Towards Understanding Machine Learning Testing in Practise

Shome, Arumoy; Cruz, Luís; Van Deursen, Arie

DOI
10.1109/CAIN58948.2023.00028

Publication date
2023

Document Version
Final published version

Published in
Proceedings - 2023 IEEE/ACM 2nd International Conference on AI Engineering - Software Engineering for AI, CAIN 2023

2023 IEEE/ACM 2nd International Conference on AI Engineering – Software Engineering for AI (CAIN) | 979-8-3503-0113-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/CAIN58948.2023.00028

Towards Understanding Machine Learning Testing in Practise

Arumoy Shome
Software Engineering Research Group
Delft University of Technology
Delft, Netherlands
a.shome@tudelft.nl

Luís Cruz
Software Engineering Research Group
Delft University of Technology
Delft, Netherlands
l.cruz@tudelft.nl

Arie van Deursen
Software Engineering Research Group
Delft University of Technology
Delft, Netherlands
arie.vandeursen@tudelft.nl

***Abstract*—Visualisations drive all aspects of the Machine Learning (ML) Development Cycle but remain a vastly untapped resource by the research community. ML testing is a highly interactive and cognitive process which demands a human-in-the-loop approach. Besides writing tests for the code base, bulk of the evaluation requires application of domain expertise to generate and interpret visualisations. To gain a deeper insight into the process of testing ML systems, we propose to study visualisations of ML pipelines by mining Jupyter notebooks. We propose a two prong approach in conducting the analysis. First, gather general insights and trends using a qualitative study of a smaller sample of notebooks. And then use the knowledge gained from the qualitative study to design an empirical study using a larger sample of notebooks. Computational notebooks provide a rich source of information in three formats—text, code and images. We hope to utilise existing work in image analysis and Natural Language Processing for text and code, to analyse the information present in notebooks. We hope to gain a new perspective into program comprehension and debugging in the context of ML testing.**

***Index Terms*—AI Engineering, Machine Learning Testing, Data Mining, Computational Notebooks, Image Analysis, Natural Language Processing, NLP for Code**

I. Introduction

Visualisations are the "bread and butter" of a data scientist since they drive all aspects of the Machine Learning Development Cycle. Visualisations are used as early as the data exploration phase to understand the underlying dataset and gain insights prior to training. During the training phase, visualisations are used to evaluate and compare the predictive performance of various ML models. Once a model is selected, visualisations aid practitioners in testing black-box ML systems for non-functional properties such as fairness and explainability [1]. Finally, once a model is deployed, visualisations are used to continually monitor its health and automatically trigger a new training cycle if the performance degrades due to data drifts [2].

Computational Notebooks have become wide-adopted by the data science community to develop ML pipelines. Computational notebooks are a perfect fit for the data science workflow as they allow practitioners to interweave text, code and visualisations in a single cohesive document. Although there is an abundance of publicly available computational notebooks, they still remain a vastly untapped resource by the research community.

Testing ML systems is a highly interactive and cognitive process which demands a human-in-the-loop approach. In addition to writing tests for the code base, bulk of the evaluation requires application of domain expertise to generate and interpret visualisations. ML testing is a relatively new field of research. As such, many of the contributions work in an experimental setting, however their feasibility in a more practical environment remains unclear.

We propose a novel approach to understanding the current challenges of ML testing in practise. To gain a deeper insight into the process of ML testing itself, we propose to study the visualisations generated for ML pipelines by mining Jupyter notebooks in the wild.

II. Methodology

We propose to collect Jupyter notebooks from *Kaggle*—a popular online repository for data science and ML computational notebooks[1]. Kaggle provides a stable API to download notebooks based on filters specified by the user.

This allows us to adopt different search strategies based on the goal of the study. To gain a general perspective on ML testing, we can mine notebooks from popular Kaggle competitions. Alternatively, we may choose to focus on a single functional or non-functional test property. For instance, to understand how practitioners test for fairness in ML systems, we can collect notebooks that are associated with data science competitions focusing on fairness or datasets that have been cited by prior scientific contributions in fairness testing.

We only consider notebooks written in Python[2] due to its popularity and rich ecosystem of data science packages. As an additional measure of quality, we only consider notebooks that are fully reproducible in a containerised Docker environment and utilise stable and well tested python packages[3].

Jupyter notebooks contain a mix of plain-text (written in Markdown, a popular markup language[4]) and code cells. Since

[1]https://kaggle.com
[2]https://python.org
[3]Jupyter provides a Docker image containing all popular packages https://hub.docker.com/r/jupyter/datascience-notebook
[4]https://daringfireball.net/projects/markdown/

979-8-3503-0113-7/23/$31.00 ©2023 IEEE
DOI 10.1109/CAIN58948.2023.00028

notebook cells are represented internally as JSON fields[5], Jupyter notebooks are machine parsable thus allowing us to separate the plain-text and code cells into individual files.

We wish to conduct the analysis in two phases. First, a qualitative study using a smaller sample of notebooks (for instance, 5% of the top voted notebooks from the top 5 competitions or datasets) to gather general insights and trends. And next, utilise our knowledge from the qualitative study to design an empirical study using a larger sample of notebooks.

III. Challenges

The data collection process of downloading a sufficiently large sample of Jupyter notebooks poses a significant engineering challenge. This requires a working knowledge of web technologies to programmatically download notebooks using the Kaggle API. Furthermore, reproducing all notebooks in a Docker container can be a resource and time exhaustive process and thus needs to be efficiently parallelised across the resources of the computing device.

Notebooks are notoriously well known for not adhering to software engineering standards and best practises. In addition to external packages, notebooks also require the associated dataset(s) in order to be reproduced. Prior work in mining Jupyter notebooks is limited. However there are some relevant contributions that we wish to adapt for our study. Specifically, *Pimentel et al* [3] and *Quaranta et al* [4] mine a large quantity of Jupyter notebooks which can aid with the data collection pipeline for this study and provide helpful guidelines on reproducing computational notebooks.

We also expect significant challenges in isolating and extracting the code cells that generate visualisations. *Bavishi et al* [5] mine Jupyter notebooks to create an automated tool that suggests visualisation code for a given dataset and text prompt from the user. We wish to adapt their methodology for isolating visualisation code cells, to collect the images for our study.

IV. Expected Outcomes

Jupyter notebooks present a rich source of information in three different formats—text, code and images. We hope to gain insights into the ML testing process by mining each source of information separately. To the best of our knowledge, this has not been attempted before.

We hope to utilise the existing literature on image analysis to derive interesting insights from the visualisations collected in the study [6], [7]. Additionally, this study also presents us with an opportunity to utilise the state-of-the-art developments in Natural Language Processing (NLP) both for plain-text and code [8]–[10]. By combining all three information sources, we hope to gain a new perspective into program comprehension and debugging in the context of ML testing.

[5]https://ipython.org/ipython-doc/3/notebook/nbformat.html

V. Outcomes Beyond ML Testing

We also see opportunities to make scientific contributions that go beyond ML testing. In particular, we hope to gain interesting insights into reproducibility and maintainability of computational notebooks from our data collection and preprocessing pipeline.

Our systematic methodology for obtaining, parsing and analysing computational notebooks may be packaged into an automated analytics tools. Given a notebook as input, the tool should produce meaningful insights using all three data sources. With this tool, we wish to aid future researchers who want to mine computational notebooks.

Prior studies have been conducted to understand ML pipelines and formalise their development process [11], [12]. However the data collected from these studies are not openly accessible due to confidentially restrictions. Our methodology of mining computational notebooks can also be used to validate prior claims and provide empirical evidence which is reproducible and publicly available.

We also see links to explainability in AI systems. Specifically, we hope to gain insights into what tools and techniques practitioners are using to understand black-box models and how they vary as the complexity of the underlying data and ML model changes.

References

[1] J. M. Zhang, M. Harman, L. Ma, and Y. Liu, "Machine learning testing: Survey, landscapes and horizons," *IEEE Transactions on Software Engineering*, 2020.

[2] E. Breck, N. Polyzotis, S. Roy, S. Whang, and M. Zinkevich, "Data validation for machine learning." in *MLSys*, 2019.

[3] J. F. Pimentel, L. Murta, V. Braganholo, and J. Freire, "A large-scale study about quality and reproducibility of jupyter notebooks," in *2019 IEEE/ACM 16th international conference on mining software repositories (MSR)*. IEEE, 2019, pp. 507–517.

[4] L. Quaranta, F. Calefato, and F. Lanubile, "Kgtorrent: A dataset of python jupyter notebooks from kaggle," in *2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR)*. IEEE, 2021, pp. 550–554.

[5] R. Bavishi, S. Laddad, H. Yoshida, M. R. Prasad, and K. Sen, "Vizsmith: Automated visualization synthesis by mining data-science notebooks," in *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2021, pp. 129–141.

[6] M. Nixon and A. Aguado, *Feature extraction and image processing for computer vision*. Academic press, 2019.

[7] J. R. Parker, *Algorithms for image processing and computer vision*. John Wiley & Sons, 2010.

[8] P. Devanbu, M. Dwyer, S. Elbaum, M. Lowry, K. Moran, D. Poshyvanyk, B. Ray, R. Singh, and X. Zhang, "Deep learning & software engineering: State of research and future directions," *arXiv preprint arXiv:2009.08525*, 2020.

[9] D. Zan, B. Chen, F. Zhang, D. Lu, B. Wu, B. Guan, Y. Wang, and J.-G. Lou, "When neural model meets nl2code: A survey," *arXiv preprint arXiv:2212.09420*, 2022.

[10] M. Allamanis, E. T. Barr, P. Devanbu, and C. Sutton, "A survey of machine learning for big code and naturalness," *ACM Computing Surveys (CSUR)*, vol. 51, no. 4, pp. 1–37, 2018.

[11] N. Sambasivan, S. Kapania, H. Highfill, D. Akrong, P. Paritosh, and L. M. Aroyo, ""everyone wants to do the model work, not the data work": Data cascades in high-stakes ai," in *proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021, pp. 1–15.

[12] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, and D. Dennison, "Hidden technical debt in machine learning systems," *Advances in neural information processing systems*, vol. 28, pp. 2503–2511, 2015.