How to Make a Python Package Available via PIP

Peter Rochford

4024 Goss Rd

Fairfax, VA 22032

Created: 12/26/2016

Revised: 8/12/2022

Table of Contents

[Purpose 2](#_Toc142753166)

[Step 1: Structure Your Project 2](#_Toc142753167)

[Step 2: Write the setup.py File 3](#_Toc142753168)

[Step 3: Create a PyPI Account 5](#_Toc142753169)

[Step 4: Install Packaging Tools 5](#_Toc142753170)

[step 5: Create the Distribution Package 5](#_Toc142753171)

[Step 6 (Optional): Upload the Package to TestPyPI 5](#_Toc142753172)

[Step 7 (Optional): Install Your Package from TestPyPI 6](#_Toc142753173)

[Step 8: Upload the Package to PyPI 6](#_Toc142753174)

[Step 9: Install Your Package 6](#_Toc142753175)

[Project Files 8](#_Toc142753176)

[README.md 8](#_Toc142753177)

[Creating a LICENSE 8](#_Toc142753178)

[Creating a MANIFEST.in 8](#_Toc142753179)

[Generating Distribution Archives 8](#_Toc142753180)

[Configuring Your Project 9](#_Toc142753181)

[project.toml 9](#_Toc142753182)

[Releasing New Versions of Pip Package 10](#_Toc142753183)

[Troubleshooting 11](#_Toc142753184)

[No module named twine 11](#_Toc142753185)

[SSLError 12](#_Toc142753186)

[Filename has already been used 13](#_Toc142753187)

[File already exists 13](#_Toc142753188)

[FileNotFoundError UserDict.py 14](#_Toc142753189)

[HTTPError: 400 Bad Request from https://test.pypi.org/legacy/ 15](#_Toc142753190)

## Purpose

These notes describe the steps taken to make the OpenDSS\_SciVis Python library available to the Python community via the Python Packaging Index (PyPI). Creating a Python package and publishing it to PyPI allows others to easily install it using pip. Here's a step-by-step guide to help you package your GitHub project and make it available on PyPI.

## Step 1: Structure Your Project

Ensure that your project has the following structure:

* **your\_package** is the directory that contains your Python package code.
* **\_\_init\_\_.py** is an empty file that tells Python that this directory should be considered a Python package.
* **setup.py** is the script for building, distributing, and installing modules.
* **README.md** contains information about your project.
* **LICENSE.txt** contains the license information.

A suggested minimal project structure to use is:

YourPackage/

├── LICENSE.txt

├── README.md

├── setup.py

├── your\_package /

│ ├── \_\_init\_\_.py

│ └── example.py

└── tests/

The project structure currently used for the OpenDSS\_SciVis project is:

OpenDSS\_SciVis

├── Documents/

│ └── PIP notes.docx

├── LICENSE.txt

├── README.md

├── Examples/

│ ├── example1.py (first plot example)

│ ├── example2.py (second plot example)

│ └── … (other examples)

├── opendss\_scivis

│ ├── \_\_init\_\_.py

│ └── \*.py (source files)

└── Test/

## Step 2: Write the setup.py File

Create a setup.py file at the root of your project with the following content.

from setuptools import setup, find\_packages

setup(

name='my\_package',

version='0.1',

packages=find\_packages(),

install\_requires=[

'numpy',

'pandas'

],

author='Your Name',

author\_email='your.email@example.com',

description='A short description of your project.',

long\_description=open('README.md').read(),

long\_description\_content\_type='text/markdown',

url='https://github.com/your\_user\_name/your\_project\_name',

classifiers=[

'Development Status :: 3 - Alpha',

'Intended Audience :: Developers',

'License :: OSI Approved :: MIT License',

'Programming Language :: Python :: 3.6',

],

)

Modify the content as needed for your project. Replace placeholders like your\_project\_name and Your Name with the actual details of your project. Make sure to test your package locally before publishing it to PyPI to ensure everything works as expected. For the OpenDSS\_SciVis package the following setup.py file is used.

'''

A script to create a pip release of the OpenDSS\_SciVis package.

Description of the package can be found in the wiki:

https://github.com/PeterRochford/OpenDSS\_SciVis/wiki

Created on Aug 11, 2023

@author: Kevin Wu

'''

from setuptools import setup, find\_packages

setup(

name='OpenDSS\_SciVis',

version='1.0',

packages=find\_packages(),

install\_requires=[

'matplotlib',

'numpy',

'pandas',

'scipy',

'SkillMetrics'

],

author='Peter Rochford',

author\_email='peter.rochford@xatorcorp.com',

description='A Python package for analysis and visualization of data produced by the OpenDSS software application.',

long\_description=open('README.md').read(),

long\_description\_content\_type='text/markdown',

url='https://github.com/PeterRochford/OpenDSS\_SciVis/tree/main',

classifiers=[

'Development Status :: 3 - Alpha',

'Intended Audience :: Developers',

'License :: OSI Approved :: MIT License',

'Programming Language :: Python :: 3.11',

],

)

## Step 3: Create a PyPI Account

If you don't have a PyPI account, create one at [pypi.org](https://pypi.org/).

## Step 4: Install Packaging Tools

Install the required tools to build and distribute your package:

pip install setuptools

pip install twine

You need to have setuptools installed in order to run setup.py. setuptools is a package that provides infrastructure for building and distributing Python packages. It includes the setup() function that is used in the setup.py file to define the package metadata and dependencies.

You need twine to upload Python packages to PyPI. It provides a secure and reliable way to upload your package and make it available for others to install using pip.

## step 5: Create the Distribution Package

Navigate to your project directory and run the following command:

python setup.py sdist bdist\_wheel

This will create the distribution package in the dist directory. This command should output a lot of text and once completed should generate two files in the dist directory:

% ls -1 dist

OpenDSS\_SciVis-1.0.1.tar.gz

OpenDSS\_SciVis-1.0.1-py3-none-any.whl

When you run the setup.py script it will create two types of distribution packages:

1. Source Distribution (sdist): This is a compressed archive of your source code and can be used by users who want to build the package from source.
2. Wheel Distribution (bdist\_wheel): This is a pre-built binary distribution, and it's what most users will install. Wheel distributions are generally faster to install and can include compiled extensions.

The filenames will include the package name, version number, and extensions like .tar.gz (for the source distribution) and .whl (for the wheel distribution).

## Step 6 (Optional): Upload the Package to TestPyPI

Once you have built a distribution package, you should check its contents to make sure that everything is included and there are no errors or missing files. You can do this by extracting the contents of the .tar.gz file (for source distributions) or .whl file (for wheel distributions) and checking that all files are present.

If everything is fine, you can upload your package to PyPI using twine, which is a tool for uploading Python packages to PyPI. It is recommended you use TestPyPI before publishing to the main PyPI repository.

twine upload --repository-url https://test.pypi.org/legacy/ dist/\*

You'll be prompted to enter your PyPI username and password.

## Step 7 (Optional): Install Your Package from TestPyPI

After uploading your packages to Test PyPI, try installing them using pip from another virtual environment or machine. This will help ensure that installation works correctly for other users.

pip install –i <https://testpypi.python.org/pypi> OpenDSS\_SciVis

If you are upgrading the package then include the –upgrade option.

pip install –i <https://testpypi.python.org/pypi> OpenDSS\_SciVis --upgrade

## Step 8: Upload the Package to PyPI

if everything looks good on TestPyPi then go ahead and upload it on actual pypi server:

twine upload dist/\*

You'll be prompted to enter your PyPI username and password.

## Step 9: Install Your Package

Once successfully uploaded, anyone (including yourself) can install your package with:

pip install OpenDSS\_SciVis

This will generate a lot of output as shown below but once finished your GitHub project will be available on the target computer platform.

% pip install OpenDSS\_SciVIs

Defaulting to user installation because normal site-packages is not writeable

Collecting OpenDSS\_SciVIs

Downloading OpenDSS\_SciVis-1.0.1-py3-none-any.whl (40 kB)

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 40.6/40.6 kB 3.8 MB/s eta 0:00:00

Collecting scipy

Downloading scipy-1.11.1-cp310-cp310-macosx\_12\_0\_arm64.whl (29.6 MB)

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 29.6/29.6 MB 53.9 MB/s eta 0:00:00

Requirement already satisfied: matplotlib in /Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packages (from OpenDSS\_SciVIs) (3.5.3)

Requirement already satisfied: numpy in /Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packages (from OpenDSS\_SciVIs) (1.23.3)

Requirement already satisfied: pandas in /Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packages (from OpenDSS\_SciVIs) (1.4.4)

Requirement already satisfied: SkillMetrics in ./git/SkillMetrics (from OpenDSS\_SciVIs) (1.2.1)

Requirement already satisfied: fonttools>=4.22.0 in /Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packages (from matplotlib->OpenDSS\_SciVIs) (4.37.1)

Requirement already satisfied: python-dateutil>=2.7 in /Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packages (from matplotlib->OpenDSS\_SciVIs) (2.8.2)

Requirement already satisfied: packaging>=20.0 in /Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packages (from matplotlib->OpenDSS\_SciVIs) (21.3)

Requirement already satisfied: pillow>=6.2.0 in /Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packages (from matplotlib->OpenDSS\_SciVIs) (9.2.0)

Requirement already satisfied: cycler>=0.10 in /Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packages (from matplotlib->OpenDSS\_SciVIs) (0.11.0)

Requirement already satisfied: kiwisolver>=1.0.1 in /Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packages (from matplotlib->OpenDSS\_SciVIs) (1.4.4)

Requirement already satisfied: pyparsing>=2.2.1 in /Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packages (from matplotlib->OpenDSS\_SciVIs) (3.0.9)

Requirement already satisfied: pytz>=2020.1 in /Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packages (from pandas->OpenDSS\_SciVIs) (2022.2.1)

Requirement already satisfied: xlsxwriter>=3 in /Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packages (from SkillMetrics->OpenDSS\_SciVIs) (3.0.3)

Requirement already satisfied: six>=1.5 in /Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packages (from python-dateutil>=2.7->matplotlib->OpenDSS\_SciVIs) (1.16.0)

Installing collected packages: scipy, OpenDSS\_SciVIs

Successfully installed OpenDSS\_SciVIs-1.0.1 scipy-1.11.1

## Project Files

### README.md

The project uses the Markdown file (README.md) which appears to be most popular for GitHub projects. A README.md was created that follows the outline used for the corresponding file in the SkillMetrics GitHub project.

### Creating a LICENSE

It’s important for every package uploaded to the Python Package Index to include a license. A LICENSE.txt file containing a MIT License was created with the initial release of the OpenDSS\_SciVis package on 8/12/2023.

### Creating a MANIFEST.in

The MANIFEST.in file is used in Python packaging to define the list of files that should be included in the source distribution (sdist) but are not automatically included by default.

When you create a source distribution using the python setup.py sdist command, certain files are included automatically, such as Python source files (.py), the README, LICENSE, and setup.py. However, there might be additional files or directories that you want to include in the distribution, such as documentation, examples, data files, or configuration files.

The MANIFEST.in file allows you to define rules to include or exclude these additional files. Here's an example of what a MANIFEST.in file might look like:

include README.md

include LICENSE.txt

recursive-include myproject/data \*

In this example, we're including the README.md and LICENSE.txt files at the top level of our project directory, as well as all the contents of the myproject/data directory. The MANIFEST.in file provides a way to have fine-grained control over the contents of the source distribution, ensuring that users who install your package from source have all the necessary files.

When you run setup.py, setuptools will use the instructions in your MANIFEST.in file to determine which additional files should be included in the source distribution package.

If you're building a package that includes only standard Python files and basic README and LICENSE files, you might not need a MANIFEST.in. It's primarily useful when you have additional files that need to be part of the distribution.

### Generating Distribution Archives

Archives uploaded to PyPI can be installed by pip. Make sure you have the latest version of the Python Packaging (PyPA) build installed by running the command:

% python3 -m pip install --upgrade build

This command should output a lot of text and once completed should generate two files in the dist directory:

% ls -1 dist

SkillMetrics-1.2.1-py3-none-any.whl

SkillMetrics-1.2.1.tar.gz

## Configuring Your Project

A few files need to be created to have a package that can be distributed using hatchling via PyPI: <https://packaging.python.org/en/latest/tutorials/packaging-projects/>. The “Creating the package files” subsection outlines the files that need to be created.

packaging\_tutorial/

├── LICENSE

├── pyproject.toml

├── README.md

├── src/

│ └── example\_package\_YOUR\_USERNAME\_HERE/

│ ├── \_\_init\_\_.py

│ └── example.py

└── tests/

Since I had already hosted the Skill Metrics Python library as a Git repository on GitHub (<https://github.com/PeterRochford/SkillMetrics>), I created the following files in the top-level directory tree of the repository following the web page instructions:

* LICENSE.txt
* project.toml
* README.md
* MANIFEST.in

### project.toml

Listed below is how I set up the file to work with hatchling.

[build-system]

requires = ["hatchling"]

build-backend = "hatchling.build"

[project]

name = "SkillMetrics"

version = "1.2.1"

authors = [

{ name="PeterRochford", email="rochford.peter1@gmail.com" },

]

description = "A package for calculating skill of model predictions against observations"

readme = "README.md"

license = { file="LICENSE.txt" }

requires-python = ">=3.10"

dependencies = [

'matplotlib >= 3',

'numpy >= 1.23',

'pandas >= 1.4',

'xlsxwriter >= 3',

]

classifiers = [

"Programming Language :: Python :: 3",

"License :: OSI Approved :: GNU General Public License v3 (GPLv3)",

"Operating System :: OS Independent",

]

keywords = [

"meteorology",

"verification",

"weather",

]

[project.urls]

"Homepage" = "http://github.com/PeterRochford/SkillMetrics"

"Bug Tracker" = "://github.com/PeterRochford/SkillMetrics/issues"

## Releasing New Versions of Pip Package

1. If you have made changes to your distribution, increment the version number in the following two files:

/Users/Peter/git/OpenDSS\_SCiVis/opendss\_scivis/\_\_init\_\_.py

and edit the following line

\_\_version\_\_='1.2.2'

1. To find the version of a currently installed OpenDSS SciVis package, use the pip show command:

% pip show OpenDSS\_SciVIs

Name: OpenDSS-SciVis

Version: 1.0.1

Summary: A Python package for analysis and visualization of data produced by the OpenDSS software application.

Home-page: https://github.com/PeterRochford/OpenDSS\_SciVis/tree/main

Author: Peter Rochford

Author-email: peter.rochford@xatorcorp.com

License:

Location: /Users/Peter/Library/Python/3.10/lib/python/site-packages

Requires: matplotlib, numpy, pandas, scipy, SkillMetrics

Required-by:

1. Next check what version has been currently uploaded to the [Test PyPI](https://test.pypi.org/) site as this may differ from what has been released.

Graphical user interface, text, application, email

Description automatically generated

1. Next follow Step 5 to 9 above create and release the latest version.

## Troubleshooting

This section contains many solutions for troubleshooting problems in creating a pip package release. While many of them refer to earlier processes using setuptools and twine they are retained for future reference if needed.

### No module named twine

#### Problem

When trying to upload the OpenDSS\_SciVis package I get the following error.

% python3 -m twine upload --repository testpypi dist/\*

/Library/Frameworks/Python.framework/Versions/3.10/bin/python3: No module named twine

#### Solution

This occurs because the twine package is not installed.

% python3 -m pip install --upgrade twine

Collecting twine

Downloading twine-4.0.2-py3-none-any.whl (36 kB)

...

Installing collected packages: webencodings, zipp, rfc3986, Pygments, pkginfo, more-itertools, mdurl, docutils, bleach, requests-toolbelt, readme-renderer, markdown-it-py, jaraco.classes, importlib-metadata, rich, keyring, twine

Successfully installed Pygments-2.14.0 bleach-6.0.0 docutils-0.19 importlib-metadata-6.1.0 jaraco.classes-3.2.3 keyring-23.13.1 markdown-it-py-2.2.0 mdurl-0.1.2 more-itertools-9.1.0 pkginfo-1.9.6 readme-renderer-37.3 requests-toolbelt-0.10.1 rfc3986-2.0.0 rich-13.3.2 twine-4.0.2 webencodings-0.5.1 zipp-3.15.0

### SSLError

#### Problem

I encountered the following SSLError when trying to upload a Python package to either the PyPi or TestPyPi web sites when using setup.py or twine on an Apple computer running Mac OS X. This error appears to specifically occur when using twine with Python 2.

% twine upload -r testpypi dist/\*

Uploading distributions to https://test.pypi.org/legacy/

Uploading OpenDSS\_SciVis-1.1.5-py2-none-any.whl

SSLError: HTTPSConnectionPool(host='test.pypi.org', port=443): Max retries exceeded with url: /legacy/ (Caused by SSLError(SSLError(1, u'[SSL: TLSV1\_ALERT\_PROTOCOL\_VERSION] tlsv1 alert protocol version (\_ssl.c:590)'),))

I get a similar error when explicitly specifying the legacy repository

% twine upload --repository-url https://test.pypi.org/legacy/ dist/\*

Uploading distributions to https://test.pypi.org/legacy/

Enter your username: PeterRochford

Enter your password:

Uploading OpenDSS\_SciVis-1.1.5-py2-none-any.whl

SSLError: HTTPSConnectionPool(host='test.pypi.org', port=443): Max retries exceeded with url: /legacy/ (Caused by SSLError(SSLError(1, u'[SSL: TLSV1\_ALERT\_PROTOCOL\_VERSION] tlsv1 alert protocol version (\_ssl.c:590)'),))

It appears to arise because Mac OS X uses the pyOpenSSL 0.13.1 package that is incompatible with the pyOpenSSL versions being used by PyPi and TestPyPi. Also my Mac OS X does not permit me to upgrade pyOpenSSL.

#### Solution (No longer works as of 1/26/2019)

The only way I found to solve the problem was to install and use a virtual environment. First I installed the Python package and put virtualenv in /usr/local/bin.

% pip install --user virtualenv

% sudo /usr/bin/easy\_install virtualenv

Next I created a virtual environment called pypienv and started the virtual environment

% /usr/local/bin/virtualenv pypienv && cd pypienv

% source bin/activate.csh

Next I installed the required setup tools.

[pypienv] % pip install -U setuptools pip wheel

Requirement already up-to-date: setuptools in ./lib/python2.7/site-packages

Requirement already up-to-date: pip in ./lib/python2.7/site-packages

Requirement already up-to-date: wheel in ./lib/python2.7/site-packages

I also had to update the .pypirc file for the new repository URLs as given above in Uploading Your Project to PyPI.

Finally, upload the Python package to either the PyPi or TestPyPi web site.

% cd /Users/Peter/git/OpenDSS\_SciVis

% twine upload -r testpypi dist/\*

% twine upload --repository-url https://test.pypi.org/legacy/ dist/\*

Uploading distributions to https://upload.pypi.org/legacy/

Uploading OpenDSS\_SciVis-1.1.5-py2-none-any.whl

Uploading OpenDSS\_SciVis-1.1.5.tar.gz

### Filename has already been used

#### Problem

When trying to upload a new version of the Skill Metrics package I get the following error.

% python3 -m twine upload --repository-url https://test.pypi.org/legacy/ dist/\*

Enter your username: PeterRochford

Enter your password:

Uploading distributions to https://test.pypi.org/legacy/

Uploading OpenDSS\_SciVis-1.1.5-py2-none-any.whl

100%|█████████████████████████████████████████████████████████████████████████████████| 53.2k/53.2k [00:00<00:00, 72.7kB/s]

NOTE: Try --verbose to see response content.

HTTPError: 400 Client Error: This filename has already been used, use a different version. See https://test.pypi.org/help/#file-name-reuse for url: https://test.pypi.org/legacy/

#### Solution

This occurs because of the presence of old packages in the dist folder. Simply delete them or move them out of the folder.

% python3 -m twine upload --repository-url https://test.pypi.org/legacy/ dist/\*

Enter your username: PeterRochford

Enter your password:

Uploading distributions to https://test.pypi.org/legacy/

Uploading OpenDSS\_SciVis-1.1.6-py2-none-any.whl

100%|█████████████████████████████████████████████████████████████████████████████████| 57.7k/57.7k [00:00<00:00, 73.9kB/s]

Uploading OpenDSS\_SciVis-1.1.6.tar.gz

100%|█████████████████████████████████████████████████████████████████████████████████| 49.1k/49.1k [00:01<00:00, 36.4kB/s]

### File already exists

#### Problem

When trying to upload a replacement version of the Skill Metrics package I get the following error.

% sudo python3 -m twine upload --repository-url https://test.pypi.org/legacy/ dist/\*

Uploading distributions to https://test.pypi.org/legacy/

Enter your username: PeterRochford

Enter your password:

Uploading OpenDSS\_SciVis-1.1.8-py3-none-any.whl

100%|██████████████████████████████████████████████████████████████████████████████████| 69.5k/69.5k [00:00<00:00, 100kB/s]

NOTE: Try --verbose to see response content.

HTTPError: 400 Bad Request from https://test.pypi.org/legacy/

File already exists. See https://test.pypi.org/help/#file-name-reuse for more information.

#### Solution

This occurs because a file for that version already exists on the test.pypi.org site. Simply delete the packages from the site.

1. Go to the OpenDSS\_SciVis package:

[https://test.pypi.org/manage/project/OpenDSS\_SciVis/releases](https://test.pypi.org/manage/project/SkillMetrics/releases)

1. You'll see a listing like this one:

Graphical user interface, application

Description automatically generated

1. Point to "options" and click on "delete".

### FileNotFoundError UserDict.py

#### Problem

The following error occurred when trying to build the distribution using hatchling.

% python3 -m build

\* Creating venv isolated environment...

\* Installing packages in isolated environment... (hatchling)

\* Getting dependencies for sdist...

\* Building sdist...

\* Building wheel from sdist

\* Creating venv isolated environment...

\* Installing packages in isolated environment... (hatchling)

\* Getting dependencies for wheel...

\* Building wheel...

Traceback (most recent call last):

File "/Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packages/pep517/in\_process/\_in\_process.py", line 351, in <module>

main()

File "/Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packages/pep517/in\_process/\_in\_process.py", line 333, in main

json\_out['return\_val'] = hook(\*\*hook\_input['kwargs'])

File "/Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packages/pep517/in\_process/\_in\_process.py", line 249, in build\_wheel

return \_build\_backend().build\_wheel(wheel\_directory, config\_settings,

File "/private/var/folders/jy/p3021mrj5d54jd3wf686d0rc0000gp/T/build-env-tzuwjpjg/lib/python3.10/site-packages/hatchling/build.py", line 41, in build\_wheel

return os.path.basename(next(builder.build(wheel\_directory, ['standard'])))

File "/private/var/folders/jy/p3021mrj5d54jd3wf686d0rc0000gp/T/build-env-tzuwjpjg/lib/python3.10/site-packages/hatchling/builders/plugin/interface.py", line 144, in build

artifact = version\_api[version](directory, \*\*build\_data)

File "/private/var/folders/jy/p3021mrj5d54jd3wf686d0rc0000gp/T/build-env-tzuwjpjg/lib/python3.10/site-packages/hatchling/builders/wheel.py", line 318, in build\_standard

record = archive.add\_file(included\_file)

File "/private/var/folders/jy/p3021mrj5d54jd3wf686d0rc0000gp/T/build-env-tzuwjpjg/lib/python3.10/site-packages/hatchling/builders/wheel.py", line 59, in add\_file

file\_stat = os.stat(included\_file.path)

FileNotFoundError: [Errno 2] No such file or directory: '/private/var/folders/jy/p3021mrj5d54jd3wf686d0rc0000gp/T/build-via-sdist-z4709rdv/OpenDSS\_SciVis-1.1.9/mitmenv/lib/python2.7/UserDict.py'

ERROR Backend subprocess exited when trying to invoke build\_wheel

#### Solution

The problem occurs because of the presence of a mitmenv directory containing any old builds. To resolve the problem simply remove the directory and all its contents. Then run the build again using python3:

% python3 -m build

### HTTPError: 400 Bad Request from https://test.pypi.org/legacy/

#### Problem

The following error occurred when trying to upload the distribution using twine.

% python3 -m twine upload --repository testpypi dist/\*

Uploading distributions to https://test.pypi.org/legacy/

Uploading OpenDSS\_SciVis-1.2.1-py3-none-any.whl

100% ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 305.2/305.2 kB • 00:00 • 11.8 MB/s

WARNING Error during upload. Retry with the --verbose option for more details.

ERROR HTTPError: 400 Bad Request from https://test.pypi.org/legacy/

Invalid value for project\_urls. Error: Use valid URL.

#### Solution

The problem occurs because of an invalid URL in the [project.uls] section of your pyproject.toml file. For example, you may have forgotten the “http” for “Bug Tracker”.

% [project.urls]

"Homepage" = "[http://github.com/PeterRochford/OpenDSS\_SciVis](http://github.com/PeterRochford/SkillMetrics)"

"Bug Tracker" = "[://github.com/PeterRochford/OpenDSS\_SciVis/issues](http://github.com/PeterRochford/SkillMetrics/issues)"

To resolve the problem check that the URLs are correct by testing them in a web browser.

"Homepage" = "[http://github.com/PeterRochford/OpenDSS\_SciVis](http://github.com/PeterRochford/SkillMetrics)"

"Bug Tracker" = "[http://github.com/PeterRochford/OpenDSS\_SciVis/issues](http://github.com/PeterRochford/SkillMetrics/issues)"

Then build the package again using hatchling:

% python3 -m build

\* Creating venv isolated environment...

\* Installing packages in isolated environment... (hatchling)

\* Getting build dependencies for sdist...

\* Building sdist...

\* Building wheel from sdist

\* Creating venv isolated environment...

\* Installing packages in isolated environment... (hatchling)

\* Getting build dependencies for wheel...

\* Building wheel...

Successfully built OpenDSS\_SciVis-1.2.1.tar.gz and OpenDSS\_SciVis-1.2.1-py3-none-any.whl