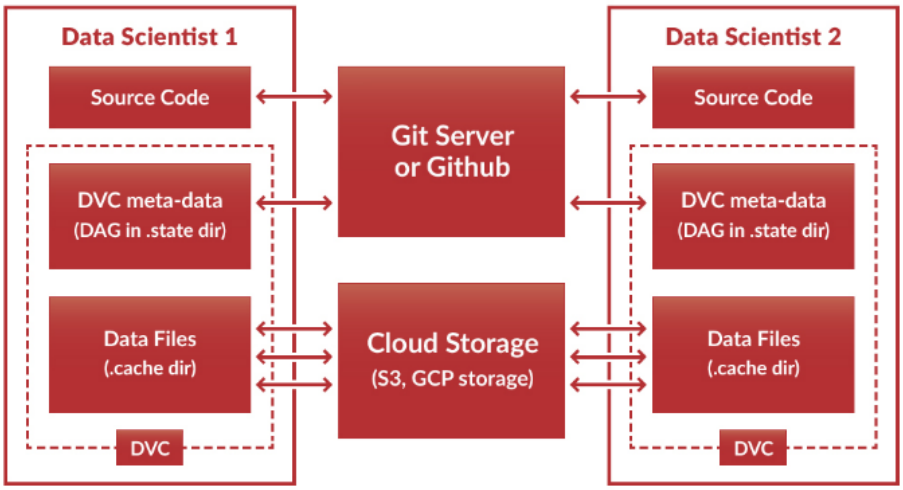
How to version control your production machine learning models

Algorithmia <https://algorithmia.com/blog/author/algorithmia>
25 June 2018 | 7 min read



Source: KDnuggets <https://www.kdnuggets.com/2017/05/data-version-control-iterative-machine-learning.html>

Machine learning is about rapid experimentation and iteration, and without keeping track of your modeling history you won't be able to learn much. Versioning lets you keep track of all of your models, how well they've done, and what hyperparameters you used to get there. This post will walk through why data versioning is important, tools to get it done with, and how to version your models that go into production.

The Importance of Model Versioning

If you've spent time working with Machine Learning, one thing is clear: it's an iterative process. There are so many different parts of your model—how you use your data, hyperparameters, parameters, algorithm choice, architecture—and the optimal combination of all of those is the holy grail of machine learning. But while there is some method to the madness, much of finding the right balance is trial and error. Even the best machine learning engineers working on the most complex deep learning projects still need to tinker to get their models right.

With that in mind, here are some of the reasons why versioning is so important to machine learning projects:

1. Finding the best model

Throughout that iterative process of updating and tinkering with the different parts of your model, your accuracy on your dataset will vary accordingly. In order to keep track of the best models you've created and the associated tradeoffs, you need to have a data versioning system in practice.

2. Failure tolerance

When pushing new versions of models into production, they can fail for any number of reasons. You want to update your models to take new data into account or incorporate speed improvements, but it's tough to be sure how they'll perform in real time. If you do encounter an issue with a production model, you *need* to be able to revert quickly to the previous working version.

3. Increased complexity and file dependencies

With traditional software versioning, there are only a couple of types of files to keep track of – your code, and your dependencies. With machine learning though, things are a bit more complex. First and foremost, you have datasets (typically not part of a normal software deployment). You need to keep track of what data you train and test on, and if that changes over time.

Additionally, saving your models in most of the popular deep learning frameworks results in a file that you need to keep track of. Finally, models are often written in different languages and rely on multiple frameworks, which makes dependency tracking even more important.

4. Gradual, staged deployment

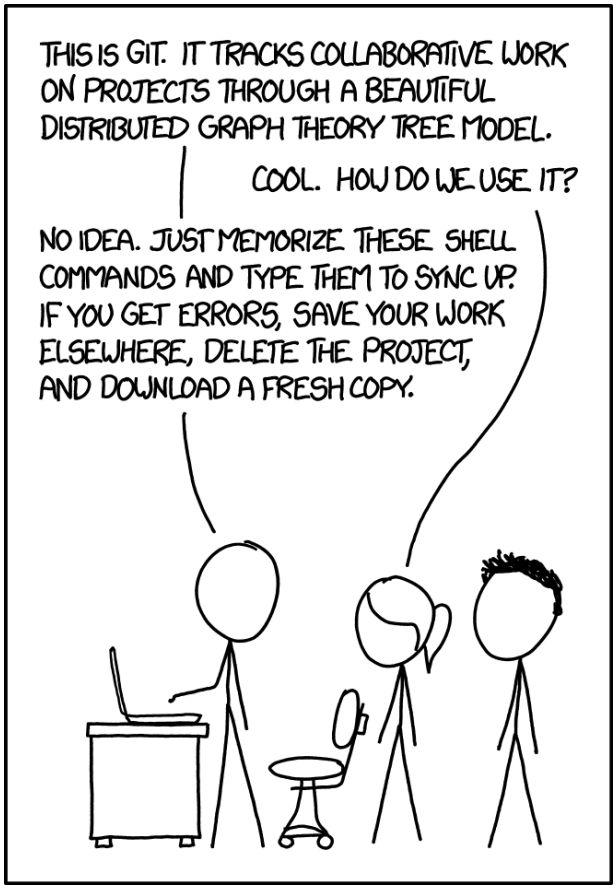
If and when you make significant updates to your production models, those major changes are rarely deployed immediately and in one shot. To ensure failure tolerance and test appropriately, new models are typically rolled out gradually until teams can be sure that they're working properly. Versioning gives you the tools to deploy the right data versions at the right times.

Versioning Tools to Get The Job Done

It's hard to understate how nascent the field of production Machine Learning is, and that means the tools supporting this ecosystem are only starting to be fully developed. Here are some of the solutions that practitioners are currently using, and some new entrants too.

1. Git

Git is *the* versioning protocol used across the board to monitor and version software development and deployment. You might be familiar with GitHub <https://github.com/> or BitBucket <https://bitbucket.org/>, which are web-based commercial implementations of this open-source tool. Git tracks any changes made to your code and gives you functionality around implementing, storing, and merging those changes. Pretty much everyone uses it in one way or another.



Source: xkcd <https://xkcd.com/1597/>

But alas, Git is not without its issues. In addition to the often perplexing nature of using the actual protocol, its missing a lot of the functionality that you need for machine learning (because it wasn't created for ML!). Git itself doesn't allow you to track data, changes to model files, and model dependencies. There are extensions that can help, but those solutions are tough to implement and rarely complete.

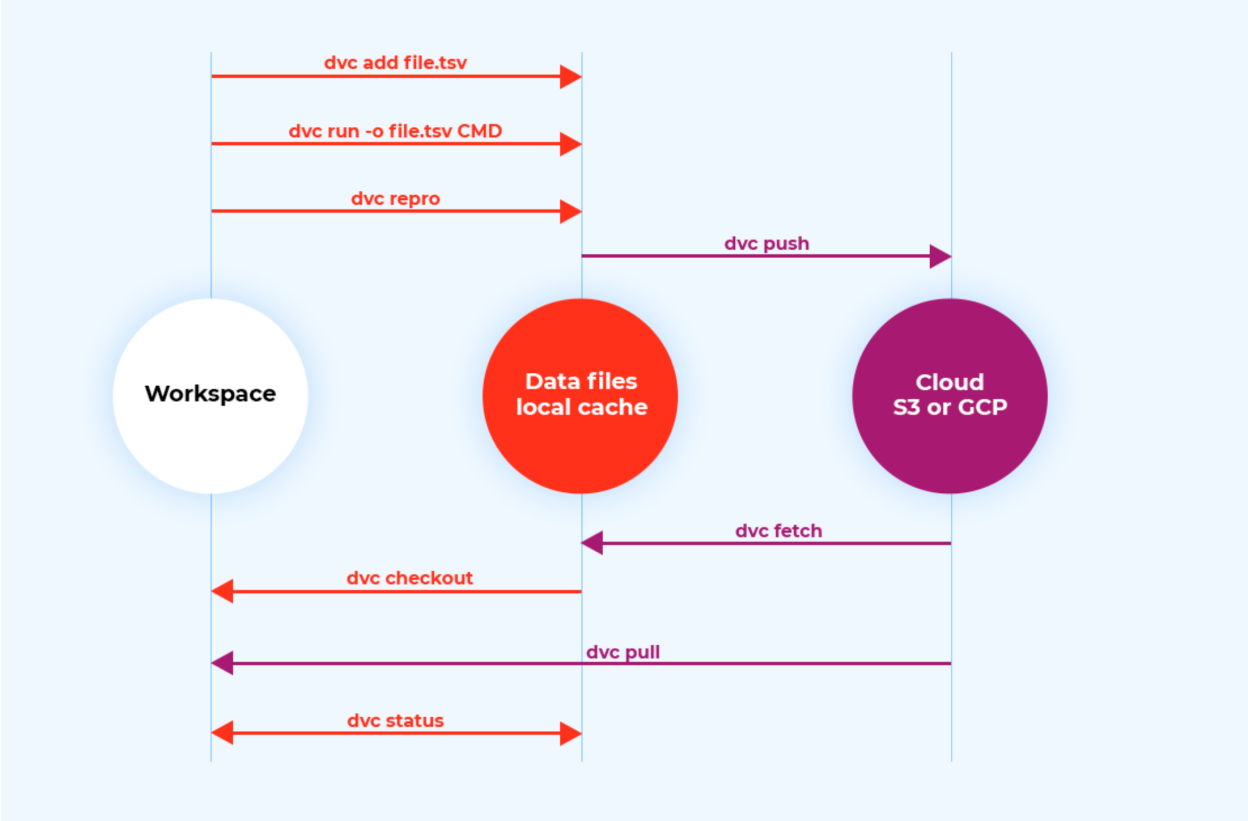
2. Sandbox environments

Data scientists often rave about Jupyter Notebooks, a sandbox-type environment that lets you run code in cells and insert Markdown in between (or at least I rave about them). Jupyter Notebooks are like writing a book with code in them: you can be detailed about what each cell does, and organize things in a visually pleasing way. Separating code into cells and sections is a viable way to version your different models.

When it comes to deployment and production though, versioning your models in a notebook doesn't really cut it. Jupyter Notebooks are a tool for exploration and visualization, not for managing dependencies and tracking minute changes to hyperparameters.

3. Data Version Control (DVC)

Data Version Control (DVC) is a Git extension that adds functionality for managing your code and data together. It works directly with cloud storage (AWS S3 or Google GCP) to push your data changes. For example, according to their tutorial, "DVC streamlines large data files and binary models into a single Git environment and this approach will not require storing binary files in your Git repository." It's a streamlined version of combining Git with machine learning specific functionality for data management.



For a tutorial on how to implement DVC in your project and why it's so helpful, check out this walkthrough <https://becominghuman.ai/how-to-version-control-your-machine-learning-task-ii-d37da60ef570>.

4. Commercial solutions

The traditional business wisdom tells us that if there's a problem, there's a company. There are a few companies starting out attempting to solve the data versioning problem. Comet.ml <https://www.comet.ml/> is an automatic versioning solution that tracks and organizes all of your team's modeling efforts. You can easily compare experiments, see the differences in code between two models, and invite team members to collaborate on a project.

5. Platforms as a Service and Algorithmia

Even once you've found a way to manage data versioning during your training and experimentation process, much of the complexity resides in inference: deploying the right models in the right places at the right times. If you're using a Platform as a Service to deploy your machine learning models, it might offer some functionality around data versioning.

If you're deployed on the Algorithmia platform, we productionize your models as independent microservices with individual endpoints. That means you can continue to reference historical versions of your models in production without having to worry about them breaking or getting deprecated. It's as simple as appending the model name in our

API with a version number.

Further Reading

How to Version Control your Machine Learning task <https://towardsdatascience.com/how-to-version-control-your-machine-learning-task-cad74dce44c4> (Towards Data Science) – "*A component of software configuration management, version control, also known as revision control or source control, is the management of changes to documents, computer programs, large web sites, and other collections of information. Revisions can be compared, restored, and with some types of files, merged.*"

Version Control for Data Science <https://www.datacamp.com/community/blog/version-control-data-science> (DataCamp) – "*Keeping track of changes that you or your collaborators make to data and software is a critical part of any project, whether it's research, data science, or software engineering. Being able to reference or retrieve a specific version of the entire project aids in reproducibility for you leading up to publication, when responding to reviewer comments, and when providing supporting information for reviewers, editors, and readers.*"

Managing and versioning Machine Learning models in Python <https://www.slideshare.net/fridiculous/managing-and-versioning-machine-learning-models-in-python> (SlideShare) – "*Practical machine learning is becoming messy, and while there are lots of algorithms, there is still a lot of infrastructure needed to manage and organize the models and datasets. Estimators and Django-Estimators are two python packages that can help version data sets and models, for deployment and effective workflow.*"

Data Version Control: iterative machine learning <https://www.kdnuggets.com/2017/05/data-version-control-iterative-machine-learning.html> (KDnuggets) – "*Today, we are pleased to announce the beta version release of new open source tool—data version control or DVC. DVC is designed to help data scientists keep track of their ML processes and file dependencies. Your existing ML processes can be easily transformed into reproducible DVC pipelines regardless of which programming language or tool was used.*"

Machine learning|recipes <https://algorithmia.com/blog/category/machine-learningrecipes>

Algorithmia

More posts from Algorithmia <https://algorithmia.com/blog/author/algorithmia>

More articles

<https://algorithmia.com/blog/statistics-and-machine-learning-whats-the-difference>

21 May 2020 6 min read

Statistics and machine learning: what's the difference? <https://algorithmia.com/blog/statistics-and-machine-learning-whats-the-difference>

<https://algorithmia.com/blog/appen-and-algorithmia-combined-deployment-and-training-pipeline-to-ensure-ai-ml-business-value>

20 May 2020 3 min read

Appen and Algorithmia: Combined deployment and training pipeline to ensure AI/ML business value <https://algorithmia.com/blog/appen-and-algorithmia-combined-deployment-and-training-pipeline-to-ensure-ai-ml-business-value>

<https://algorithmia.com/blog/the-8-best-machine-learning-books>

14 May 2020 4 min read

The 8 best machine learning books <https://algorithmia.com/blog/the-8-best-machine-learning-books>

RESOURCES

Getting Started <https://algorithmia.com/getting-started>

Docs <https://algorithmia.com/developers>

Learn <https://algorithmia.com/learn>

Research <https://algorithmia.com/research>

Contact Us <https://algorithmia.com/contact>