**PYTHON LIBRARIES**

1. **Numpy**

Numpy is considered as one of the most popular machine learning library in Python.

TensorFlow and other libraries uses Numpy internally for performing multiple operations on Tensors. Array interface is the best and the most important feature of Numpy.

Features Of Numpy

1. Interactive**:** Numpy is very interactive and easy to use.
2. Mathematics**:** Makes complex mathematical implementations very simple.
3. Intuitive: Makes coding real easy and grasping the concepts is easy.
4. Lot of Interaction**:** Widely used, hence a lot of open source contribution.

**Uses of Numpy?**

This interface can be utilized for expressing images, sound waves, and other binary raw streams as an array of real numbers in N-dimensional.

For implementing this library for machine learning having knowledge of Numpy is important for full stack developers.

## ****TensorFlow****

If you are currently working on a machine learning project in Python, then you may have heard about this popular open source library known as TensorFlow.

This library was developed by Google in collaboration with Brain Team. TensorFlow is a part of almost every Google application for machine learning.

## ****Features of TensorFlow****

TensorFlow is optimized for speed, it makes use of techniques like XLA for quick linear algebra operations.

**3. Easily Trainable**

It is easily trainable on CPU as well as GPU for distributed computing.

**6. Open Source**

The best thing about this machine learning library is that it is open source so anyone can use it as long as they have internet connectivity.

## ****Uses of TensorFlow?****

All the libraries created in TensorFlow are written in C and C++. However, it has a complicated front-end for Python. Your Python code will get compiled and then executed on TensorFlow distributed execution engine built using C and C++.

## ****Scikit-learn****

It is a Python library is associated with NumPy and SciPy. It is considered as one of the best libraries for working with complex data.

There are a lot of changes being made in this library. One modification is the cross-validation feature, providing the ability to use more than one metric. Lots of training methods like logistics regression and nearest neighbors have received some little improvements.

## ****Features Of Scikit-Learn****

**1. Cross-validation:** There are various methods to check the accuracy of supervised models on unseen data.

**2. Unsupervised learning algorithms:** Again there is a large spread of algorithms in the offering – starting from clustering, factor analysis, principal component analysis to unsupervised neural networks.

**3. Feature extraction:** Useful for extracting features from images and text (e.g. Bag of words)

**4. Keras**

Keras is considered as one of the coolest machine learning libraries in Python. It provides an easier mechanism to express neural networks. Keras also provides some of the best utilities for compiling models, processing data-sets, visualization of graphs, and much more.

**Features Of Keras**

* It runs smoothly on both CPU and GPU.
* Keras supports almost all the models of a neural network – fully connected, convolutional, pooling, recurrent, embedding, etc. Furthermore, these models can be combined to build more complex models.
* Keras, being modular in nature,  is incredibly expressive, flexible, and apt for innovative research.
* Keras is a completely Python-based framework, which makes it easy to debug and explore.

1. **Pandas**

**Pandas** is a Python package that provides fast, flexible, and expressive data structures designed to make working with structured (tabular, multidimensional, potentially heterogeneous) and time series data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, **real world** data analysis in Python. Additionally, it has the broader goal of becoming **the most powerful and flexible open source data analysis / manipulation tool available in any language.**

1. **Selenium**

[SeleniumLibrary](https://github.com/robotframework/SeleniumLibrary) is a web testing library for [Robot Framework](https://robotframework.org/) that utilizes the [Selenium](https://www.seleniumhq.org/) tool internally. The project is hosted on [GitHub](https://github.com/robotframework/SeleniumLibrary) and downloads can be found from [PyPI](https://pypi.python.org/pypi/robotframework-seleniumlibrary).

SeleniumLibrary works with Selenium 3 and 4. It supports Python 3.6 or newer. In addition to the normal [Python](https://python.org/) interpreter, it works also with [PyPy](https://pypy.org/).

SeleniumLibrary is based on the [old SeleniumLibrary](https://github.com/robotframework/OldSeleniumLibrary) that was forked to [Selenium2Library](https://github.com/robotframework/Selenium2Library) and then later renamed back to SeleniumLibrary. See the [Versions](https://robotframework.org/SeleniumLibrary/#versions) and [History](https://robotframework.org/SeleniumLibrary/#history) sections below for more information about different versions and the overall project history.

**STEPS TO INSTALL A LIBRARY**

* pip is the preferred installer program. Starting with Python 3.4, it is included by default with the Python binary installers.
* A virtual environment is a semi-isolated Python environment that allows packages to be installed for use by a particular application, rather than being installed system wide.
* venv is the standard tool for creating virtual environments, and has been part of Python since Python 3.3. Starting with Python 3.4, it defaults to installing pip into all created virtual environments.
* virtualenv is a third party alternative (and predecessor) to venv. It allows virtual environments to be used on versions of Python prior to 3.4, which either don’t provide venv at all, or aren’t able to automatically install pip into created environments.

**pip install package name**

**python -m pip install SomePackage**

**python -m pip install --upgrade SomePackage**