
test*cookieproj_rtdDocumentation*

Release 0.0.dev0

Victor Calderon <victor.calderon90@gmail.com>

May 25, 2018

Contents

1	Contents	3
1.1	Getting Started	3
1.2	Project Structure	6
1.3	Commands	7

Short Description of the very complicated project

Author: Victor Calderon (victor.calderon90@gmail.com)

This documentation is part of the repository [test_cookieproj_rtd](#).

1.1 Getting Started

Short Description of the very complicated project

Author: Victor Calderon (victor.calderon90@gmail.com)

1.1.1 Downloading and Installing Repository

Short Description of the very complicated project

Author: Victor Calderon (victor.calderon90@gmail.com)

Table of Contents

- *Downloading repository*

Downloading repository

This documentation is part of the repository `test_cookieproj_rtd`.

To download the repository to your computer, follow the following commands.

```
cd /path/to/where/you/want/to/download/repo
git clone https://github.com/vcalderon2009/test_cookieproj_rtd.git
cd test_cookieproj_rtd
```

The next step is to install and activate the project environment before being able to run any of the project's commands. See *Using the Project's environment* for more information.

Project based on the [modified version of cookiecutter data science project template](#)

1.1.2 Using the Project's environment

Short Description of the very complicated project

Author: Victor Calderon (victor.calderon90@gmail.com)

Table of Contents

- *Installing Environment & Dependencies*
 - *Show all available functions in the Makefile*
 - *Create environment*
 - *Activating the environment*
 - *Updating environment*
 - *Deactivating environment*
 - *Auto-activate environment*

Installing Environment & Dependencies

To use the scripts in this repository, you must have [Anaconda](#) installed on the systems that will be running the scripts. This will simplify the processes of installing all the dependencies.

For reference, see: [Manage Anaconda Environments](#)

The package counts with a **Makefile** with useful commands and functions. You must use this Makefile to ensure that you have all of the necessary *dependencies*, as well the correct **conda environment**.

Show all available functions in the Makefile

You can use the *Makefile* for running common tasks like *updating environments*, *cleaning extra files*, and more.

To show all available functions in the Makefile, run:

```
make show-help

Available rules:

clean                Deletes all build, test, coverage, and Python artifacts
clean-build          Remove build artifacts
clean-pyc             Removes Python file artifacts
clean-test           Remove test and coverage artifacts
environment          Set up python interpreter environment - Using environment.yml
lint                 Lint using flake8
remove_environment   Delete python interpreter environment
```

(continues on next page)

(continued from previous page)

```
test_environment    Test python environment is setup correctly
update_environment  Update python interpreter environment
```

Create environment

In order to properly run the commands of this project, you should install the **necessary packages** before. For this, you will to have installed **Anaconda**, because otherwise you will not be able to use this command.

The name of the environment and its dependencies are explicitly shown in the `environment.yml` file. In order to create the environment, you must run:

```
make environment
```

The main file that lists all of the dependencies for the project can be found as `environment.yml`.

Activating the environment

Once the environment has been **installed**, you can now *activate* the environment by typing

```
source activate test_cookieproj_rtd
```

Note: Depending on your installation of Anaconda, you might have to use the command:

```
conda activate test_cookieproj_rtd
```

instead.

Updating environment

You can always update the project's environment. The package dependencies are handled by the `environment.yml` file, and sometimes these packages need to be updated.

You can update the project's environments by running:

```
make update_environment
```

This will update the versions of each of the necessary packages.

Deactivating environment

Once you are done running the scripts of this project, you should **deactivate** the environment. To do so, run:

```
source deactivate
```

Note: Depending on your installation of Anaconda, you might have to use the command:

```
conda deactivate
```

instead.

Auto-activate environment

To make it easier to activate the necessary environment, one can use the `conda-auto-env` package, which **activates** the necessary environment **automatically**.

See the link above for more information!

Project based on the [modified version of cookiecutter data science project template](#)

Project based on the [modified version of cookiecutter data science project template](#)

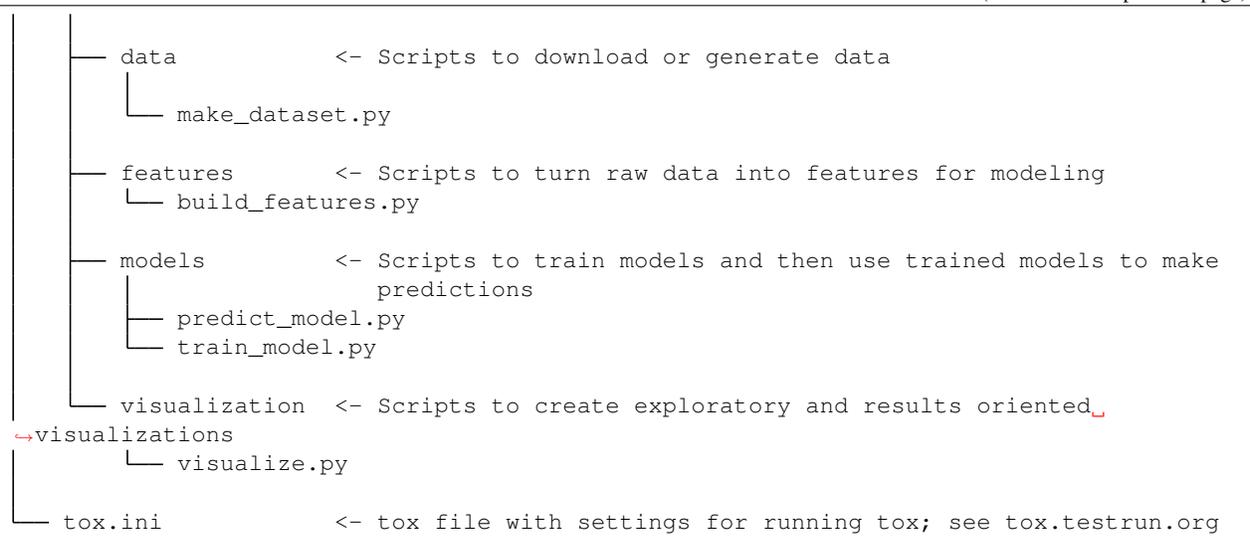
1.2 Project Structure

The organization of the project is the following:

— LICENSE	
— Makefile	<- Makefile with commands like `make data` or `make train`
— README.md	<- The top-level README for developers using this project.
— data	
— external	<- Data from third party sources.
— interim	<- Intermediate data that has been transformed.
— processed	<- The final, canonical data sets for modeling.
— raw	<- The original, immutable data dump.
— docs	<- A default Sphinx project; see sphinx-doc.org for details
— models	<- Trained and serialized models, model predictions, or model
↔ summaries	
— notebooks	<- Jupyter notebooks. Naming convention is a number (for
↔ ordering),	the creator's initials, and a short `-` delimited
↔ description, e.g.	`1.0-jqp-initial-data-exploration`.
— references	<- Data dictionaries, manuals, and all other explanatory
↔ materials.	
— reports	<- Generated analysis as HTML, PDF, LaTeX, etc.
— figures	<- Generated graphics and figures to be used in reporting
— requirements.txt	<- The requirements file for reproducing the analysis
↔ environment, e.g.	generated with `pip freeze > requirements.txt`
— environment.yml	<- The Anaconda environment requirements file for reproducing
↔ the analysis environment.	This file is used by Anaconda to create the project
↔ environment.	
— src	<- Source code for use in this project.
— __init__.py	<- Makes src a Python module

(continues on next page)

(continued from previous page)



Project based on the [modified version of cookiecutter data science project template](#)

1.3 Commands

The Makefile contains the central entry points for common tasks related to this project.

1.3.1 Syncing data to S3

- `make sync_data_to_s3` will use `aws s3 sync` to recursively sync files in `data/` up to `s3://[OPTIONAL] your-bucket-for-syncing-data` (do not include `'s3://'`)/`data/`.
- `make sync_data_from_s3` will use `aws s3 sync` to recursively sync files from `s3://[OPTIONAL] your-bucket-for-syncing-data` (do not include `'s3://'`)/`data/` to `data/`.

Project based on the [modified version of cookiecutter data science project template](#)