

DVC (DATA VERSION CONTROL)

- STORE AND PROCESS DATA FILES (or) DATASETS
 - TO PRODUCE OTHER DATA (or) MACHINE LEARNING MODELS
 - ↳ TRACK AND SAVE DATA AND MACHINE LEARNING MODELS (THE SAME WAY YOU CAPTURE CODE)
 - ↳ COMPARE MODEL METRICS
 - ↳ CREATE AND SWITCH B/W VERSIONS OF DATA AND ML MODELS EASILY
 - ↳ IN DATA INGESTION PHASE YOU GET DIFFERENT TYPE OF DATA.
- YOU CAN MANAGE ENTIRE VERSIONS OF DATA & MODELS USING DVC
- DVC WILL ONLY TRACK THE DATA FILES
- GIT WILL BE USED TO TRACK CODE INFORMATION AND CONFIGURATION FILE THAT IS REQUIRED BY

DVC

- FIRST DO THE GIT INIT IN THE VS CODE TERMINAL
- INSTALL DVC
 - ↳ PIP INSTALL -R REQUIREMENTS.TXT FILE
(or)
 - ↳ PIP INSTALL DVC
- CREATE A DATA FOLDER INSIDE THAT
 - ↳ CREATE A SAMPLE DATA.TXT FILE
- DVC INIT IN THE TERMINAL
 - ↳ ONCE YOU PRESS ENTER .DVC FOLDER WILL BE CREATED WITH IN tmp .GITIGNORE, config FILE WILL BE CREATED
- IN config file YOU CAN PROVIDE DETAILS ABOUT THE REMOTE REPOSITORY WHERE YOU WANT TO STORE THE DATA (YOU CAN GIVE THOSE DETAILS)
- NEXT USING GIT STATUS COMMAND IN THE TERMINAL YOU CAN SEE UPDATED DVC FILES

IN IT

```
Your branch is up to date with 'origin/main'.
```

```
Changes to be committed:  
(use "git restore --staged <file>..." to unstage)  
  new file:  .dvc/.gitignore  
  new file:  .dvc/config  
  new file:  .dvcignore
```

```
(F:\DVC Intro\venv) F:\DVC Intro>
```

- ABOVE PROVIDED FILES IN THE IMAGE NEED TO BE TRACKED
 - ↳ .DVC IGNORE (IF YOU WANT IGNORE ANY FILES YOU CAN MENTION HERE SAME AS .GIT IGNORE)
 - ↳ COMMIT CODE MORE FREQUENTLY .
- DVC ADD DATA / DATA.TXT (THIS IS THE FILE YOU NEED TO TRACK. (IT CAN BE CSV FILE , MODEL FILE IT CAN BE ANY FILE))

The screenshot shows a terminal window in a dark-themed code editor. The current directory is F:\DVC Intro\venv. The terminal output shows:

```
(F:\DVC Intro\venv) F:\DVC Intro git commit -m "dvc init"
error: pathspec 'init' did not match any file(s) known to git
```

Below this, the command was run again:

```
(F:\DVC Intro\venv) F:\DVC Intro git commit -m "dvc init"
[main branch] dvc init
 3 files changed, 0 insertions(+)
 create mode 100644 .dvc/.gitignore
 create mode 100644 .dvc/config
 create mode 100644 .dvcignore
```

→ ONCE AFTER ENTERING THE ABOVE
COMMAND

IT SAYS TO TRACK CHANGES WITH GIT RUN:
GIT ADD 'DATA\DATA.TXT'

The screenshot shows a terminal window in a dark-themed code editor. The current directory is F:\DVC Intro\venv. The terminal output shows:

```
(F:\DVC Intro\venv) F:\DVC Intro> dvc add data\data.txt
100% Adding... [1/1 [00:00, 9.45file/s]
```

Below this, instructions are provided:

```
To track the changes with git, run:
  git add 'data\data.txt.dvc'

To enable auto staging, run:
  dvc config core.autostage true
```

```
git add 'data\data.txt.dvc'
To enable auto staging, run:
    dvc config core.autostage true
(F:\DVC Intro\venv) F:\DVC Intro[]
```

→ You can see .GITIGNORE in DATA folder where

- GITIGNORE file is created so that it will say GIT that it DON'T NEED to TRACK DATA FILE

→ But .GITIGNORE & DATA.TEXT.DVC will be tracked

The screenshot shows a code editor interface with the following details:

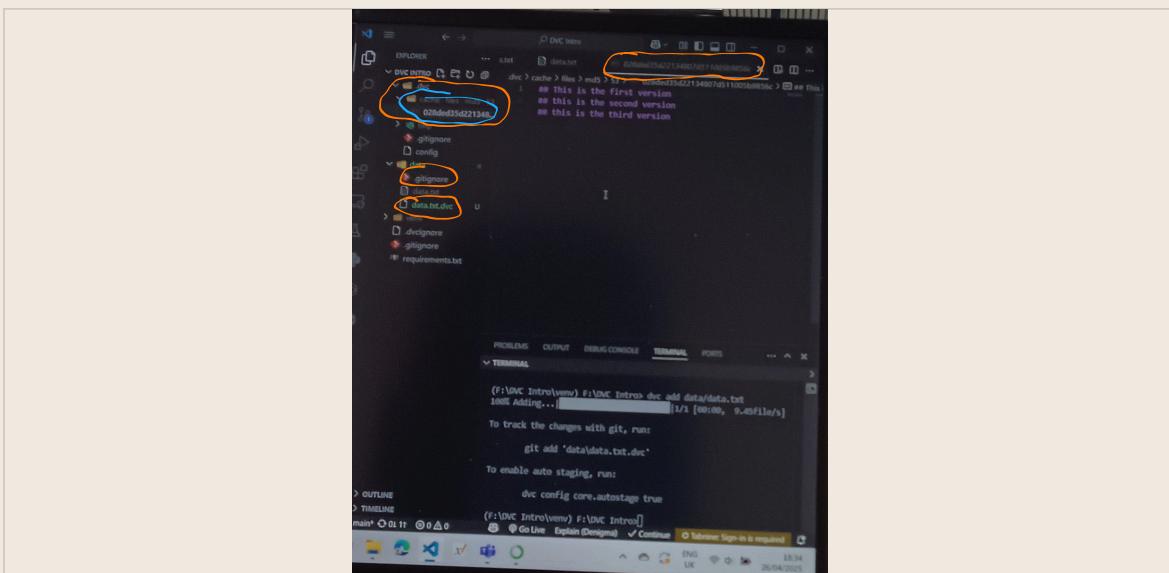
- File Explorer:** Shows a project structure with folders like `DVC INTRO`, `dvc`, `data`, and `venv`. Inside `data`, there are files `data.txt` and `data.txt.dvc`. The `data.txt.dvc` file is highlighted with a red circle.
- Terminal:** Shows the command `dvc add data\data.txt` being run. The output indicates "100% Adding..." and "1/1 [00:00, 9.45file/s]". It also provides instructions to track changes with `git add 'data\data.txt.dvc'` and to enable auto staging with `dvc config core.autostage true`.

- **MDS** IS THE HASH KEY OF DATA.TXT FILE
- ↳ WHAT EVER CONTENT IN THE ABOVE FILE DVC IS CREATING HASH KEY (MDS)
 - ↳ THIS HASH MAPS TO A SPECIFIC DATA (IN OUR DATA.TXT FILE)
 - ↳ IF YOU CHANGE THE CONTENT IN DATA.TXT FILE ANOTHER HASH KEY (MDS) WILL BE GENERATED

HOW THIS IS GETTING MAPPED ?

IS IT STORED IN SPECIFIC LOCATION?

- YOU CAN FIND THAT IN THE •DVC FOLDER
INSIDE IT •DVC/CACHE





- .gitignore & DATA.TXT.DVC WILL BE TRACKED BY GIT
- SO THAT YOU CAN STORE YOUR CACHE FOLDER ANYWHERE (GOOGLE CLOUD, AWS, AZURE ETC)
- BECAUSE DATA.TXT.DVC HAS REFERENCE INFORMATION (MD5) OF (DATA.TXT WHICH IS NOT GETTING TRACKED BY GIT)

```
(F:\DVC Intro\venv) F:\DVC Intro>dvc add data/data.txt
100% Adding... [1/1 [00:00, 15.62File/s]
```

To track the changes with git, run:

```
git add 'data\data.txt.dvc'
```

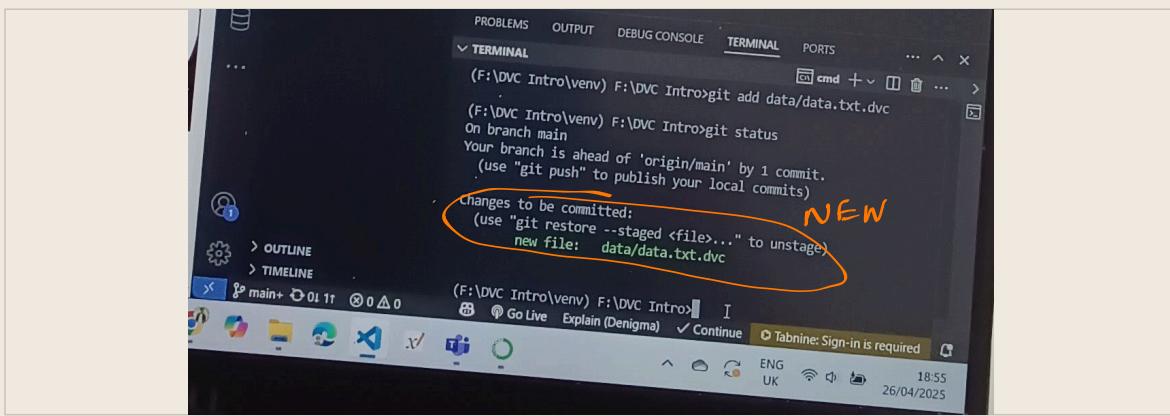
To enable auto staging, run:

```
dvc config core.autostage true
```

→ Once you add the data to DATA.TXT.DVC
↳

Commit with DVC ADD DATA|DATA.TXT command

→ MD5 hash key will be changed (you will have one more folder)



→ Now push DATA|DATA.TXT.DVC to Git

→ Enter GIT LOG in CMD

→ You can see all the version commits

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL FOLDERS
TERMINAL V1
...skipping...
commit 3d3c0bf33c3505c907f2d2f3ea1fbbe9c35a34ed HEAD -> main
Author: ManideepUddagowri <mudagowri.manideep@gmail.com>
Date: Sat Apr 26 18:57:53 2025 +0100
dvc
V2
commit 8fa1affb4dcdaa777fdb560c1cd5f0d562275ab2
Author: ManideepUddagowri <mudagowri.manideep@gmail.com>
Date: Sat Apr 26 18:11:51 2025 +0100
dvc init
V3
commit 2a57f58da2f8495d17eb1f4ca3f6a4515ff26a66 origin/main
Author: ManideepUddagowri <mudagowri.manideep@gmail.com>
Date: Sat Apr 26 17:57:16 2025 +0100
commit

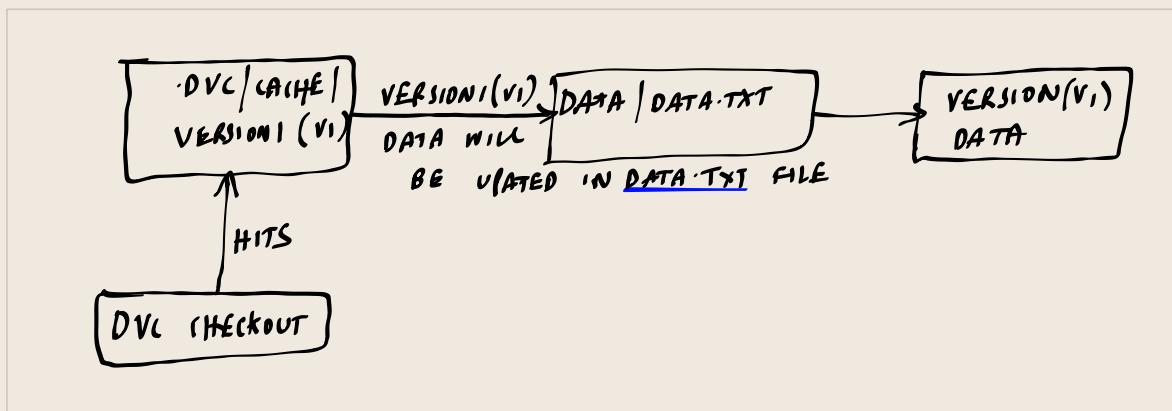
```

Go Live Explain (Denigma) Continue Tabnine: Sign-in is required ENG UK 15:14 28/04/2025

- IF YOU WANT SEE THE DATA VERSION1(V1)
copy
AND PASTE IT IN NEW CMD
- ACTIVATE VENV & ENTER GIT CHECKOUT
(PASTE THE
VERSION) WHICH IS VERSION1(V1)
- NOW IT WILL BE SWITCHED TO VERSION1(V1)
DATA IF YOU OBSERVE DATA.TXT.DVC FILE
YOU CAN SEE THE UPDATED (MDS)
- BUT DATA.TXT FILE WILL SHOW THE BEFORE

DATA UNTIL YOU DO DVC CHECKOUT

- HERE ONCE YOU DO DVC CHECK OUT IT WILL GO BACK TO THE VERSION (V₁) DATA IN THE CACHE FOLDER & CHECK FOR THAT VERSION & DISPLAYS THE DATA IN DATA / DATA.TXT FILE



- IF YOU JUST WANT TO GO BACK
→ ENTER GIT CHECKOUT YOUR BRANCH NAME (MAIN)
→ AND ENTER DVC CHECKOUT

The image shows two side-by-side terminal windows, each with an Explorer sidebar on the left.

Left Terminal Window:

- Explorer:** Shows a project structure with a .dvc folder containing cache, files, and md5 subfolders. Inside md5, there are three files: 1, 2, and 3. An orange bracket groups 1, 2, and 3, with a handwritten note '3' next to it.
- Terminal:**

```
F:\DVC Intro>conda activate venv/
(F:\DVC Intro\venv) F:\DVC Intro>dvc checkout
Building workspace index [0.00 [00:00, ?entry/s]
Comparing indexes [1.00 [00:00, ?entry/s]
Applying changes [0.00 [00:00, ?file/s]

(F:\DVC Intro\venv) F:\DVC Intro>git checkout master
error: pathspec 'master' did not match any file(s) known to git
```

Right Terminal Window:

- Explorer:** Shows a similar project structure. Inside md5, there are four files: 1, 2, 3, and 4. An orange bracket groups 1, 2, 3, and 4, with a handwritten note '4' next to it.
- Terminal:**

```
F:\DVC Intro>git checkout master
Previous HEAD position was f8f41aff dvc init
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

(F:\DVC Intro\venv) F:\DVC Intro>dvc checkout
Building workspace index [1.00 [00:00, 48.0entry/s]
Comparing indexes [3.00 [00:00, 2.00entry/s]
Applying changes [1.00 [00:00, ?152file/s]

(F:\DVC Intro\venv) F:\DVC Intro>
```