17.1.1

## Create a Machine Learning Environment

**The** lead data scientist, Jill, has asked you to create the same development environment that she is using. This step will help you run the code smoothly without conflicts.

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.

Check out the macOS instructions below, or jump to the **Windows instructions**.

# 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`.

2. Update the global conda environment by typing `conda update conda` and press Enter.

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

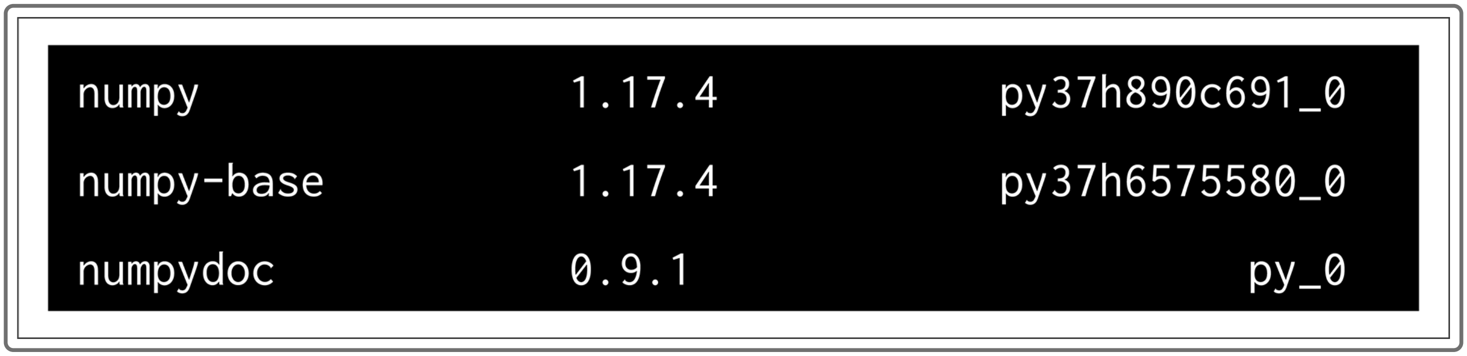
## Check Dependencies for the imbalanced-learn Package

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

Retake

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:



```
numpy                1.17.4           py37h890c691_0
numpy-base           1.17.4           py37h6575580_0
numpydoc             0.9.1                      py_0
```

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.

> **REWIND**
>
> To deactivate an active environment, type `conda deactivate`.

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

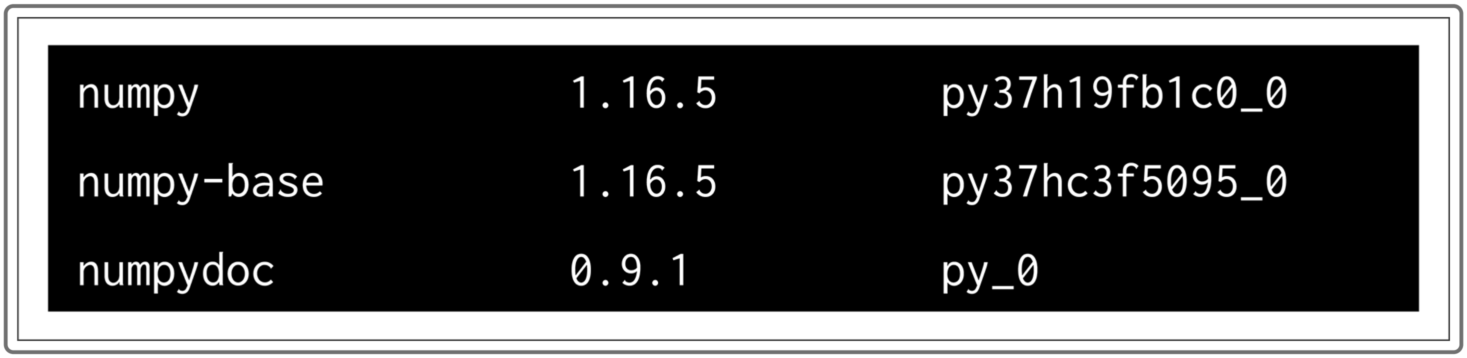
## Check Dependencies for the imbalanced-learn Package

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

Retake

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:



```
numpy            1.16.5       py37h19fb1c0_0
numpy-base       1.16.5       py37hc3f5095_0
numpydoc         0.9.1        py_0
```

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.

**SKILL DRILL**

Determine what version of `scipy` and `scikit-learn` you have in your mlenv environment.

**Hint:** For Windows, you may need to quit the mlenv Anaconda Prompt and launch it again to use `python -c "import <package_name>;print(<package_name>.__version__)"`.

**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.

Retake

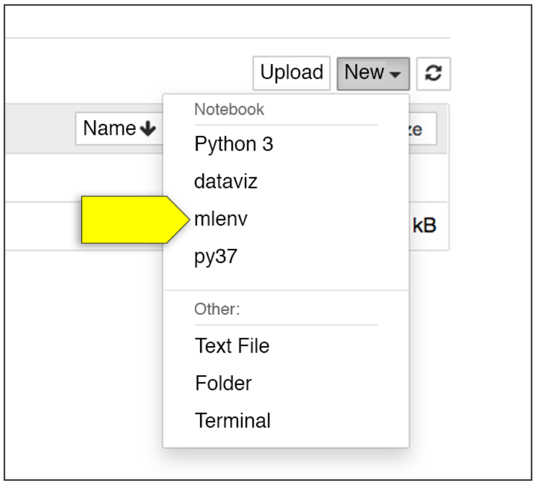
## Add the Machine Learning Environment to Jupyter Notebook

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.

**REWIND**

The command `python -m ipykernel install --user --name mlenv` tells Python to use the IPython kernel to install the mlenv environment in the Jupyter kernels. A **kernel** is a computer program that runs and examines the Python code. A kernel interfaces between the application and your computer memory.

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



Now we can begin our machine learning journey.