Your new virtual environment will use Python 3.7 and accompanying Anaconda packages. After creating the new virtual environment, you'll install the imbalanced-learn library in that environment.

**NOTE**

Consult the [imbalanced-learn documentation](https://imbalanced-learn.readthedocs.io/en/stable/) for additional information about the imbalanced-learn library.

**macOS Setup**

Before we create a new environment in macOS, we'll need to update the global conda environment:

1. If your PythonData environment is activated when you launch the command line, deactivate the environment.

**REWIND**

To deactivate an active environment, type conda deactivate.

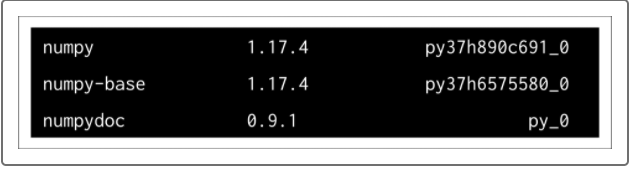
1. Update the global conda environment by typing conda update conda and press Enter.
2. After all the packages are collected, you'll see the prompt Proceed ([y]/n)?. Press the "Y" key (for "yes") and press Enter.
3. In the command line, type conda create -n mlenv python=3.7 anaconda. The name of your new environment is mlenv.
4. After all the packages are collected, you'll see the prompt Proceed ([y]/n)?. Press the "Y" key (for "yes") and press Enter.
5. Activate your mlenv environment by typing conda activate mlenv and press Enter.

**Check Dependencies for the imbalanced-learn Package**

Before we install the imbalanced-learn package, we need to confirm that all of the package dependencies are satisfied in our mlenv environment:

* NumPy, version 1.11 or later
* SciPy, version 0.17 or later
* Scikit-learn, version 0.21 or later

On the command line, you can check all packages that begin with numpy, scipy, and scikit-learn when you type conda list | grep and press Enter. The grep command will search for patterns of the text numpy in our conda list. For example, when we type conda list | grep numpy and press Enter, the output should be as follows:



As you can see, our numpy dependency meets the installation requirements for the imbalanced-learn package.

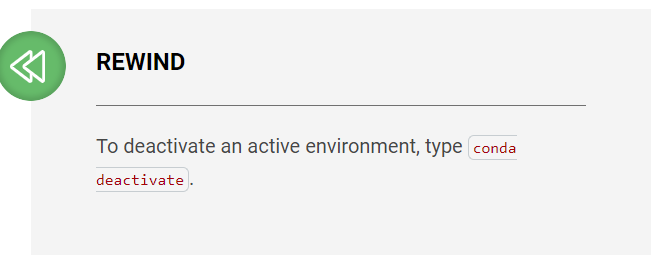
Additionally, you can type python followed by the command argument -c, and then "import `package\_name`;print(`package\_name`.\_\_version\_\_)" to verify which version of a package is installed in an environment, where `package\_name` is the name of the package you want to verify.

Type python -c "import numpy ;print(numpy.\_\_version\_\_)" and then press Enter to see the version of numpy in your mlenv environment.

## Windows Setup

Before we create a new environment in Windows, we'll need to update the global conda environment:

1. Launch the Anaconda Prompt, or open your PythonData Anaconda Prompt and deactivate this environment.



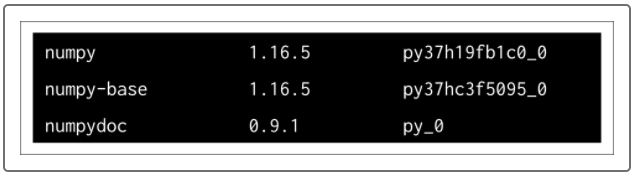
1. Update the global conda environment by typing conda update conda and press Enter
2. After all the packages are collected, you'll see the prompt Proceed ([y]/n)?. Press the "Y" key (for "yes") and press Enter.
3. In the command line, type conda create -n mlenv python=3.7 anaconda.
4. After all the packages are collected, you'll see the prompt Proceed ([y]/n)?. Press the "Y" key (for "yes") and press Enter.
5. Activate your mlenv environment by typing conda activate mlenv and press Enter, or open your Anaconda Prompt (mlenv).

**Check Dependencies for the imbalanced-learn Package**

Before we install the imbalanced-learn package, we need to confirm that all of the package dependencies are satisfied in our mlenv environment:

* NumPy, version 1.11 or later
* SciPy, version 0.17 or later
* Scikit-learn, version 0.21 or later

In the Anaconda Prompt, you can check all packages that begin with numpy, scipy, and scikit-learn when you type conda list | findstr and press Enter. The findstr command will search for patterns of the text in our conda list. For example, when we type conda list | findstr numpy and press Enter, the output should be as follows:



From the output, we can see that our numpy dependency meets the installation requirements for the imbalanced-learn package.

Additionally, you can type python followed by the command argument -c, and then "import `package\_name`;print(`package\_name`.\_\_version\_\_)" to verify which version of a package is installed in an environment, where `package\_name` is the name of the package you want to verify:

Type python -c "import numpy;print(numpy.\_\_version\_\_)" and press Enter to see the version of numpy in your mlenv environment.

**NOTE**

By updating our global conda environment, the numpy, scipy, and scikit-learn dependencies should meet the requirements for installing the imbalanced-learn package.

**NOTE**

If you encounter an error while starting Jupyter Notebook, you may need to install the environment\_kernels module with pip install environment\_kernels

## Install the imbalanced-learn Package

Now that our dependencies have been met, we can install the imbalanced-learn package in our mlenv environment.

With the mlenv environment activated, either in the Terminal in macOS or in the Anaconda Prompt (mlenv) in Windows, type the following:

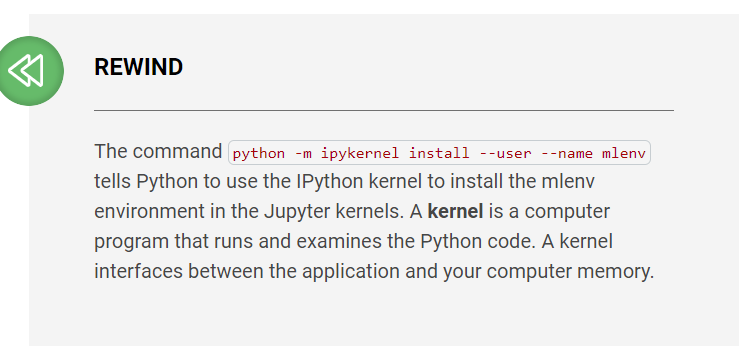
conda install -c conda-forge imbalanced-learn

Then press Enter.

After all the packages are collected, you'll see the prompt Proceed ([y]/n)?. Press the "Y" key (for "yes") and press Enter.

## Add the Machine Learning Environment to Jupyter Notebook

To use the mlenv environment we just created in the Jupyter Notebook, we need to add it to the kernels. In the command line, type python -m ipykernel install --user --name mlenv and press Enter.



To check if the mlenv is installed, launch the Jupyter Notebook and click the "New" dropdown menu:

