

SETTING A ML DEV ENVIRONMENT

Welcome to Machine Learning-I course in George Brown College. The first logical step to learn Machine learning is to make sure we have our environment set up correctly. This is what we will do here. Setting up our development environment right. There are so many options we could accomplish this with. I will chart out the easiest (beginner) way, minimalistic (minimal footprint) way and an effective (advanced CLI users) way.

Before we set up the environment, we need to see what are the minimal requirements (hardware and software) required to have a decent environment for doing machine learning.

Hardware Requirements

First, to perform machine learning and deep learning on any dataset, the software/program requires a computer system powerful enough to handle the computing power necessary. So, the following is required:

- 1. Central Processing Unit (CPU)** — Intel Core i5 6th Generation processor or higher. An AMD equivalent processor will also be optimal. Generally, a high clock speed CPU would be required.
- 2. RAM** — 8 GB bare minimum, 16 GB or higher is recommended for big datasets and to run Deep Learning training. The more the merrier, if you can afford.
- 3. Operating System** — Ubuntu or Mac or Microsoft Windows 10. I recommend updating Windows 10 to the latest version before proceeding forward.



4. Graphics Processing Unit (GPU) [OPTIONAL] — NVIDIA

GeForce GTX 960 or higher. AMD GPUs are not able to perform deep learning regardless. For more information on NVIDIA GPUs for deep learning please visit

<https://developer.nvidia.com/cuda-gpus>.

Two free options for GPU:

1. Kaggle

You can use either Kaggle CPU, GPU and TPU Kernels for free (limited-use per week).

Pros: Free, Reliable

Cons: Limited use (30 hours per week)

2. Google Colab

You can use Google Colab CPU, GPU for free (limited-hour-usage).

Pros: Free, Good to run tasks with small training times

Cons: Boots you out after certain hours in the middle of long-running training tasks. (Read as REAL PAIN!!!!)

Note: In the case of laptops, the ideal expensive option would be to purchase a gaming laptop from any vendor deemed suitable such as Alienware, ASUS, Lenovo Legion, Acer Predator etc.



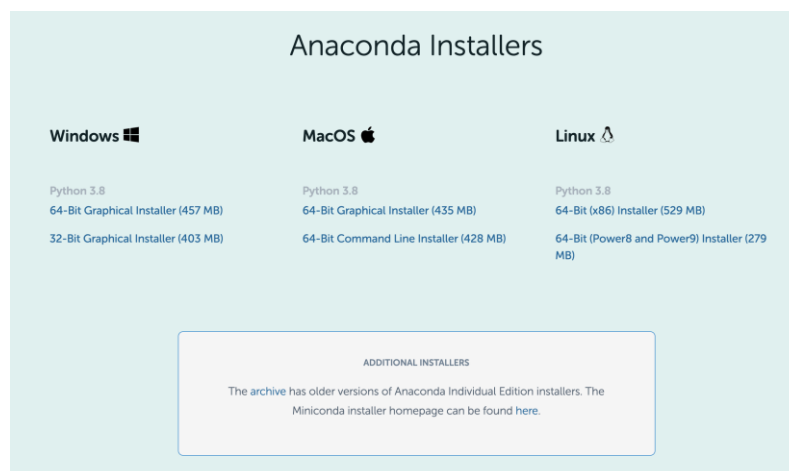
Mac Users

For the MAC users, these are the options that are available.

1. Quick way (Beginners) - Anaconda
2. Minimalist way (Beginners) - Miniconda
3. Effective way (Advanced) - Virtualenv

Quick way (Beginners) - Anaconda

Goto <https://www.anaconda.com/products/individual#Downloads>



At this point in time, you have two options.

1. Graphical Installation (Beginners)
2. Command Line Installation (Advanced)

I will show you both for completeness here. Please be discreet in choosing one that works for you.

Choose 64-bit Graphical Installer under MacOS.

Double-click on the downloaded pkg file.

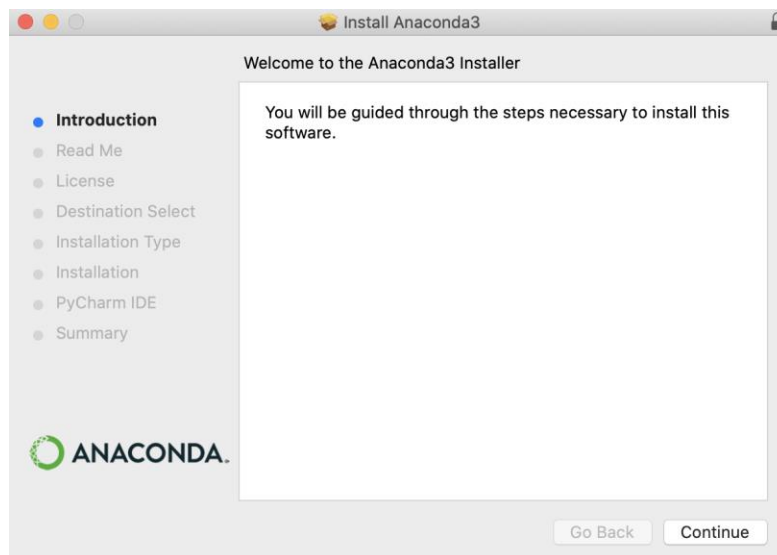


Anaconda3-2020.11-MacOSX-x86_64.pkg
Installer package - 456.5 MB

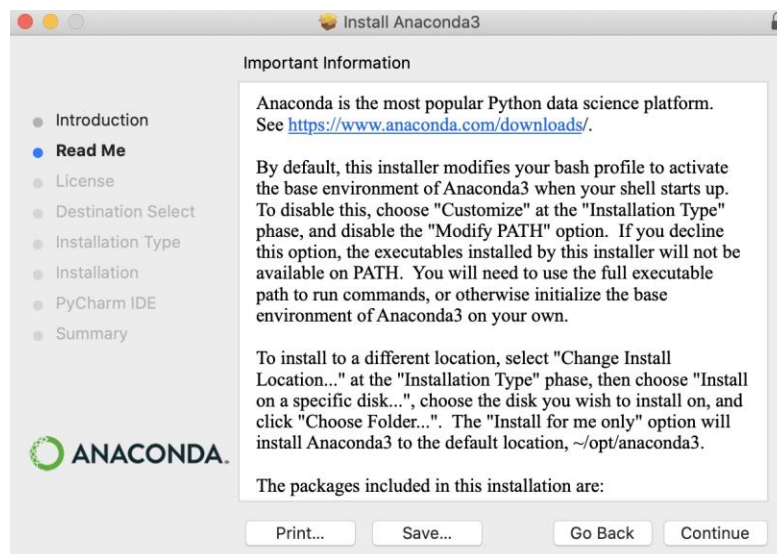


Follow these instructions as shown below in a series of steps.

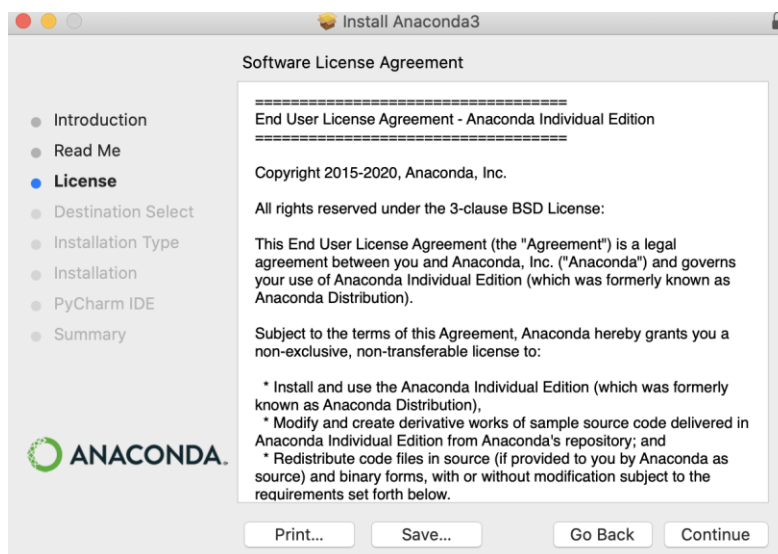
Introduction step



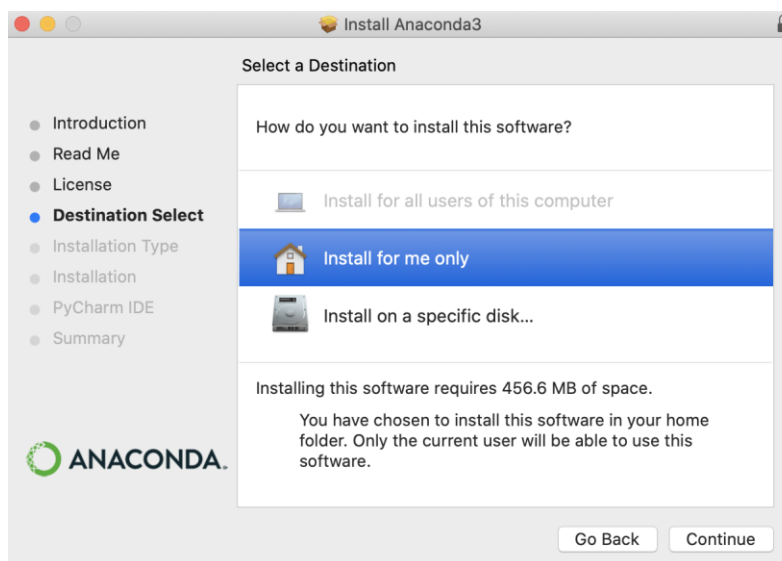
Read Me step



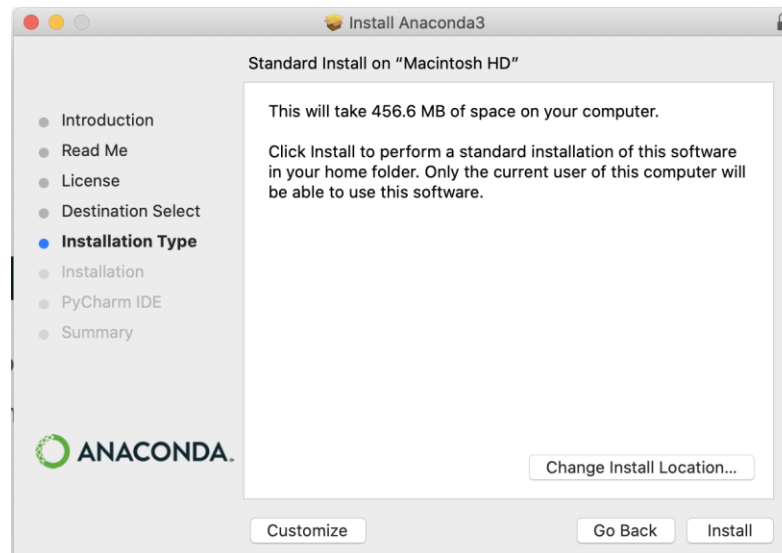
License Agreement



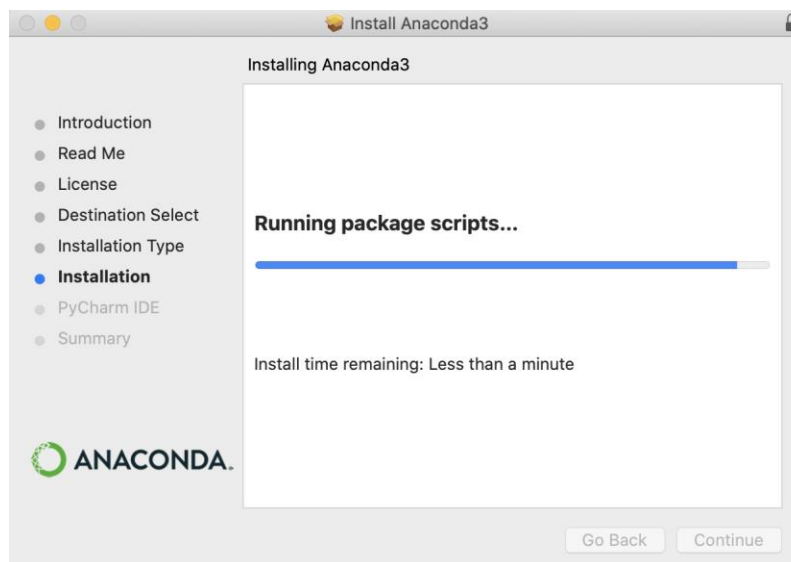
Select Destination folder



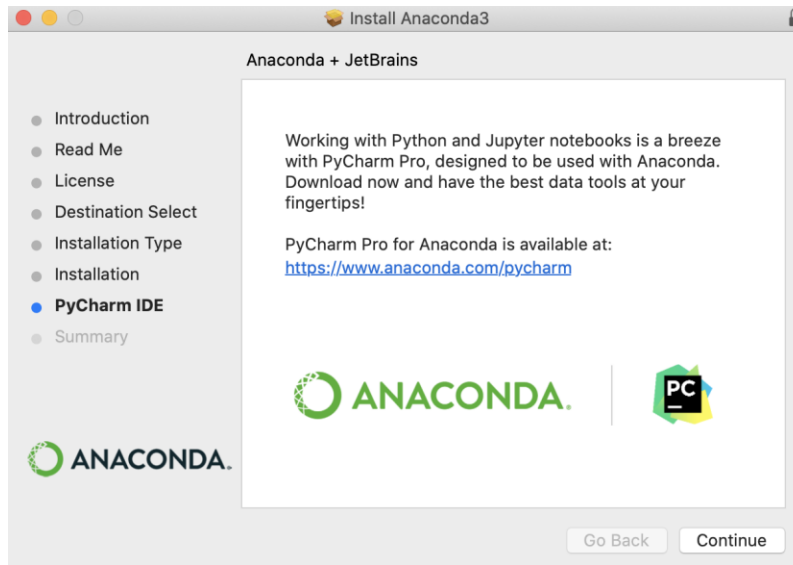
Select Type of Installation



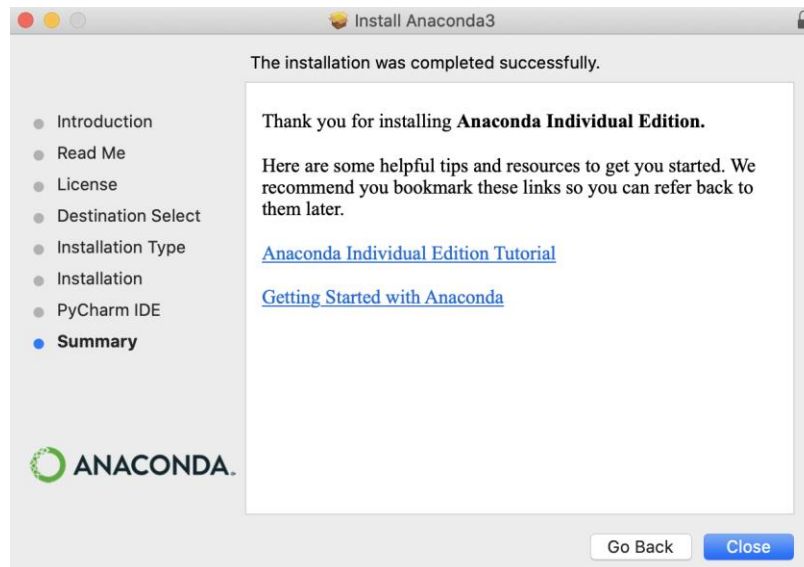
Installation Step (Please note that it will take anywhere between a minute to a couple of minutes)



PyCharm IDE Step



Summary



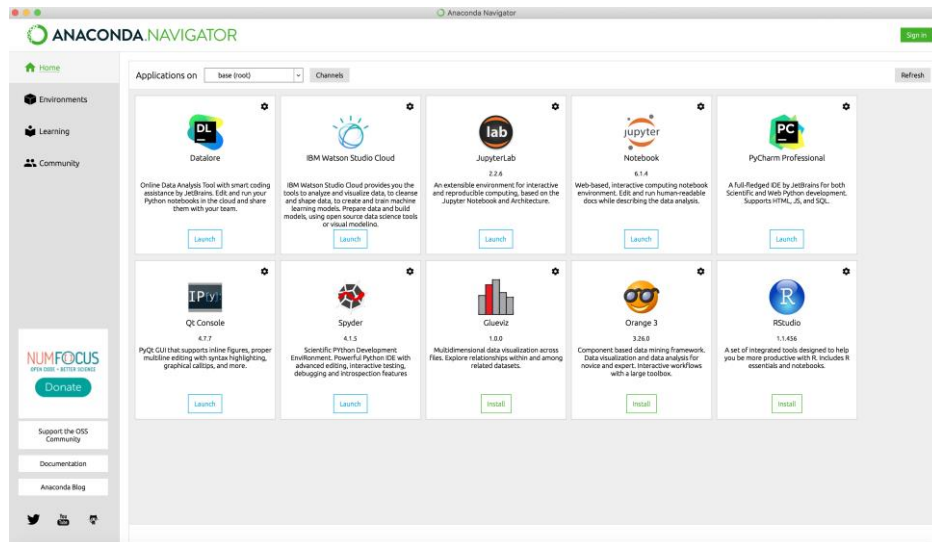
Go to Applications Folder and find the Anaconda-Navigator



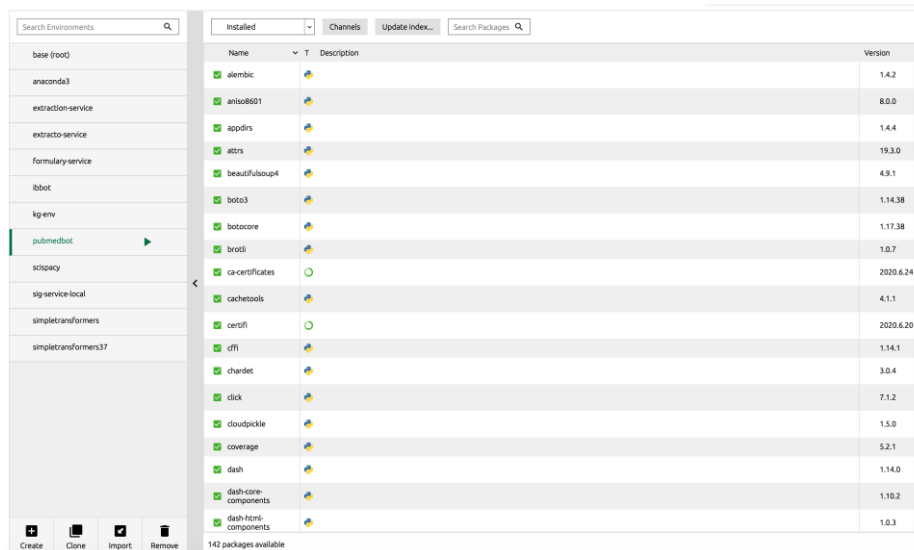
Launch the Anaconda Navigator by clicking on this icon.



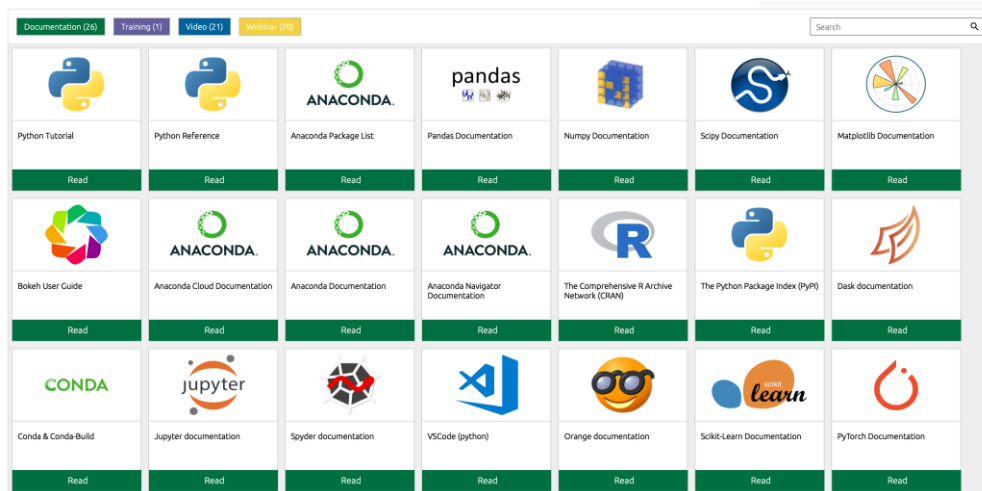
Home page



Environments (Please note you will have only base environment in your machine)



Learning



In Home page, click on Jupyter Notebook icon.

>>>>> Jupyter Notebook Launched >>>>>>>

Sanity Check in your Command Line Interface for successful Anaconda installation.

Go to Terminal and type

```
conda -version
```

As of December 2020, latest conda version is 4.9.2

Create a conda environment

```
conda create --name test python=3.7
```

Activate the created environment

```
conda activate test
```



Look for packages to install in this newly created environment

```
conda search scikit-learn  
conda install scikit-learn
```

Test your environment by using importing libraries by import statements

```
python  
import numpy as np  
import sklearn  
import scipy
```

Lastly, if you want to come out of the working conda environment, type

```
conda deactivate
```

Congratulations. You have Anaconda installed successfully in your local machine. Now you can tackle all the problems in the world. (I wish!)



Miniconda – for a lesser footprint

Go to <https://docs.conda.io/en/latest/miniconda.html>

Miniconda

Miniconda is a free minimal installer for conda. It is a small, bootstrap version of Anaconda that includes only conda, Python, the packages they depend on, and a small number of other useful packages, including pip, zlib and a few others. Use the `conda install` command to install 720+ additional conda packages from the Anaconda repository.

See if Miniconda is right for you.

Windows installers

Windows

Python version	Name	Size	SHA256 hash
Python 3.8	Miniconda3 Windows 64-bit	57.0 MiB	4fa220ba497babb0d6608c8843545372a99f5331c8128099aecd883f627c61
	Miniconda3 Windows 32-bit	54.2 MiB	9c2ef78bae97246c85c286733ca38f61feb8a463f98a2a511fca881ce7ebc661
Python 2.7	Miniconda2 Windows 64-bit	54.1 MiB	697382548483294a874b782bda8c4594980e0d4787051baa87b0a4b7326c
	Miniconda2 Windows 32-bit	47.7 MiB	c8849c26f8b4954057bc04e99ad7261ff613f4a60218e464150443762617b

MacOSX installers

MacOSX

Python version	Name	Size	SHA256 hash
Python 3.8	Miniconda3 MacOSX 64-bit bash	54.5 MiB	a9e8a7ba555d872e813236495b649ea8f44ce0a098f04c35706139f6478
	Miniconda3 MacOSX 64-bit pkg	62.0 MiB	586f3a73c7ffa8053695c9c3b0ba050f583bc784764508d744927154085e317fa
Python 2.7	Miniconda2 MacOSX 64-bit bash	40.3 MiB	6e2961e28a2239c148764563880e6a638f8c860828c2b6f1878c30840988045
	Miniconda2 MacOSX 64-bit pkg	48.4 MiB	9ca4313e8162a939c7a5a4f48d657722594f8b9a98472803063c3a7f66fa1da

Linux installers

Linux

Python version	Name	Size	SHA256 hash
Python 3.8	Miniconda3 Linux 64-bit	89.9 MiB	1314098489f15468276794accfc9844611514a5a72fe1771ab83e870e9584a7
	Miniconda3 Linux 32-bit	62.7 MiB	f387ede03fa4dc318407775e62d59803638285c2758ab8b64c27144adafcc4
Python 2.7	Miniconda2 Linux 64-bit	48.7 MiB	b8280de1a8ba868c4c948fedaace7380a252b33050ef0878a1504a817476b8979
	Miniconda2 Linux 32-bit	39.0 MiB	2e28ac4379ca5262a7612f84ad26b1a2f272f8208994facdecb28e8af51749979

Download the file depending on your OS.

Now, you will have the Python interpreter itself, along with a command-line tool called conda which operates as a crossplatform package manager geared toward Python packages, similar in spirit to the apt or yum tools that Linux users might be familiar with.

Install only required libraries as and when you need them.

```
conda install ipython-notebook
```

```
conda install numpy
```

```
conda install scipy
```

```
conda install matplotlib
```

```
conda install pandas
```

```
conda install sklearn
```



Test your environment by using importing libraries by import statements

python

```
import numpy as np  
import sklearn  
import scipy
```

Lastly, if you want to come out of the working conda environment, type

```
conda deactivate
```

Now, you will have a minimalistic miniconda (not with all the heavy batteries... just minimal libraries installed in them).

Further Readings

<https://conda.io/projects/conda/en/latest/user-guide/getting-started.html>



Another easy way for experienced CLI users – virtualenv

Creating a virtual environment

```
virtualenv {environment_name} -p python3.7
```

In Terminal, type

```
virtualenv ml-venv -p python3.7
```

Activate the environment

```
source ml-venv/bin/activate
```

Create a requirements.txt file with the following libraries as contents

numpy

scipy

pandas

matplotlib

scikit-learn

```
pip install -r requirements.txt
```



Test your environment by using importing libraries by import statements

python

```
import numpy as np
import sklearn
import scipy
import pandas as pd
```

Lastly, if you want to come out of the working virtual environment, type

```
deactivate
```

Congratulations. You've installed your own custom virtual environment with minimal footprints that are easy to port and deploy in other cloud environments.



Windows Users

Windows users follow these steps. Caution: I don't have a Windows machine. But I can help you with troubleshooting installation issues.

Steps

1. Download Anaconda
2. Install Anaconda & Python
3. Start and Update Anaconda
4. Create an Anaconda Environment
5. Install essential libraries

Step 1: Download Anaconda

In this step, we will download the Anaconda Python package for your platform.

Anaconda is a free and easy-to-use environment for scientific Python.

- 1. Install Anaconda (Python 3.7 version) [Download](#)

Step 2: Install Anaconda

In this step, we will install the Anaconda Python software on your system. Installation is very easy and quick once you download the setup. Open the setup and follow the wizard instructions.

#Note: *It will automatically install Python and some basic libraries*

It might take 5 to 20 minutes according to your system.



Step 3: Update Anaconda

Open Anaconda Prompt to type the following command(s). Don't worry Anaconda Prompt is just works same as cmd.

```
conda update conda  
conda update --all
```

Step 4: Create an Anaconda Environment

Here we will create a new anaconda environment for our specific usage so that it will not affect the root of Anaconda.

Open Anaconda Prompt to type the following commands.

Create a conda environment named “ml-venv” (you can change the name) by invoking the following command:

```
conda create -n ml-venv pip python=3.7
```

Activate the conda environment by issuing the following command:

```
conda activate ml-venv  
(ml-venv)C:> # Your prompt should change
```

Create a requirements.txt file with following libraries as contents

numpy

scipy

pandas

matplotlib

scikit-learn

```
pip install -r requirements.txt
```



Test your environment by using importing libraries by import statements

python

```
import numpy as np
import sklearn
import scipy
import pandas as pd
```

Lastly, if you want to come out of the working virtual environment, type

```
conda deactivate
```

Congratulations! 😊 You have successfully created the environment on Windows!

