Project Prerequisites

Anaconda Navigator:

Anaconda Navigator is a desktop graphical user interface that comes with the Anaconda distribution and allows us to run programs and manage anaconda packages, environments, and channels without having to use command-line commands. Packages can be found on Anaconda.org or in a local Anaconda Repository using Navigator. It's compatible with Windows, Mac OS X, and Linux. Many scientific packages rely on certain versions of other programs to run.

Anaconda is a package and environment manager that can be run from the command line. This assists data scientists in ensuring that each version of each package has all of the dependencies it needs and functions properly.

Navigator is a point-and-click interface for working with packages and environments that eliminates the need to type anaconda instructions into a terminal window. We may use it to search packages, install them in an environment, execute them, and update them all from within the navigator.

Anaconda is an open-source software package that includes Jupyter, spyder, and other tools for large-scale data processing, data analytics, and scientific computing. R and Python are supported by Anaconda.

Scikit-Learn (Sklearn)

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon **NumPy, SciPy** and **Matplotlib**.

Installing scikit-learn:

There are different ways to install scikit-learn:

- Install the latest official release. This is the best approach for most users. It will provide a stable version and pre-built packages are available for most platforms.
- Install the version of scikit-learn provided by your operating system or Python distribution. This is a quick option for those who have operating systems or Python distributions that distribute scikit-learn. It might not provide the latest release version.
- Building the package from source. This is best for users who want the latest-and-greatest features and aren't afraid of running brand-new code. This is also needed for users who wish to contribute to the project.

NumPy in Python:

NumPy in Python is a library that is used to work with arrays and was created in 2005 by Travis Oliphant. NumPy library in Python has functions for working in domain of Fourier transform, linear algebra, and matrices. Python NumPy is an open-source project that can be used freely. NumPy stands for Numerical Python.

How to install NumPy Python?

Installing the NumPy library is a straightforward process. You can use pip to install the library.Go to the command line and type the following:

pip install numpy

If you are using Anaconda distribution, then you can use conda to install NumPy. conda install numpy

Once the installation is complete, you can verify it by importing the NumPy library in the python interpreter. One can use the numpy library by importing it as shown below.

import numpy

If the import is successful, then you will see the following output.

```
>>> import numpy

>>> numpy.__version__

'1.17.2'
```

Pandas:

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data.

In 2008, developer Wes McKinney started developing pandas when in need of high performance, flexible tool for analysis of data.

Prior to Pandas, Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data — load, prepare, manipulate, model, and analyze.

Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Matplotlib

Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy. As such, it offers a viable open source alternative to MATLAB. Developers can also use matplotlib's APIs (Application Programming Interfaces) to embed plots in GUI applications.
A Python matplotlib script is structured so that a few lines of code are all that is required in most instances to generate a visual data plot. The matplotlib scripting layer overlays two APIs:

- The pyplot API is a hierarchy of Python code objects topped by *matplotlib.pyplot*
- An OO (Object-Oriented) API collection of objects that can be assembled with greater flexibility than pyplot. This API provides direct access to Matplotlib's backend layers.

Matplotlib and Pyplot in Python

The pyplot API has a convenient MATLAB-style stateful interface. In fact, matplotlib was originally written as an open source alternative for MATLAB. The OO API and its interface is more customizable and powerful than pyplot, but considered more difficult to use. As a result, the pyplot interface is more commonly used, and is referred to by default in this article.

Understanding matplotlib's pyplot API is key to understanding how to work with plots:

- ***matplotlib.pyplot.figure: Figure*** is the top-level container. It includes everything visualized in a plot including one or more ***Axes***.
- ***matplotlib.pyplot.axes****:* ***Axes*** contain most of the elements in a plot**:** ***Axis, Tick, Line2D, Text,*** etc., and sets the coordinates. It is the area in which data is plotted. Axes include the X-Axis, Y-Axis, and possibly a Z-Axis, as well.

For more information about the pyplot API and interface, refer to ***What Is Pyplot In Matplotlib***

Installing Matplotlib

Matplotlib and its dependencies can be downloaded as a binary (pre-compiled) package from the Python Package Index (PyPI), and installed with the following command:

```
python -m pip install matplotlib
```

Flask:

Flask is a web framework that provides libraries to build lightweight web applications in python. It is developed by **Armin Ronacher** who leads an international group of python enthusiasts (POCCO). It is based on WSGI toolkit and jinja2 template engine. Flask is considered as a micro framework.

WSGI:

It is an acronym for web server gateway interface which is a standard for python web application development. It is considered as the specification for the universal interface between the web server and web application.

Jinja2:

Jinja2 is a web template engine which combines a template with a certain data source to render the dynamic web pages.

Installation of Flask Environment:

To install flask on the system, we need to have python 2.7 or higher installed on our system. However, we suggest using python 3 for the development in the flask.

Install virtual environment (virtualenv)

virtualenv is considered as the virtual python environment builder which is used to create the multiple python virtual environment side by side. It can be installed by using the following command.

```
$ pip install virtualenv
```

Once it is installed, we can create the new virtual environment into a folder as given below.

1. $ mkdir new
2. $ cd new
3. $ virtualenv venv

To activate the corresponding environment, use the following command on the Linux operating system.

```
$ venv/bin/activate
```

On windows, use the following command.

```
$ venv\scripts\activate
```

We can now install the flask by using the following command.

```
$ pip install flask
```

However, we can install the flask using the above command without creating the virtual environment.

To test the flask installation, open python on the command line and type python to open the python shell. Try to import the package flask.

Jupyter Notebooks:

Jupyter Notebooks can be used in the same way. Jupyter Notebooks are a popular system that allows us to integrate our code, descriptive text, output, graphics, and interactive interfaces into a single notebook file that we can edit, view, and use in a web browser.

The *Jupyter Notebook* is a server-client application that allows editing and running notebook documents via a web browser. The *Jupyter Notebook* can be executed on a local desktop requiring no internet access (as described in this document) or can be installed on a remote server and accessed through the internet.

In addition to displaying/editing/running notebook documents, the *Jupyter Notebook* has a “Dashboard” (Notebook Dashboard), a “control panel” showing local files and allowing to open notebook documents or shutting down their kernels.

Installation:

Step 1: The browser

Step “zero” consists in installing a modern standard-compliant browser. Either Mozilla Firefox or Google Chrome will work well. Try to avoid MS Explorer.

Step 2: Installation

The easiest way to install the *Jupyter Notebook App* is installing a scientific python distribution which also includes scientific python packages. The most common distribution is called **Anaconda**:

- Download Anaconda Distribution (a few 100MB), Python 3, 64 bits.
- Install it using the default settings for a *single user*.