# All Python Libraries You Need For Machine Learning And Data Science

Patrick Loeber

Published on Jun 27, 2021

5 min read

In this article I show you all Python libraries I use for Machine Learning, Deep Learning, and Data Science. The list is not exhaustive, there are of course many other great libraries out there, but it covers 95% of the frameworks I use.

I've split the frameworks roughly into these categories:

- The base libraries you almost always need for Machine Learning and Data Science
- Optional helpful additions for ML
- Deep Learning Frameworks
- Frameworks for Computer Vision
- Frameworks for NLP
- Frameworks for deployment in the web.

Video URL: https://youtu.be/vVg7WrelMeA

## The Base

- numpy
- pandas
- matplotlib

- scikit-learn

The number 1 package you have to know and which is also the base for many other packages is of course numpy. It's needed for matrix operations and linear algebra and is much faster than using Python lists.

Then we have pandas - The Python Data Analysis Library. It's a powerful tool to load the data into so called data frames, which is basically a table that you can then easily analyze, modify, and also visualize.

Pandas also has functions to plot the data, but this is just referencing the matplotlib API. Matplotlib is the base package for plotting and visualizing data in Python. With matplotlib you can create all kinds of plots like line plots, bar plots, pie plots and many many more. This is essential to analyze and understand your data.

The base Machine Learning library in Python is scikit-learn. It offers almost all the „classical" Machine Learning models you need, so it offers models for Regression, Classification, Clustering, and Dimensionality Reduction. Additionally, there are algorithms to preprocess data, e.g., for feature extraction or feature normalization.

These are the top 4 Python libraries, which I also call the **Machine Learning Tech Stack**. If you are serious about ML and Data Science you should definitely get experience with those frameworks.

# Optional For ML

- seaborn
- XGBoost
- imbalanced-learn

Seaborn is another data visualization library based on matplotlib. It provides additional visualization methods and seaborn plots often look a little bit more beautiful than plain matplotlib plots.

Then we have XGBoost. XGBoost is an optimized distributed gradient boosting library that implements highly efficient parallel tree boosting algorithms such as Gradient boosting decision trees (GBDT). They typically perform really well and can be often seen on Kaggle where they dominate other algorithms. Definitely give this a shot if you need an efficient and powerful model.

Another great library is imbalanced-learn. This is a super helpful library when you have to deal with imbalanced data, e.g., if you have a lot of samples from the negative class but not from the positive class. You should address this problem in your preprocessing steps and imbalanced-learn offers a lot of different algorithms to do this, for example different under- and oversampling methods.

# Deep Learning

- TensorFlow
- PyTorch

Now let's have a look at what you need for Deep Learning and when working with Neural Networks. Here you should choose one of the two popular Deep Learning frameworks TensorFlow or PyTorch. I'm not going into the argument which one is better now, both are awesome and can get the job done. So just pick one in the beginning and try to get some experience with the framework.

They will build the base for Computer Vision and NLP tasks. But for those 2 specific fields let's have a look at some more helpful libraries.

# Computer Vision

- OpenCV
- Pillow

For Computer Vision the number 1 most important library is OpenCV. It offers powerful algorithms for real time image and video processing. Some techniques could be used for preprocessing or labelling the data and then combine it with TensorFlow or PyTorch, but it also has algorithms for full pipelines, e.g., for object detection, object segmentation, and face recognition algorithms .

Next to this you can use Pillow, the Python Imaging Library. This also offers image processing algorithms and is a little bit more light weight. I mainly use it when I have to load and convert images, and also for some image displaying and drawing tasks.

# NLP

- HuggingFace
- NLTK
- spaCy

For NLP I mainly use 3 libraries. The first one is the HuggingFace Transformers library, which offers many pretrained State-of-the-art Natural Language Processing models and algorithms that can be combined directly with both PyTorch and TensorFlow. It's one of the most popular NLP frameworks in Python right now.

Then there is NLTK, the Natural Language Toolkit. This is another essential library when working with language data. It offers algorithms for text classification, tokenization, stemming, tagging, and many more text processing techniques.

The last NLP library I use is spaCy. spaCy also contains powerful NLP algorithms and is designed to build production ready systems real fast. They provide a great free course on their website that you can check out if you want to get started with it.

# Web

- Flask
- FastAPI
- Streamlit

For deployment in the web I like to be able to build APIs real fast. My two favorite frameworks for this are Flask and FastAPI. You might be asking why not Django. Django is awesome and I would definitely use this for full blown web apps, but if you just need to quickly build an API with a few endpoints then Flask or FastAPI are the better choice.

Flask is more established and very beginner friendly. So if you don't have much experience with web apps than use this. FastAPI on the other hand is pretty new, but it's getting more and more popular. It's one of the fastest Python web frameworks out there and has a lot of great features and design choices that allow for rapid production ready API development. Definitely give this a try if you are no longer a complete beginner.

The last web framework I sometimes use is Streamlit. Streamlit makes it super simple to build beautiful web apps without having to worry about implementing the UI. You get beautiful widgets just out of the box and can add for example buttons, sliders, and plots with just one line of code. I use this a lot when I quickly need an app with a nice UI to demonstrate my machine learning models.

# End

Alright so that's it. This list covers 95% of the libraries I use. I hope you found this helpful. And as I said, I know there are many other great frameworks out there, so let me know which ones you can recommend.

Reference link: https://patloeber.hashnode.dev/python-libraries-for-machine-learning