Getting Started with Data Version Control (DVC)

Dheeraj Bhat — Updated On June 13th, 2023

Beginner Data Science Machine Learning

Introduction

If you are reading this blog, you might have been familiar with what Git is and how it has been an integral part of software development. Similarly, Data Version Control (DVC) is an open-source, Git-based version management for Machine Learning development that instills best practices across the teams. A system called data version control manages and tracks changes to data and machine learning models in a collaborative and reproducible manner. It draws inspiration from version control systems used in software development, such as Git, but tailors specifically to data science projects.

Learning Objectives

In this article you will develop basic understanding of:

- What is Git?
- What is Data Version Control?
- Understand the basics of Data Version Control

This article was published as a part of the Data Science Blogathon.

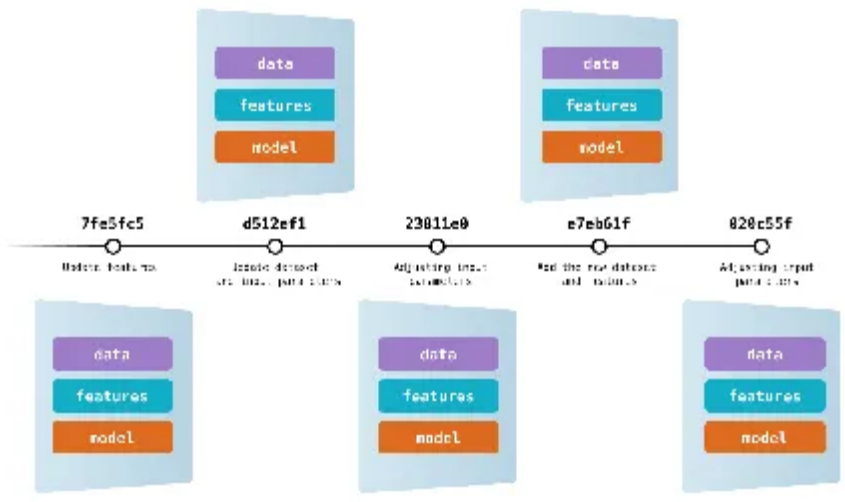
Table of contents

Advantages of Data Version Control (DVC)

ML Project Version Control

DVC lets you connect with storage providers like AWS S3, Microsoft Azure Blob Storage, Google Drive, Google Cloud Storage, HDFS, etc., to store ML models and datasets.

ML Experiment Management



PyPi repository using the following command line:

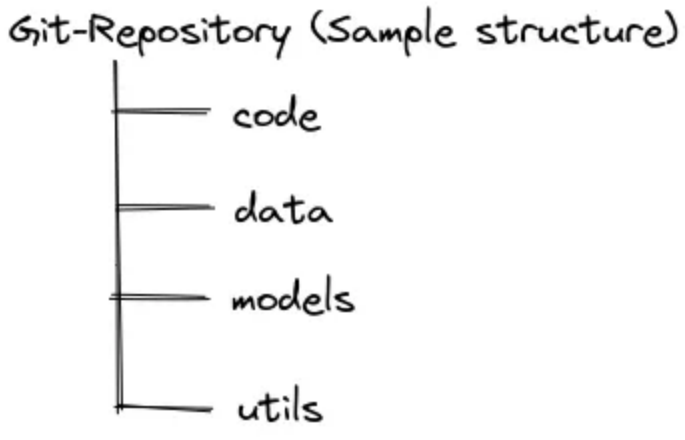
```
pip install dvc
```

Depending on the type of remote storage that will be used, we have to install optional dependencies: [s3], [gdrive], [gs], [azure], [ssh], [hdfs], [webdav], [oss]. Use [all] to include them all. In this blog, we will be using google drive as remote storage, so pip install dvc[gdrive] for installing gdrive dependencies.

Learn More: Tracking ML Experiments With Data Version Control

Getting Started

In this blog, we will see how to use dvc for tracking data and ml models with gdrive as remote storage. Imagine the Git repository which contains the following structure:



Gdrive Remote Configuration

Now, we need to configure gdrive remote storage. Go to your google drive and create a folder called dvc_storage in it

Now, use the following command to use the dvc_storage folder created in the google drive as remote storage:

```
dvc remote add myremote gdrive://folder-id

# example: dvc remote add myremote gdrive://0AIac4JZqHhKmUk9PDA
```

Now, we need to commit the changes to git repository by using the command:

```
git add -A
git commit -m "configure dvc remote storage"
```

To push the data to remote storage, we use the following command:

```
dvc push
```

Then, we push the changes to git using the command:

```
git push
```

To pull data from dvc, we can use the following command:

```
dvc pull
```

DVC Pipelines

We can make use of DVC pipelines to reproduce the workflows in our repository. The main advantage of this is that we can go back to a particular point in time and run the pipeline to reproduce the same result that we had achieved during the previous time. There are different stages in the DVC pipeline like *prepare*, *train*, and *evaluate*, with each of them performing different tasks. The DVC pipeline is nothing but a DAG (Directed Acyclic Graph). In this DAG graph, there are nodes and edges, with nodes representing the stages and edges representing the direct dependencies. The pipeline is defined in a YAML file (dvc.yaml). A simple dvc.yaml file is as follows:

```
    deps:
      - src/cleanup.sh
      - data/raw
    outs:
      - data/clean.csv
  train:
    cmd: python src/model.py data/model.csv
    deps:
      - src/model.py
      - data/clean.csv
    outs:
      - data/predict.dat
  evaluate:
    cmd: python src/evaluate.py data/predict.dat
    deps:
      - src/evaluate.py
      - data/predict.dat
```

Use the *prepare* stage to run the data cleaning and pre-processing steps. Use the *train* stage to train the machine learning model using the data from the prepare stage. The *evaluate* stage uses the trained model and predictions to provide different plots and metrics.

Conclusion

This blog helps you with the basics of Data Version Control and set up dvc using google drive as remote storage. For advanced uses (like CI/CD etc.), we need to set up DVC remote configuration using the Google Cloud project (click here). There are also other storage types supported like AWS S3, Microsoft Azure Blob Storage, self-hosted SSH servers, HDFS, HTTP, etc. DVC has most of the commands analogous to git (like dvc fetch, dvc checkout, and dvc status, etc, and a lot more). It also has Visual Studio Extension which makes things easier for developers using VS Code. Check out their GitHub repository to learn more about DVC and everything it offers.

Key Takeaways:

- Understanding the basics of DVC
- Become acquainted with the use cases of DVC
- Installation and use of DVC in a git repository
- GDrive Remote configuration in DVC

References

- https://dvc.org/
- https://github.com/iterative/dvc

The media shown in this article is not owned by Analytics Vidhya and is used at the Author's discretion.

Frequently Asked Questions

Q1. What is the DVC command?

A. The DVC command is a command-line tool that provides various functionalities for interacting with DVC projects. It includes commands for initializing a DVC project, tracking data files, managing data pipelines, running experiments, and collaborating with other team members. It serves as the primary interface for interacting with DVC's features.

for reproducibility and efficient collaboration.

Q3. What is DVC used for?

A. DVC is used for managing and versioning large datasets, machine learning models, and experiments. It helps streamline the data pipeline, enables reproducibility, and facilitates collaboration among data scientists and machine learning engineers.

Q4. Why use DVC instead of Git?

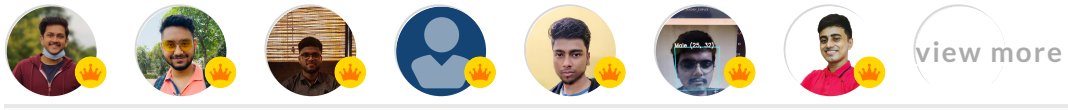
A. DVC complements Git by focusing on versioning and managing data and machine learning models, while Git primarily handles source code. DVC's dedicated functionality for data and models includes handling large files efficiently, storing data separately, and enabling reproducibility, which are essential for machine learning projects.

blogathon data science data science projects DVC Studio git machine learning software development

About the Author

Dheeraj Bhat

Our Top Authors



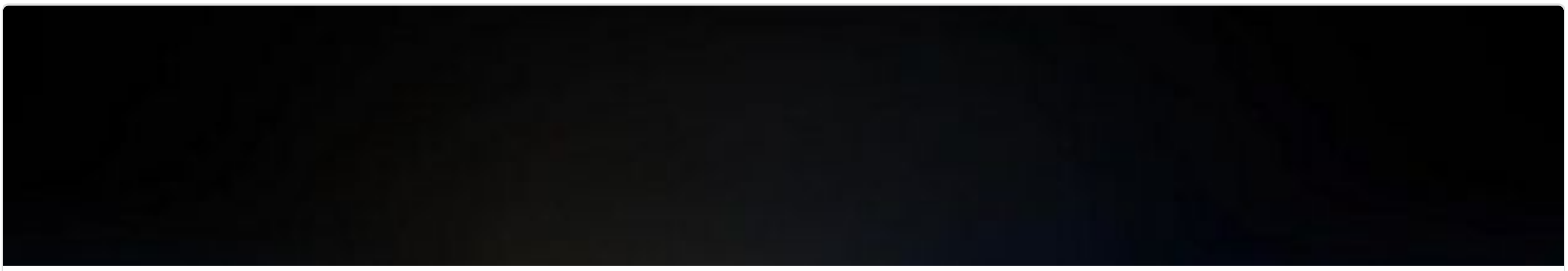
Download

Analytics Vidhya App for the Latest blog/Article

Next Post

OpenAI Veterans' Anthropic Startup Raises $450M to Compete with OpenAI

Top Resources



Building an LLM Model using Google Gemini API

Ajay Kumar Reddy - DEC 15, 2023