Como fazer controle de versões de dados e modelos de Machine Learning usando o DVC?

Alexandre Ray, Data Scientist

28/11/2019

creditas

O problema

creditas

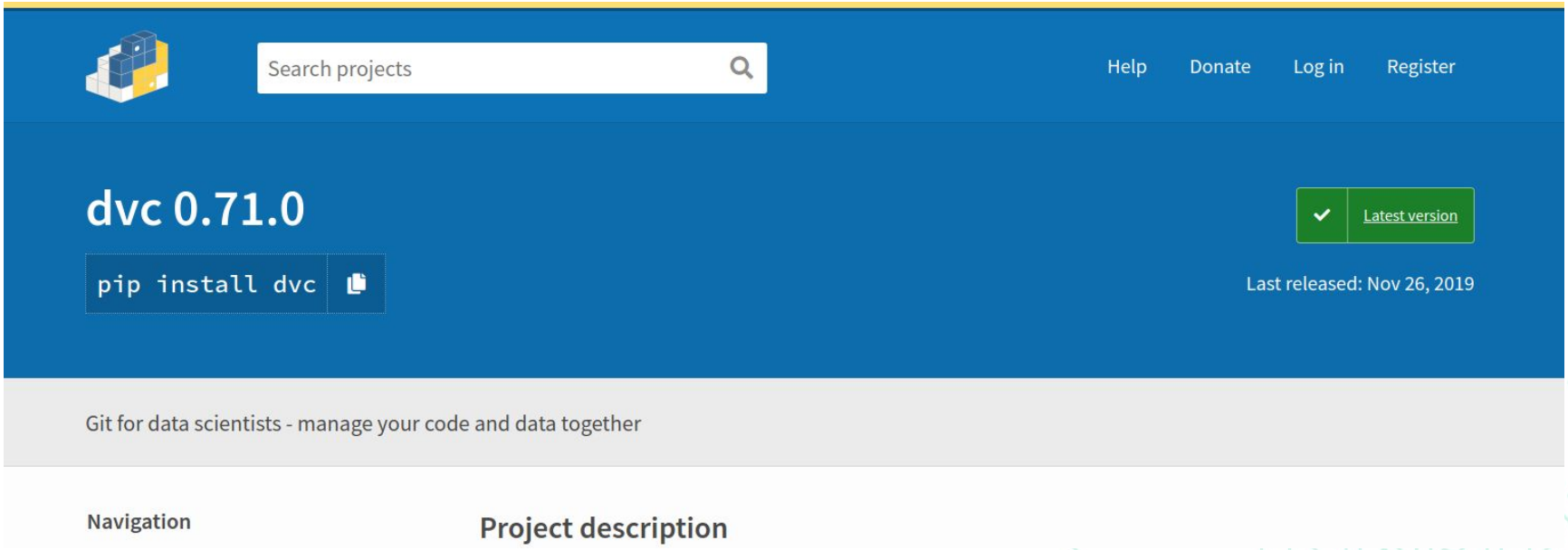
O que é

- Ferramenta para versionar artefatos (datasets e modelos)
- Baseada no Git
- Reprodutibilidade

Exemplo...

creditas

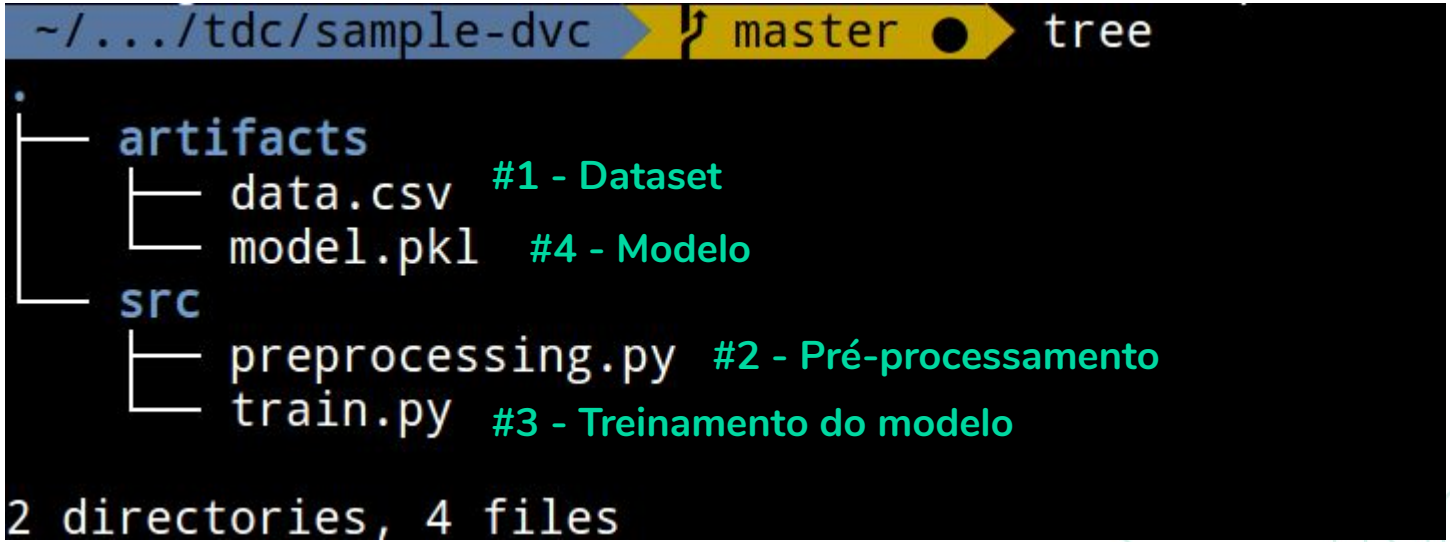
Instalação



creditas

Exemplo

Vamos supor que temos a estrutura de pastas de um projeto conforme a figura abaixo



Os arquivos da pasta src serão versionados pelo git e os arquivos da pasta artifacts serão versionados pelo DVC

Exemplo

```
~/.../tdc/sample-dvc  master ●  dvc init

You can now commit the changes to git.

+---------------------------------------------------------------------+
|                                                                     |
|        DVC has enabled anonymous aggregate usage analytics.         |
|     Read the analytics documentation (and how to opt-out) here:     |
|             https://dvc.org/doc/user-guide/analytics                |
|                                                                     |
+---------------------------------------------------------------------+

What's next?
------------
- Check out the documentation: https://dvc.org/doc
- Get help and share ideas: https://dvc.org/chat
- Star us on GitHub: https://github.com/iterative/dvc
~/.../tdc/sample-dvc  master ●  git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   .dvc/.gitignore
        new file:   .dvc/config

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        artifacts/
        src/
```

Para inicializar o DVC no repositório, basta digitar o comando git init

creditas

Exemplo

```
~/.../tdc/sample-dvc  master ●  dvc add artifacts
100% Add
1.00 [00:00<00:00,  3.26file/s]


To track the changes with git, run:

        git add .gitignore artifacts.dvc
~/.../tdc/sample-dvc  master ●  git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   .dvc/.gitignore
        new file:   .dvc/config

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        .gitignore
        artifacts.dvc
        src/
```

Para realizar o tracking da pasta artifacts, usamos o comando dvc add artifacts

Perceba que foi criado um arquivo artifacts.dvc
Esse arquivo é um metafile e é versionado pelo git

Automaticamente foi criado um .gitignore para que a pasta artifacts não seja mais versionada pelo git

Exemplo

```
~/.../tdc/sample-dvc  master ●  git add .
~/.../tdc/sample-dvc  master ●  git commit -m "Add scripts and DVC files"
[master (root-commit) f6bbdfe] Add scripts and DVC files
 6 files changed, 17 insertions(+)
 create mode 100644 .dvc/.gitignore
 create mode 100644 .dvc/config
 create mode 100644 .gitignore
 create mode 100644 artifacts.dvc
 create mode 100644 src/preprocessing.py
 create mode 100644 src/train.py
```

Exemplo

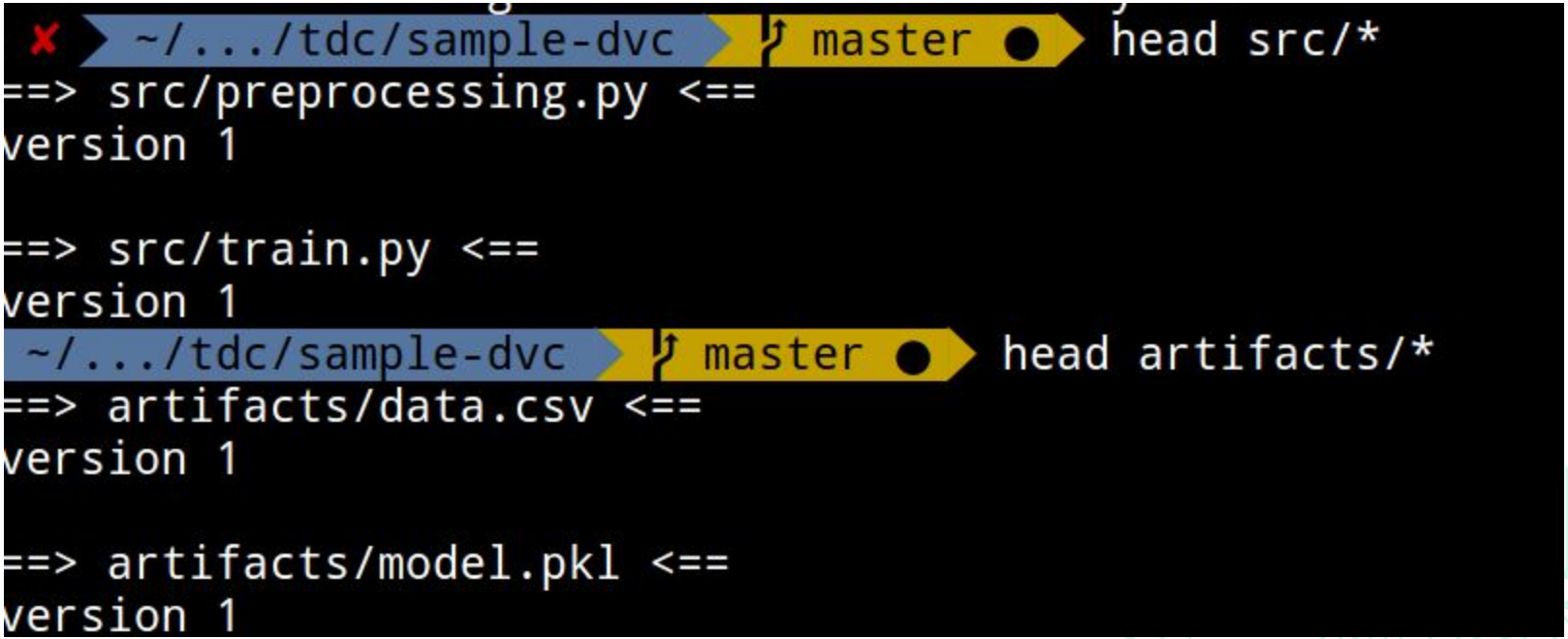
No término de um experimento, nós usamos a tag do git para “travar” todo o código que foi produzido até aquele momento

```
~/.../tdc/sample-dvc  master  git tag -a "experiment00" -m "Experiment 00"
~/.../tdc/sample-dvc  master  git tag
experiment00
```

creditas

Exemplo

Vamos supor que fizemos uma iteração do projeto e todos os arquivos foram alterados para a versão 1. A figura abaixo mostra o conteúdo de todos os arquivos



creditas

Exemplo

```
~/.../tdc/sample-dvc  master ●  dvc status
artifacts.dvc:
        changed outs:
                modified:           artifacts
~/.../tdc/sample-dvc  master ●  dvc add artifacts
WARNING: Output 'artifacts' of 'artifacts.dvc' changed because it is 'modified'
100% Add                                                    1.00/1.00 [00:00<00:00,  4.60file/s]

To track the changes with git, run:

        git add artifacts.dvc
~/.../tdc/sample-dvc  master ●  git status
On branch master
Your branch is based on 'origin/master', but the upstream is gone.
  (use "git branch --unset-upstream" to fixup)
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   artifacts.dvc
        modified:   src/preprocessing.py
        modified:   src/train.py

no changes added to commit (use "git add" and/or "git commit -a")
```

Usando o dvc status, podemos ver que o conteúdo da pasta artifacts foi alterado

Usamos novamente o comando dvc add para adicionar as novas mudanças. Em seguida, podemos ver que o metafile artifacts.dvc foi modificado

creditas

Exemplo

Vamos criar uma tag para o primeiro experimento

```
 ~/.../tdc/sample-dvc  master ●  git add .
 ~/.../tdc/sample-dvc  master ●  git commit -m "Experiment 01"
[master a8bb033] Experiment 01
 3 files changed, 4 insertions(+), 2 deletions(-)
 ~/.../tdc/sample-dvc  master  git tag -a "experiment01" -m "Experiment 01"
 ~/.../tdc/sample-dvc  master  git tag
experiment00
experiment01
```

creditas

Exemplo

Vamos supor que já acabamos o experimento 1 e fizemos um segundo experimento. Para simular esse comportamento, vamos apenas trocar o conteúdo dos arquivos por version 2

```
~/.../tdc/sample-dvc  master ●  head src/*
==> src/preprocessing.py <==
version 2

==> src/train.py <==
version 2
~/.../tdc/sample-dvc  master ●  head artifacts/*
==> artifacts/data.csv <==
version 2

==> artifacts/model.pkl <==
version 2
```

Exemplo

```
~/.../tdc/sample-dvc  master ●  dvc add artifacts
WARNING: Output 'artifacts' of 'artifacts.dvc' changed because it is 'modified'
100% Add                                                    1.00/1.00 [00:00<00:00,  4.52file/s]

To track the changes with git, run:

        git add artifacts.dvc
~/.../tdc/sample-dvc  master ●  git status
On branch master
Your branch is based on 'origin/master', but the upstream is gone.
  (use "git branch --unset-upstream" to fixup)
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   artifacts.dvc
        modified:   src/preprocessing.py
        modified:   src/train.py

no changes added to commit (use "git add" and/or "git commit -a")
```

creditas

Exemplo

```
~/.../tdc/sample-dvc  master ●  git add .
~/.../tdc/sample-dvc  master ●  git commit -m "Experiment 02"
[master 0afd5ec] Experiment 02
 3 files changed, 4 insertions(+), 4 deletions(-)
~/.../tdc/sample-dvc  master  git tag -a "experiment02" -m "Experiment 02"
~/.../tdc/sample-dvc  master  git tag
experiment00
experiment01
experiment02
```

creditas

Exemplo

Vamos supor que queremos voltar no experimento 01. Basta ir até a tag correspondente

```
~/.../tdc/sample-dvc  master  git checkout experiment01
Note: checking out 'experiment01'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

  git checkout -b <new-branch-name>

HEAD is now at a8bb033... Experiment 01
```

creditas

Exemplo

Vemos que os arquivos da pasta src (versionados pelo git) já foram trocados. Entretanto, os arquivos versionados pelo dvc ainda não foram trocados.

```
~/.../tdc/sample-dvc ➦ a8bb033 head src/*
==> src/preprocessing.py <==
version 1

==> src/train.py <==
version 1
~/.../tdc/sample-dvc ➦ a8bb033 head artifacts/*
==> artifacts/data.csv <==
version 2

==> artifacts/model.pkl <==
version 2
```

Exemplo

Basta usar o comando dvc checkout para trazer os arquivos versionados pelo DVC

```
 ~/.../tdc/sample-dvc  ➦ a8bb033  dvc checkout
 ~/.../tdc/sample-dvc  ➦ a8bb033  head src/*
==> src/preprocessing.py <==
version 1

==> src/train.py <==
version 1
 ~/.../tdc/sample-dvc  ➦ a8bb033  head artifacts/*
==> artifacts/data.csv <==
version 1

==> artifacts/model.pkl <==
version 1
```

creditas

Comandos usados neste exemplo:

- dvc init
- dvc status
- dvc add <FOLDER_NAME>

Lista completa de comandos:

```
Available Commands:
  COMMAND         Use dvc COMMAND --help for command-specific help.
    init          Initialize DVC in the current directory.
    get           Download data from DVC repository.
    get-url       Download or copy files from URL.
    destroy       Remove DVC-files, local DVC config and data cache.
    add           Track data files or directories with DVC.
    remove        Remove DVC-file outputs.
    move          Rename or move a DVC controlled data file or a directory.
    unprotect     Unprotect data files or directories.
    run           Generate a stage file from a command and execute the command.
    repro         Check for changes and reproduce stages and dependencies.
    pull          Pull data files from a DVC remote storage.
    push          Push data files to a DVC remote storage.
    fetch         Fetch data files from a DVC remote storage.
    status        Show changed stages, compare local cache and a remote storage.
    gc            Collect unused data from DVC cache or a remote storage.
    import        Download data from DVC repository and take it under DVC control.
    import-url    Download or copy file from URL and take it under DVC control.
    config        Get or set config options.
    checkout      Checkout data files from cache.
    remote        Manage remote storage configuration.
    cache         Manage cache settings.
    metrics       Commands to add, manage, collect and display metrics.
    install       Install DVC git hooks into the repository.
    root          Relative path to the repository's directory.
    lock          Lock DVC-files.
    unlock        Unlock DVC-files.
    pipeline      Manage pipelines.
    commit        Save changed data to cache and update DVC-files.
    diff          Show a diff of a DVC controlled data file or a directory.
    version       Show DVC version and system/environment information.
    update        Update data artifacts imported from other DVC repositories.
```

Outros comandos

- dvc remote add -d s3remote s3://<BUCKET_URL> (https://dvc.org/doc/command-reference/remote)
- dvc push (https://dvc.org/doc/command-reference/push)
- dvc pull (https://dvc.org/doc/command-reference/pull)
- dvc run (https://dvc.org/doc/get-started/pipeline)

Referências

- https://dvc.org/
- https://github.com/iterative/dvc
- https://www.kdnuggets.com/2019/09/version-control-data-science-tracking-machine-learning-models-datasets.html

creditas

Conheça nossas vagas e canais de Product Technology e Agile

about.me/alexandreray

vagas.creditas.com.br

Nosso Twitter
@CreditasTech

Blog sobre Product Technology
medium.com/creditas-tech

Comunidade no Meetup
meetups Creditas

Linkedin e Instagram
Creditas Br

creditas