Data Versioning (DVC)

→ Why Data Versioning?

Suppose you built a ML Model

↳ Collect data → data-V1.csv
↳ Clean it → data-V2.csv
↳ Add new feature → data-V3.csv
↳ Train a model → model-V1.pkl
↳ Try new Params → model-V2.pkl
↳ Then you came back next month with version of data give best model?

Real world Problem with D.V

→ ~~Data~~ Trained a best model, but didnot know which data used
→ Your team mate donot have access to best data.
→ You need to track data + model + Params But git only track code.

So why DVC?

	DVC	Git
Versioning	✓	✓
Handles ~~git~~ large file	✓	✗
Team Sharing	✓	✓
Track Experiment	✓	✗

FREEMIND

Date

Page

What is DVC

It is Git But for large Dataset and ML Pipelining

→ Version Control datasets and model file (without bloating Git)
→ Track with Version of data for which Version of code.
→ Share with Team / Cloud easily.

are simple hai yr.
Git har changes per ek Version banata hi
~~But DVC nahi~~

How DVC Work.

(1) Theory → for every version of git code DVC Maintain a Data

Git DVC

V1
- Code
- txt
- .ignore
- .dvc-1

Data

V2
- Code
- txt
- ignore
- .dvc-2

Data2

Basically git has large file so, it just store reference of code and Pointer DVC

Now DVC Coding.

→ Made a file and defined a Simple df. and perform git.

→ Since I have my git repo in Parent Dir MLOps, what I will is that I will just track my Day3_DVC folder.

→ DVC Install and DVC init → .dvc, .dvcignore

→ Creating a Space where DVC will Store Models/Data
→ As of know I will use my local and make a Dir 'S3'
→ Used dvc remote add -d myremote S3
→ Now we will give folder/file to tracked by DVC
↳ we need to de-track that file/folder first from git
↳ git rm -r --cache <file/folder name>
↳ git commit -m 'Removed'

↳ now we use [dvc add file/folder] to track them,
↳ in my case I will track Day3DVC/DataDir

↳ After Doing this we got two file
↳ DataDir.dvc → Contain Id of Version
↳ .gitignore. → Contain /DataDir

↓ we can add these two file in our git to track changes

Date
Page

→ Now 'dvc commit' and 'dvc Push' → This will Push our data in our remote space [in my case S3].
  ↳ Here My S3 folder will contain 2 file [for 1 file push 2 file]
    ↳ 1. file contain Id
    ↳ 2. file contain Data

→ Now technically You should git your first Version.

→ Now we can Make V_2 of data.
  ↳ Change you data.
    ↳ Dvc commit & Push
    ↳ See DVC status
    ↳ git Push it

Now our main USP was to roll back to version and version

→ list all version → git log --oneline [Copy shold of Version]
→ go to that version → git checkout <shaid>
→ Pull → Check Dvc status → You see Change out.
Now Pull that Particular Version of data → dvc Pull

or.

git checkout Main
dvc Pull