Python Packages for   
Data Science



**Anish Mahapatra**

Contents

[Introduction 4](#_Toc114524721)

[Python Libraries – Overview 4](#_Toc114524722)

[Why Use Python Libraries for Data Science 4](#_Toc114524723)

[Ease of learning 5](#_Toc114524724)

[Less Code 5](#_Toc114524725)

[Prebuilt Libraries 6](#_Toc114524726)

[Platform Independent 6](#_Toc114524727)

[Massive Community Support 6](#_Toc114524728)

[List of Python Libraries for Data Science – 2022 (Note: Include Features, image, pros & cons, and applications for each library) 6](#_Toc114524729)

[Python Libraries for Math 6](#_Toc114524730)

[NumPy 6](#_Toc114524731)

[SciPy 8](#_Toc114524732)

[Theano 9](#_Toc114524733)

[Python Libraries for Data Exploration and Visualization 10](#_Toc114524734)

[Pandas 10](#_Toc114524735)

[Matplotlib 11](#_Toc114524736)

[Plotly 11](#_Toc114524737)

[Seaborn 11](#_Toc114524738)

[Ggplot 11](#_Toc114524739)

[Altair 11](#_Toc114524740)

[Autoviz 12](#_Toc114524741)

[Pydot 12](#_Toc114524742)

[Python Libraries for Machine Learning 12](#_Toc114524743)

[Keras 12](#_Toc114524744)

[SciKit-Learn 12](#_Toc114524745)

[PyTorch 12](#_Toc114524746)

[Pycaret 13](#_Toc114524747)

[TensorFlow 13](#_Toc114524748)

[Requests 13](#_Toc114524749)

[Python Libraries for Data Mining and Data Scrapping 13](#_Toc114524750)

[Scrapy 13](#_Toc114524751)

[BeautifulSoup 13](#_Toc114524752)

[SQLAlchemy 14](#_Toc114524753)

[Python Libraries For Natural Language Processing 14](#_Toc114524754)

[NLTK 14](#_Toc114524755)

[SpaCy 14](#_Toc114524756)

[Gensim 14](#_Toc114524757)

[Bonus Python Libraries! 14](#_Toc114524758)

[OpenCV 14](#_Toc114524759)

[Mahotas 15](#_Toc114524760)

[SimpleITK 15](#_Toc114524761)

[Pillow 15](#_Toc114524762)

[Selenium 15](#_Toc114524763)

[PyTest 15](#_Toc114524764)

[PyUnit 16](#_Toc114524765)

[Conclusion 16](#_Toc114524766)

[Frequently Asked Questions (FAQs) 16](#_Toc114524767)

[Which library is most used in Python? 16](#_Toc114524768)

[Which software is best for Python for data science? 16](#_Toc114524769)

[Which Python library should I learn first? 16](#_Toc114524770)

[Which Python library is used for machine learning? 16](#_Toc114524771)

[How many libraries are in Python? 16](#_Toc114524772)

[Author 16](#_Toc114524773)

# Introduction

You might have seen different statistics on Python as being one of the best languages to learn. We are going to differ in our opinion here. Python is **the** **best** language to learn. The reason is that python is closer to a human-interpretable language than machine-level languages like C++ or Java. It is an intuitive language that can be used across a wide range of applications.

Now, what does the term packages or library even mean? Python is a plus and play language. The idea is that if you are looking to implement a simple or even a complex logic, it is likely that someone has already done it before. This logic is then put in a form that makes it reusable – this is known as a package or a library (the terms are used interchangably). So, why this blog?

With Python, ignorance is not bliss. Ignorance would mean you spend more time implementing a logic, even if it is readily available. Through this blog, we would like for you to get a brief overview on the range of packages that are available across a variety of applications. In the next section, let’s go over the fundamentals of python packages and how we can leverage them to maser Python!

# Python Libraries – Overview

The term “library” us used to collectively describe a reusable chunk of code. A python library consists of code that we can reuse while writing code for a given application. However, just to go a little bit in detail, a collection of modules is called a package and a collection of packages is called a module. Now, a fundamental question come to mind, when people are writing all this code, why would they build libraries for eveyone to use?

This is one of the reasons Python has grown to be one of the most widely used languages in the world. Besides it’s ease of use and wide applications, there is an extremely supportive community around python with millions of possible solutions for any issues you face. Python can be used for applications such as back-end, front-end, middle ware, data science, machine learning, artificial intelligence, deep learning and even something as simple as mathematics!

In the next section, let’s understand why we should leverage the libraries that are available in python.

## Why Use Python Libraries for Data Science

Let’s take a simple, fundamental example of a function and extrapolate the use of python libraries from there. Let’s say we are trying to add two numbers and we need to use this is ten places in out code.

**Method A**

a = 1, b = 6

We will mention c = a + b in ten places in our code.

**Method B**

We can define a function,

*Def calculation\_function(a, b):   
 c = a + b   
 return c*

Now, we will mention this function is ten places in our code, instead of the code directly like in Method A.

**Use-Case**

Now, let’ say that the logic is changed by the business and we need to make it multiplication (\*), instead of addition (+).

In Method A, you will have to go to ten places and change the code manually. This is error-prone and inefficient.

In Method B, you will have to change one character in the function. This will apply in 10 places in a consistent and efficient manner.

Now, let’s scale this up to thousands, maybe even millions of lines of code. Everytime you are trying to implement a new logic, would you rather re-write so many lines of code that is error prone or would you rather use near-perfect, well-documented, versioned code that is compliant to world-class global coding standards? Unless you are doing something extraordinarly unique, the best route for 99% of people is to use packages in python.

### Ease of learning

Python code is easier than most languages to learn. A major factor is how well python is documented. It is completely free to download, install and work with. In under 30 minutes, it is possible to download Python and even write your first Python program. There are multiple

### Less Code

The more we use packages, the less time we need to write and maintain code. Great coders know how to use Python packages efficiently, so that more of their focus can be on the logic and thinking, rather than writing code. This is something important to know about geat programmers, thep spend more time thinking and optimizing, rather than just writing simple code.

### Prebuilt Libraries

If you can think it, it’s most likely built. The logic exists in the form of Python packages, we need to apply it to our context of what we are trying to achieve. These pre-builtlibraries can be used to make our lives much easier. Not just that, even for complex applications such as image recognition, deepl learning etc., years and years of research and effort by many peeople has been condensed into simple functions that we can use in a jiffy.

### Platform Independent

Whether you are a fan of one IDE (Integratd Development Environment) or the other, packages have no barriers and can be installed on any platform. This is one of the biggest advatages of using Python – you can use it almost any technology stack you have and Python fits in. Whether you work on local machines, servers or the cloud, Python is one of the most versatile options available.

### Massive Community Support

Community support is tremendous for Python with multiple forums to help you achieve any sort of task you are looking to accomplish. There are consistent contributions to Python, with new versions being released once every couple of years. There is also tremendous support for packages and libraries in Python, where you can raise bugs on the open-source forums and developers work to resolve them in upcoming versions.

# List of Python Libraries for Data Science – 2022 (Note: Include Features, image, pros & cons, and applications for each library)

Now, we will got through different categories for packages in Python ranging from Mathematics, data exploration and visualization, machine learning, data mining & data scraping, natural language processing and if you stick around till the end, we will also have bonus Python packages.

Now remember, through this package exploration, our aim is to explore python packages that can help you in the field of Data Science and Data Analytics. And, data science starts with one main thing – math! Let’s dive into the Python libraries for mathematics.

## Python Libraries for Math

In this section, we will go over the python packages we use for mathematics.

### NumPy

Just like how we see the world in terms of visuals, smell, taste and touch, machines see the world in terms of multi-dimensional arrays. As human beings, can see and feel just 3 dimensions (X-Axis, Y-Axis and Z-Axis). Machines can process and comprehend multiple dimensions, and this is represented by multidimensional arrays.

* **Features**  
    
  NumPy is an abbreviation of numerical Python and is a package that is used to work with multi-dimentional arrays. It is a fundamental package for scientific computing with Python. It is one the #1 packages used by almost everyone in the Data Science community.   
  + NumPy has functions in the domain of matrices, fourier transformation and of course, linear algebra
  + NumPy is 50 times faster than traditional Python lists! This is because NumPy stores all of it’s arrays in one continuous memory, plus it is optimized to work with the newer CPU architectures
  + NumPy is primarily written in C, C++ to enable super fast computation, as C & C++ is a machine-level language.
* **Pros**
  + **Highly optimized**: NumPy is a highly optimized package to perform scientific computation by working with numeric arrays, which makes it a fantstic tool for data scientists
  + **Efficient for use in popular packages**; NumPy arrays are used as the input for many popular packages such as scikit-learn and tensorflow
  + **The use of ndarray object**: The array object, ndarray provides a lot of supporting functions that make ndarray very efficient to use such as elementwise addition and multiplication, the computation of Kronecker product etc., which is not supported by Python lists
* **Cons**
  + **The use of NaN**: NumPy support the use of Nan, which stands for “Not a Number”, which is supported by NumPy, but not by as many packages. This makes it difficult to interpret and work with for the user.
  + **Requires a continuous allocation of memory**: When there is continuous memory allocated contiguously, the allocation and de-allocation of memory via insertion and deletion of memory becomes costly as it requires shifting.
* **Applications**
  + NumPy is leveraged to maintain minimal memory
  + It is used as an alternative to arrays and list in Python, while working well for multidimensional arrays
  + NumPy is used in cases where there is a requirement for faster runtime behaviour

### SciPy

SciPy is an open source package used for scientific and technical computing. It has modules for integration, optimization, interpolation, linear algebra, eigenvalue, statistics, multidimensional image processing etc. Fun fact – SciPy uses NumPy underneath.

Essentially, SciPy has utility functions for signal processing, stats and optimization.

* **Features**
  + Used in scientific computing and mathematics
  + SciPy comes under the umbrella the NumPy stack, which includes packages such as matplotlib and pandas.
  + SciPy has a full set of functions for linear algebra, while NumPy has comparatively less functions for linear algebra
  + SciPy has features in the domain of:
    - Integration
    - Optimization
    - Interpolation
    - Fourier Transformation
    - Signal Processing
    - Linear Algebra
    - Eigenvalues
    - Multi-dimensional Image processing
* **Pros**
  + SciPy has classes for efficient visualization and data manipulation
  + There is better cross-functionlity with other Python libraries
  + SciPy has the option for parallel programming for certain database and web routines
  + SciPy is quick and simple to pick up
* **Cons**
  + **The use of NaN**: SciPy support the use of Nan, which stands for “Not a Number”, which is supported by NumPy, but not by as many packages. This makes it difficult to interpret and work with for the user.
  + **Can be complex for someone with no mathematics background**: SciPy is meant to be a tool that can aid scientific and mathematical exploration. However, if you do not have fundamental knowledge of what you are lloking to do, it may not be the best tool.
* **Applications**
  + **Mathematics!** SciPy is used to perofrm tasks for research and scientific computation related to mathematical functions such as linear algebra, calculus, solving differential equations and signal processing.

### Theano

Theano is a python package built on top of NumPy, to manipulate and evaluate mathematical expressions, specifically matrix-valued ones.

* **Features**
  + **Integration with NumPy:** NumPy’s ndarray objects are used by the Teano library as well
  + **Can calculate derivatives:** Theano’s classs of libraries helps it to compute derivatives for one or more functions
  + **Dynamically generate C code:** Theano can dynamically generate code in programming language C to be able to evaluate expressions faster
* **Pros**
  + **Efficient GPU use**: Theano can perform operations that are data-intensive upto 140 times fastr than on a CPU, by leveraging a GPU
  + **Reliable and fast**: Theano has been know tobe stable and efficient while calcuating expressions for large values of x
  + **Self-tests:** Theano has tools to enable self-verification and unit testing, which can help catch potential problems early on in the analysis lifecycle.
* **Cons**
  + **Newer, better versions now:** Theano is considered to be the Godfather of machine learning libraries, specifically in the deep learning arena.
  + **Development stopped:** The development of Theano stopped in late 2017. In fact, Google created Tensorflow to replace Theano
* **Applications**
  + **Computer Vision:** Theano is used in computer vision such as recognizing handwriting and sparse coding
  + **Deep Learning:** Considered as the Godfather of Python packages, Theano was one of the first packages to leverage GPU optimization

## Python Libraries for Data Exploration and Visualization

### Pandas

Arguably the most used package by Data Scientists all over the world. Pandas is a software library to work with data structures and provides functions to enable data manipulation and analysis.

* **Features**
  + Pandas is able to work with a large selection of IO tools such as csv, json, SQL, BigQuery, Excel files
  + It has methods to perform functions such as object creation, viewing data, selection of data, analyzing missing data, operations such as merge, grouping, reshaping, time series, categorical values and plotting
  + Pandas has two main objects that it works with: Pandas Series and Dataframes
* **Pros**
  + **Simple representation of data:** Python has the ablity to take multiple types of data and condense the information into a simple dataframe. This facilitates us to visualize and understand the data more efficiently.
  + **Powerful features:** Any command that is needed to manipulate data can be found within the Pandas library. From filtering to grouping to segmenting, Pandas can do it all!
  + **Handles large datasets:** One of the main reason pandas was built was to handle large dataframes efficiently
* **Cons**
  + **Steep Learning Curve**: Pandas has a steep learning curve, and users that are starting out with Pandas might take some time to get accustomed to the way that the Pandas library works
  + **Imperfect Documentation:** Documentation is not the strong suite of pandas. This can perhaps be due to the sheer amount of capability of Pandas. However, if you know the application you are looking for, there are multiple use-cases to refer to.
  + **Incompatibility with 3D matrices:** One of the biggest drawbacks is Pandas’ poor compatibility in handling 3D matrices. For applications that need to process multi-dimensional arrays, it is preferred to use packages such as NumPy.
* **Applications**
  + **Recommendation Systems:** Websites like Netflix and Spotify leverage Pandas in the background for efficient processing of large volums of data
  + **Advertising:** Personalization via advertising has taken a huge leap, with software conglomerates streamlining the process of lead generation. Pandas helps a lot of smaller companies stremline their efforts
  + **Natural Language Processing:** With the help of packages such as Pandas and Scikit Learn, it has become simpler to create NLP models that can help with a plethora of applications.

### Matplotlib

Matplotlib is a Python library that aids in visualizing and plotting data to make static, animated and interactive visualizations.

* **Features**
  + Enables a wide variety of visualizations such as line plots, subplots, images, histograms, paths, bar charts, pie charts, tables, scatter plots, filled curves, log plots, date handling and stream plots
  + Can be embeded in various IDEs as well as Jupyter Lab, and Graphial User Interfaces
  + Images and visualizations can be exported to multiple file formats
* **Pros**
  + Based on NumPy, matplotlib is fairly simple for beginners to start of with
  + Intuitive for folks who have worked with graph plotting tools such as Matlab
  + High level of customization through code
* **Cons**
  + Not all visualizations from Matplotlib are interactive
  + It is difficult to adjust the visuals from Matplotlib to look great as it is a low-level interface
  + Plotting non-basic plots in matplotlib can get complex, as it can get code-heavy
* **Applications**
  + Used to make a lot of preliminary plots for large datasets, matplotlib is helpful in visualizing data
  + Given that it uses NumPy in the backend, matplotlib is used extensively with multiple third part extensions to get the fastest results

### Plotly

* Features
* Pros
* Cons
* Applications

### Seaborn

We discussed that matplotlib has a low-level interface. Seaborn is built on top of matplotlib with a high level interface to provide informative statistical graphs and to draw attractive visualizations.

* Features
  + Has plots such as relational plots, categorical plots, distrbution plots, regression plots, matrix plots, multi-plot grids
  + There are themes to styple matplotlib visualizations
  + Seaborn is able to plot linear regression models, statistical time series and works well with NumPy as well as Pandas data structures
  + It is also fast with visualizing univariate and bivariate data
* Pros
  + Seaborn is simply faster as a visualization tool – we can pass the entire ata and seaborn does a lot of the work
  + Seaborn has an interactive and informative represenatation which lets us visualize the data in a quick fashion
* Cons
  + Visualizations are not exactly interactive
  + We are limited to the styles that seaborn has in terms of customization
* Applications
  + Seaborn is used to visualize the data in an aesthetically pleasing fashion and it is used in multiple IDEs

### Ggplot

Ggplot stands for the grammar of graphics. Ggplot is a package that was built for R, it can be used in Python using the package plotnine

* Features
  + Follows a format of data, x, y and then the rest of the aesthetics
  + Can be used to create complex plots from data present in a datframe
  + It can provide a programmatic interface to work on the visualizations, the variables to represent, how to display them and their corresponding visual properties
  + Has components such as statistical transformations, scales, facets, coordinates sytems and themes
* Pros
  + The consistent underlying theme of the grammar of graphics means that you can do more visualization with lesser code
  + The plots have a high level of abstraction ad are flexible
  + The refinement has lead to a mature and complete graphics system
* Cons
  + Ggplot of slower as compares to more fundamental graphics solutions
  + Even through the visuals that ggplot has looks nicer than the other libraries, it is difficult to change the default colours
  + Gglot might require modifications to the structure of the data for certain plots
* Applications
  + A great package to use to make quick visuals, irrespective of how layered the base data is

### Altair

Altair is a declarative statistical visualization package that s based on Vega (which is a visualization grammar.

* Features
  + Can provide aesthetic and effective visualization with a minimal amount of code
* Pros
  + Base code remains the same and the user needs to only change the “mark” attribute to get various plots
  + The code is short and simple as compared to other libraries. There is a higher focus on the relationship between the data columns than the plot details
  + It is easier to implement interactivity and faceting
* Cons
  + There is a limted amount of customization possible
  + Plotting complex machine learning models becomes difficult
  + There is no 3D visualization with Altair library for Python
* Applications
  + Altair is used to automatically visualize in a nmber of ways graphs for dataframes that preferably have less than 5,000 rows ([Source](https://altair-viz.github.io/getting_started/overview.html))

### Autoviz

* Features
* Pros
* Cons
* Applications

### Pydot

* Features
* Pros
* Cons
* Applications

## Python Libraries for Machine Learning

### Keras

* Features
* Pros
* Cons
* Applications

### SciKit-Learn

* Features
* Pros
* Cons
* Applications

### PyTorch

* Features
* Pros
* Cons
* Applications

### Pycaret

* Features
* Pros
* Cons
* Applications

### TensorFlow

* Features
* Pros
* Cons
* Applications

### Requests

* Features
* Pros
* Cons
* Applications

## Python Libraries for Data Mining and Data Scrapping

### Scrapy

* Features
* Pros
* Cons
* Applications

### BeautifulSoup

* Features
* Pros
* Cons
* Applications

### SQLAlchemy

* Features
* Pros
* Cons
* Applications

## Python Libraries For Natural Language Processing

### NLTK

* Features
* Pros
* Cons
* Applications

### SpaCy

* Features
* Pros
* Cons
* Applications

### Gensim

* Features
* Pros
* Cons
* Applications

## Bonus Python Libraries!

### OpenCV

* Features
* Pros
* Cons
* Applications

### Mahotas

* Features
* Pros
* Cons
* Applications

### SimpleITK

* Features
* Pros
* Cons
* Applications

### Pillow

* Features
* Pros
* Cons
* Applications

### Selenium

* Features
* Pros
* Cons
* Applications

### PyTest

* Features
* Pros
* Cons
* Applications

### PyUnit

* Features
* Pros
* Cons
* Applications

# Conclusion

# Frequently Asked Questions (FAQs)

## Which library is most used in Python?

## Which software is best for Python for data science?

## Which Python library should I learn first?

## Which Python library is used for machine learning?

## How many libraries are in Python?

# Author

**Anish Mahapatra**   
*Senior Data Scientist*   
LinkedIn: [https://www.linkedin.com/in/anishmahapatra](about:blank) ([LinkedIn](about:blank))   
Email: anishmahapatra01@gmail.com