/jupyterlab

9. Project Jupyter Contributors. Project Jupyter [Internet]. [cited 12 Dec 2017]. Available: https://jupyter.org/

10. Crawford K, Calo R. There is a blind spot in AI research. Nature. 2016;538: 311–313.

11. Doshi-Velez F, Kim B. Towards A Rigorous Science of Interpretable Machine Learning [Internet]. arXiv [stat.ML]. 2017. Available: http://arxiv.org/abs/1702.08608

12. Pineau J. Reproducibility in Deep Reinforcement Learning and Beyond [Internet]. Deep Reinforcement Learning Symposium, NIPS 2017; 2017 Dec 7; Long Beach, California. Available: https://twitter.com/xtimv/status/938917013086380032

13. Project Jupyter Contributors. Repo2Docker [Internet]. Available: https://github.com/jupyter/repo2docker

14. Project Jupyter Contributors. Jupyter kernels [Internet]. Project Jupyter; Available: https://github.com/jupyter/jupyter/wiki/Jupyter-kernels

15. Ross AS, Hughes MC, Doshi-Velez F. Right for the Right Reasons: Training Differentiable Models by Constraining their Explanations. Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence. 2017. p. Pages 2662–2670.

16. Ribeiro MT, Singh S, Guestrin C. Why Should I Trust You?: Explaining the Predictions of Any Classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM; 2016. pp. 1135–1144.

17. Gershgorn D. The titans of AI are getting their work double-checked by students. In: Quartz [Internet]. Quartz; 4 Nov 2017 [cited 11 Dec 2017]. Available: https://qz.com/1118671/the-titans-of-ai-are-getting-their-work-double-checked-by-students/

18. Henderson P, Islam R, Bachman P, Pineau J, Precup D, Meger D. Deep Reinforcement Learning that Matters [Internet]. arXiv [cs.LG]. 2017. Available: http://arxiv.org/abs/1709.06560