Top 10 Python Libraries for Data Science

Python has been a charmer for data scientists for a while now. The more I interact with resources, literature, courses, training, and people in Data Science, proficient knowledge of Python emerges as a good asset to have. Having said that, when I started flourishing my Python skills, I had a list of Python libraries I had to know about. A few moments later…

Python has been a charmer for data scientists for a while now. 😀

People in Data Science definitely know about the Python libraries that can be used in Data Science but when asked in an interview to name them or state its function, we often fumble up or probably not remember more than 5 libraries (it happened with me :/)

Here today, I have curated a list of 10 Python libraries that helps in Data Science and its periphery, when to use them, what are its significant features and the advantages.

In this story, I have briefly outlined 10 most useful Python libraries for data scientists and engineers, based on my recent experience and explorations. Read the full story to know about 4 bonus libraries!

1. Pandas

- *Pandas* is an open-source Python package that provides high-performance, easy-to-use data structures and data analysis tools for the labeled data in Python programming language. Pandas stand for *Python Data Analysis Library*. Who ever knew that?
- When to use? Pandas is a perfect tool for data wrangling or munging. It is designed for quick and easy data manipulation, reading, aggregation, and visualization.
- Pandas take data in a CSV or TSV file or a SQL database and create a Python object with rows and columns called a data frame. The data frame is very similar to a table in statistical software, say Excel or SPSS.
- What can you do with Pandas?
- Indexing, manipulating, renaming, sorting, merging data frame
- Update, Add, Delete columns from a data frame
- Impute missing files, handle missing data or NANs
- Plot data with histogram or box plot
- This makes *Pandas* a foundation library in learning Python for Data Science.

2. NumPy

- One of the most fundamental packages in Python, *NumPy* is a general-purpose array-processing package. It provides high-performance multidimensional array objects and tools to work with the arrays. NumPy is an efficient container of generic multi-dimensional data.

- NumPy's main object is the homogeneous multidimensional array. It is a table of elements or numbers of the same datatype, indexed by a tuple of positive integers. In NumPy, dimensions are called *axes* and the number of axes is called *rank*. NumPy's array class is called *ndarray* aka *array*.
- When to use? NumPy is used to process arrays that store values of the same datatype. NumPy facilitates math operations on arrays and their vectorization. This significantly enhances performance and speeds up the execution time correspondingly.
- What can you do with NumPy?
- Basic array operations: add, multiply, slice, flatten, reshape, index arrays
- Advanced array operations: stack arrays, split into sections, broadcast arrays
- Work with DateTime or Linear Algebra
- Basic Slicing and Advanced Indexing in NumPy Python

3. SciPy

- The SciPy library is one of the core packages that make up the SciPy stack. Now, there is a difference between SciPy Stack and SciPy, the library. *SciPy* builds on the NumPy array object and is part of the stack which includes tools like Matplotlib, Pandas, and SymPy with additional tools,
- SciPy library contains modules for efficient mathematical routines as linear algebra, interpolation, optimization, integration, and statistics. The main functionality of the SciPy library is built upon NumPy and its arrays. SciPy makes significant use of NumPy.
- When to use? SciPy uses arrays as its basic data structure. It has various modules to perform common scientific programming tasks as linear algebra, integration, calculus, ordinary differential equations, and signal processing.

4. Matplotlib

- This is undoubtedly my favorite and a quintessential Python library. You can create stories with the data visualized with Matplotlib. Another library from the SciPy Stack, Matplotlib plots 2D figures.
- When to use? Matplotlib is the plotting library for Python that provides an object-oriented API for embedding plots into applications. It is a close resemblance to MATLAB embedded in Python programming language.
- What can you do with Matplotlib?
- Histogram, bar plots, scatter plots, area plot to pie plot, Matplotlib can depict a wide range of visualizations. With a bit of effort and tint of visualization capabilities, with Matplotlib, you can create just any visualizations:
- Line plots
- Scatter plots
- Area plots
- Bar charts and Histograms
- Pie charts
- Stem plots
- Contour plots

- Quiver plots
- Spectrograms
- Matplotlib also facilitates labels, grids, legends, and some more formatting entities with Matplotlib. Basically, everything that can be drawn!

5. Seaborn

- So when you read the official documentation on Seaborn, it is defined as the data visualization library based on Matplotlib that provides a high-level interface for drawing attractive and informative statistical graphics. Putting it simply, seaborn is an extension of Matplotlib with advanced features.
- So, what is the difference between Matplotlib and Seaborn? Matplotlib is used for basic plotting; bars, pies, lines, scatter plots and stuff whereas, seaborn provides a variety of visualization patterns with less complex and fewer syntax.
- What can you do with Seaborn?
- Determine relationships between multiple variables (correlation)
- Observe categorical variables for aggregate statistics
- Analyze uni-variate or bi-variate distributions and compare them between different data subsets
- Plot linear regression models for dependent variables
- Provide high-level abstractions, multi-plot grids
- Seaborn is a great second-hand for R visualization libraries like *corrplot* and *ggplot*.

6. Scikit Learn

- Introduced to the world as a Google Summer of Code project, Scikit Learn is a robust machine learning library for Python. It features ML algorithms like SVMs, random forests, k-means clustering, spectral clustering, mean shift, cross-validation and more... Even NumPy, SciPy and related scientific operations are supported by Scikit Learn with Scikit Learn being a part of the SciPy Stack.
- When to use? Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. Supervised learning models like Naive Bayes to grouping unlabeled data such as KMeans, Scikit learn would be your go-to.
- What can you do with Scikit Learn?
- Classification: Spam detection, image recognition
- Clustering: Drug response, Stock price
- Regression: Customer segmentation, Grouping experiment outcomes
- Dimensionality reduction: Visualization, Increased efficiency
- Model selection: Improved accuracy via parameter tuning
- Pre-processing: Preparing input data as a text for processing with machine learning algorithms.
- Scikit Learn focuses on modeling data; not manipulating data. We have NumPy and Pandas for summarizing and manipulation.

7. TensorFlow

- Back in 2017, I received a TensorFlow USB as an appreciation for being an amazing speaker at a Google WTM event, haha. The USB was loaded with official documentation of TensorFlow. With no clue at that point of what TensorFlow was, I Googled it.
- TensorFlow is an AI library that helps developers to create large-scale neural networks with many layers using data flow graphs. TensorFlow also facilitates the building of Deep Learning models, push the state-of-the-art in ML/AI and allow easy deploy of ML-powered applications.
- One of the most developed websites amongst all libraries is of TensorFlow. Giants like Google, Coca-Cola, Airbnb, Twitter, Intel, DeepMind, everyone uses TensorFlow!
- When to Use? TensorFlow is quite efficient when it comes to classification, perception, understanding, discovering, predicting, and creating data.
- What to do with TensorFlow?
- Voice/Sound Recognition — IoT, Automotive, Security, UX/UI, Telecom
- Sentiment Analysis — Mostly for CRM or CX
- Text-Based Apps — Threat Detection, Google Translate, Gmail smart reply
- Face Recognition — Facebook's Deep Face, Photo tagging, Smart Unlock
- Time Series — Recommendation from Amazon, Google, and Netflix
- Video Detection — Motion Detection, Real-Time Threat Detection in Gaming, Security, Airports

8. Keras

Keras is TensorFlow's high-level API for building and training Deep Neural Network code. It is an open-source neural network library in Python. With Keras, statistical modeling, working with images and text is a lot easier with simplified coding for deep learning.

What is the difference between Keras and TensorFlow after all?

Keras is a neural network Python library while TensorFlow is an open-source library for various machine learning tasks. TensorFlow provides both high-level and low-level APIs while Keras provides only high-level APIs. Keras is *built for Python* which makes it way more user-friendly, modular and composable than TensorFlow.

What can you do with Keras?

Determine percentage accuracy

Compute loss function

Create custom function layers

Built-in data and image processing

Write functions with repeating code blocks: 20, 50, 100 layers deep

9. Statsmodels

When I first learned R, conducting statistical tests, and statistical data exploration seemed the easiest in R and avoided Python for statistical analysis until I explored Statsmodels or Python.

When to use? Statsmodels is the ultimate Python package that provides easy computations for descriptive statistics and estimation and inference for statistical models.

What to do with Statsmodels?

Linear Regression

Correlation

Ordinary Least Squares (OLS) for the economist in you!

Survival analysis

Generalized linear models and Bayesian model

Uni-variate & bi-variate analysis, Hypothesis Testing (basically, what R can do!)

10. Plotly

Plotly is a quintessential graph plotting library for Python. Users can import, copy, paste, or stream data that is to be analyzed and visualized. Plotly offers a sandboxed Python(Something where you can run a Python that is limited in what it can do) Now I've had a hard time understanding what sandboxing is but I know for a fact that Plotly makes it easy!?

When to use? You can use Plotly if you want to create and display figures, update figures, hover over text for details. Plotly also has an additional feature of sending data to cloud servers. That's interesting!

What can you do with Plotly?

The Plotly graph library has a wide range of graphs that you can plot:

Basic Charts: Line, Pie, Scatter, Bubble, Dot, Gantt, Sunburst, Treemap, Sankey, Filled Area Charts

Statistical and Seaborn Styles: Error, Box, Histograms, Facet and Trellis Plots, Tree plots, Violin Plots, Trend Lines

Scientific charts: Contour, Ternary, Log, Quiver, Carpet, Radar, Heat maps Windrose and Polar Plots

- Financial Charts
- Maps
- Subplots

- Transforms
- Jupyter Widgets Interaction

Told you, Plotly is the *quintessential* plots library. Think of visualization and plotly can do it!

Now is the time, when we have explored an interview-notes worth guide of top 10 Python libraries for data science, we look for our four bonus libraries!

1. Spacy

SpaCy is an open-source library used for advanced NLP for Python and Cython (A Python programming language to give C-like feel and performance with Python code, plus a C-inspired syntax)

2. Bokeh

Bokeh is a Python library I would like to term as for interactive data visualization. With tools like Tableau, QlikView or PowerBI why would we need Bokeh? First, Bokeh allows building complex statistical plots with simple commands real quick. It supports HTML, notebook or a server output. Second, it is possible to integrate Bokeh visualization to Flask and Django apps, or visualizations written in other libraries like matplotlib, seaborn, ggplot.

3. Gensim

Gensim is something I believe is so different from what we've seen so far. It automatically extracts semantic topics from documents with high efficiency and effortlessly. The Gensim algorithms are unsupervised, which hints that no human input is necessary —just plain text documents and the extraction is then performed.

4. NLTK

NLTK (Natural Language Toolkit) mainly works with human language more than computer language to apply natural language processing (NLP). It contains text processing libraries with which you can perform tokenization, parsing, classification, stemming, tagging and semantic reasoning of data. It may sound repetitive of what this library can do but every lib in Python was written to address some efficiency.