Top 128 Utile Python Libraries for Aspiring Data Scientists to Try

Alexzap · Following

20 min read · 2 days ago

Information is the oil of the 21st century, and analytics is the combustion engine — Peter Sondergaard (Senior VP and the Global Head of Research at Gartner Inc).

Photo by imgix on Unsplash

- More and more businesses recognize the value of strong Data Science (DS) skills. Whether you're interested in becoming a data science professional or have your sights set on a management role, this article allows you to learn the latest Python DS advances for effectively analyzing data from multiple domains, communicating with stakeholders, and positively impacting business decisions.

Why Python:

- Easy to Use, Learn and Share
- Low-Code/No-Code Automation
- Dynamic Memory Allocation
- Open-Source Projects
- Extensive GUI Support

- Wide Range of Libraries
- Machine Learning (ML)

PYPL Popularity of Programming Language, Feb 2024:

- The PYPL Popularity of Programming Language Index is created by analyzing how often language tutorials are searched on Google.
- Worldwide, Python is the most popular language, Python grew the most in the last 5 years (2.6%) and Java lost the most (-5.3%).
- Hence, Python is known to be one of the most important programming languages to learn when becoming a data scientist.
- This guide will help you build confidence in the Python learning journey, develop a specific application that helps you stand out in the job hunt, and have fun along the way.
- Throughout the study, we'll explore the world of Python DS with 22 highly engaging user-friendly projects.
- The post concludes by encouraging readers to check out the resources with project links for inspiration and finding project ideas.

Setting Up Your IDE

- Before jumping into DS, you need to set up the required software and tools and learn how to use them. This section will teach you how to install and use the Jupyter notebook within the Anaconda IDE for building a data science ecosystem.
- Download Anaconda installer for your operating system from: https://www.anaconda.com/downloads
- Once the download is complete, double-click the package to start installing Anaconda using the default settings.

- Click on Continue on the Introduction, Read Me, and License screens. Click on Agree to continue the installation, once the prompt below appears.
- On the Destination Select screen, select “Install for me only” and use the default installation PATH.
- Click on Continue to install Anaconda without the PyCharm IDE.
- After the installation completes, click on Close on Summary.
- Double-click on the Anaconda Navigator icon.
- Go to the Home page on Anaconda IDE and install Jupyter Notebook from an application. A Jupyter Notebook interface will appear in your default browser.
- With Jupyter Notebook open in your browser, you may have noticed that the URL for the dashboard is something like `https://localhost:8888/tree`. Localhost is not a website, but indicates that the content is being served from your *local* machine: your own computer.
- Browse to the folder in which you would like to create your first notebook, click the “New” drop-down button in the top-right and select “Python 3”.
- Your first Jupyter Notebook will open in new tab — each notebook uses its own tab because you can open multiple notebooks simultaneously.
- If you switch back to the dashboard, you will see the new file `Untitled.ipynb`. Each `.ipynb` file is one notebook, so each time you create a new notebook, a new `.ipynb` file will be created.
- Cells form the body of a notebook. In the screenshot of a new notebook in the section above, that box with the green outline is an empty cell.
- When we run the cell from the menu bar, its output is displayed below and the label to its left will have changed from `In [ ]` to `In [1]`.

- From the menu bar, click *Insert* and select *Insert Cell Below* to create a new code cell underneath your first and try out the following code to see what happens.
- Behind every notebook runs a kernel. When you run a code cell, that code is executed within the kernel. Any output is returned back to the cell to be displayed. The kernel's state persists over time and between cells — it pertains to the document as a whole and not individual cells.
- Most of the time when you create a notebook, the flow will be top-to-bottom.
- If your kernel is ever stuck on a computation and you wish to stop it, you can choose the Interrupt option.

Public-Domain Datasets, Models & APIs

- Awesome Public Datasets: This is a list of topic-centric public data sources in high quality. They are collected and tidied from blogs, answers, and user responses.
- Best Free Public Datasets to Use in Python
- Kaggle: Find 415,659 Datasets
- Best Dataset Search Engines for Your Python Projects
- Datasets — UCI Machine Learning Repository
- yfinance 0.2.51 Download market data from Yahoo! Finance API
- nlp-datasets Alphabetical list of free/public domain datasets with text data for use in NLP
- Many ventures provide pre-trained models, one of which is Hugging Face launched in 2017.
- https://twelvedata.com/ Access stocks, Forex and other financial assets from anywhere at any time.

I/O Data Handling

- Pandas is a fundamental data manipulation library in Python. It provides data structures like DataFrame and Series, enabling the smooth handling of structured data. Pandas provides a set of tools to facilitate data cleaning, filtering, merging, grouping, and aggregation.
- While NumPy is primarily known for its numerical computing capabilities, it also plays a significant role in data manipulation. It provides support for arrays and matrices, enabling efficient manipulation of large datasets.
- SciPy builds on NumPy and provides additional scientific computing functionalities, including statistical functions, optimization, integration, interpolation, and more.
- Polars is a DataFrame library completely written in Rust and is built to empower Python developers with a scalable and efficient framework for handling data and is considered as an alternative to the very popular pandas library. It provides a wide range of functionalities that facilitate various data manipulation and analysis tasks.

Data Processing

- Pandas is a powerful, fast, and open-source library built on NumPy. It is used for data manipulation and real-world data analysis in Python. Easy handling of missing data, Flexible reshaping and pivoting of data sets, and size mutability make pandas a great tool for performing data manipulation and handling the data efficiently.
- Polars is a promising library for data manipulation and analysis in Python, with a focus on performance and memory efficiency as compared to Pandas.
- Dask is a parallel computing library that can handle larger-than-memory datasets by partitioning them across multiple cores or machines. It

provides a Pandas-like API and can be used as a drop-in replacement for Pandas in many cases.

- Vaex is a library for working with large datasets that are too big to fit into memory. It provides lazy loading and columnar memory layout to enable fast computations on large datasets.

Code Analysis

- Pylint is a static code analysis tool that lists error which may come after execution of the python code, helps to enforce a coding standard, and look for code smells, offers simple refactoring suggestions, and other suggestions about code complexity.
- Pyflakes is a verification tool for python source code. It just doesn't verify the style at all but verifies only logistic errors. It emits very few false positives, which means that it will not display errors about missing docstrings or argument names that don't match the naming style.
- Prospector is a powerful static analysis tool for Python code. It displays information about errors, potential problems, convention violations, and complexity.

Debugging

- IceCream: Never Use Print() To Debug Your Python Code Again.
- The (conventional) module `pdb` defines an interactive source code debugger for Python programs. It supports setting (conditional) breakpoints and single stepping at the source line level, inspection of stack frames, source code listing, and evaluation of arbitrary Python code in the context of any stack frame.
- python-hunter — A flexible code tracing toolkit.

(Auto) EDA

- Automated EDA (Exploratory Data Analysis) packages can perform EDA in a few lines of Python code. In this section, we will discuss 10 Automated EDA Tools that can perform EDA and generate insights about the data:

1. Pandas-Profiling = ydata-profiling Learn how to use the ydata-profiling library in Python to generate detailed reports for datasets with many features.

2. SweetViz is an open-source Python library that generates beautiful, high-density visualizations to kickstart EDA with just two lines of code. Output is a fully self-contained HTML application.

3. AutoViz can handles large datasets by intelligently sampling data, ensuring quick and efficient visualization generation without compromising insights.

4. DataPrep lets you prepare your data using a single library with a few lines of code.

5. D-Tale: Web Client for Visualizing Pandas Objects.

7. QuickDA: Simple & Easy-to-use Python modules to perform quick EDA for any structured dataset.

8. Datatile: With one click, DataTile's crosstabs provide efficient data analytics, revealing research trends and audience insights without spending days in SPSS or Excel.

9. Lux is a Python library that facilitates fast and easy EDA by automating the visualization and data analysis process.

10. ExploriPy is designed in a way to perform AutoEDA and statistical tests. It provides an interpretation of statistical testing.

Data Visualization

- Matplotlib is a popular Python library for creating high-quality graphs, charts, and plots.
- Seaborn is a higher-level interface to Matplotlib that provides additional functionality for statistical visualization. It is built on top of Matplotlib and provides an easy-to-use API for creating complex visualizations.
- plotly.py is an interactive, open-source, and browser-based graphing library for Python.
- Altair is a Python library designed for statistical visualization. It is declarative in nature.
- Bokeh makes it simple to create common plots, but also can handle custom or specialized use-cases.
- PyGWalker: A Python Library for Exploratory Data Analysis with Visualization

Webscraping

- The requests library is used for making HTTP requests to a specific URL and returns the response.
- Scrapy is a free and open-source web scraping Python library. It's designed for large-scale web scraping.
- Beautiful Soup provides a few simple methods and Pythonic phrases for guiding, searching, and changing a parse tree:

```
!pip install beautifulsoup4
```

Automation of Excel Tasks

- With the help of libraries like Pandas, Xlwings, and OpenPyXL, Python makes it simple to:

1. I/O Excel files.
2. Automate calculations.
3. Generate reports.

Automation allows you to concentrate on important work while Python takes care of the repetitive tasks in Excel.

Time Series Analysis

- pmdarima 2.0.4 is a statistical library designed to fill the void in Python's time series analysis capabilities.
- Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects.
- Skforecast is a Python library for time series forecasting using ML models.
- Flow Forecast (FF) is an open-source deep learning for time series forecasting framework. It provides all the latest state of the art models (transformers, attention models, GRUs, ODEs) and cutting edge concepts with easy to understand interpretability metrics, cloud provider integration, and model serving capabilities.

- tsfresh automatically calculates a large number of time series characteristics, aka features.
- AutoTS is a time series package for Python designed for rapidly deploying high-accuracy forecasts at scale. In 2023, AutoTS won in the M6 forecasting competition, delivering the highest performance investment decisions across 12 months of stock market forecasting.
- Darts is a Python library for user-friendly forecasting and anomaly detection on time series.
- Automated Time Series Models in Python (AtsPy): Easily develop state of the art time series models to forecast univariate data series. This is the largest repository of automated structural and machine learning time series models.
- sktime is a unified framework for machine learning with time series.
- Greykite: A flexible, intuitive and fast forecasting and anomaly detection library.

Statistical Analysis

- statsmodels — Statistical modeling and econometrics in Python.
- Numpy and Pandas are two popular Python libraries that are often used together and provide a wide range of capabilities to support statistical modeling.
- statistics module provides functions for calculating mathematical statistics of numeric (`Real` -valued) data. It is complementary to Numpy and SciPy.
- Pingouin is a statistical package in Python that is designed for easy use and includes a variety of statistical tests and functions.

Numerical Optimization

- Optimization (`scipy.optimize`)
- The package provides several commonly used optimization algorithms. A detailed listing is available: **scipy.optimize** (can also be found by `help(scipy.optimize)`).

Digital Signal Processing (DSP)

- Signal Processing with Python
- Signal processing (`scipy.signal`)

How To apply a filter to a signal in python

```
scipy.signal.filtfilt
scipy.signal.lfilter
```

- PySDR: A Guide to SDR and DSP using Python
- ObsPy — A Python toolbox for seismology.

Unsupervised ML

- Scikit-Learn covers a wide range of ML techniques, including classification, regression, clustering, dimensionality reduction, model selection, and preprocessing:

```
!pip install scikit-learn
```

- SciKit-Learn unsupervised learning explores data by looking for structures or patterns.

- Clustering of unlabeled data (with K-Means, DBSCAN, BIRCH, etc.) can be performed with the module `sklearn.cluster`.
- sklearn.decomposition.PCA is basically a statistical procedure to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables.
- Simple Autoencoder is actually an Artificial Neural Network (ANN) that is used to decompress and compress the input data provided in an unsupervised manner. The end goal is to perfectly replicate the input with minimum loss.

```
from keras.layers import Dense,Conv2D,MaxPooling2D,UpSampling2D
from keras import Input, Model
```

- Association Rule Mining in Python is a technique used to uncover hidden relationships between variables in large datasets. It is a popular method in data mining and has a wide range of applications in various fields, such as market basket analysis, customer segmentation, and fraud detection.

```
# init setup

from pycaret.arules import *
```

- Deep Belief Networks in Python is a stack of Restricted Boltzmann Machines connected together and a feed-forward neural network:

```
from dbn.tensorflow import SupervisedDBNClassification
```

- xRBM Library implements Restricted Boltzmann Machine (RBM) and its variants in Tensorflow:

```
!pip install xRBM
```

- Hierarchical Temporal Memory is a type of neural network used for unsupervised learning along with supervised learning problems where labeled examples exist but not enough labels were generated during training time.
- Convolutional Neural Network (CNN) is a type of neural network that is used for both unsupervised and supervised learning problems. They work by taking an input image and splitting it into small square tiles called “windows.” Each window is then passed through a neuron in the first layer of the CNN which performs a convolution operation on it using a kernel matrix.
- Support Vector Machines with Scikit-learn are used for both unsupervised and supervised learning problems. They work by constructing a hyperplane in a high-dimensional space where all the training data points lie on one side of the plane and all the other data points lie on the other side.

Supervised/Deep Learning

- SciKit-Learn supervised learning is suitable for both regression and classification ML tasks. Its comprehensive set of tools, combined with its

simplicity, makes it an indispensable tool for data scientists and ML engineers.

- TensorFlow is is widely used for building, training, and deploying ML models, and has a strong focus on deep learning. TensorFlow has a comprehensive set of tools for building and training neural networks, including support for convolutional neural networks (CNNs) and long short-term memory (LSTM) networks.

```
import tensorflow as tf
```

- PyTorch is is designed for fast and flexible prototyping and has a strong focus on deep learning. PyTorch has a dynamic execution model, allowing developers to easily modify their models during training. It also has support for GPU acceleration, making it well-suited for training large and complex models.

```
import torch
```

- Theano is a numerical computation Python library made specifically for ML.
- Keras is designed for developing and evaluating neural networks within deep learning and ML models. It can run on top of Theano and TensorFlow, making it possible to start training neural networks with a little code.

Semi-Supervised Learning (SSL)

- Semi-supervised learning is a situation in which in your training data some of the samples are not labeled. The semi-supervised estimators in `sklearn.semi_supervised` are able to make use of this additional unlabeled data to better capture the shape of the underlying data distribution and generalize better to new samples. These algorithms can perform well when we have a very small amount of labeled points and a large amount of unlabeled points.
- `sklearn` has two SSL algorithms: Label Propagation and Label Spreading.

Reinforcement Learning (RL)

- RL trains an agent to make decisions by interacting with an environment.

```
import gym
```

Transfer Learning (TL)

- The intuition behind transfer learning is that if datasets in two problems contain similar data points, their feature representations are also similar and thus the weights obtained from one training can be used in solving subsequent similar problems rather than using random weights and model training from scratch.
- In multi-task learning, the pre-trained model is trained on multiple tasks simultaneously.

```
import torch
import torch.nn as nn

# Define the pre-trained model
class PreTrainedModel(nn.Module):
.....................
```

```
# Define the multi-task model
class MultiTaskModel(nn.Module):
.....................

# Load the pre-trained model
pre_trained_model = PreTrainedModel()
pre_trained_model.load_state_dict(torch.load('pre_trained_model.pt'))

# Define the multi-task model
multi_task_model = MultiTaskModel(pre_trained_model)

# Train the multi-task model on multiple tasks simultaneously
optimizer = torch.optim.Adam(multi_task_model.parameters())
```

AutoML

- Automated Machine Learning provides methods and processes to make ML available for non-Machine Learning experts, to improve efficiency of ML and to accelerate research on ML.
- H2O AutoML emphasizes ease of use and scalability. It automatically searches through possible models and preprocessing steps to find the most effective machine learning pipeline.

```
import h2o
from h2o.automl import H2OAutoML

# Start the H2O cluster (locally)
h2o.init()
```

- Auto-sklearn is an automated ML toolkit based on the popular scikit-learn library. It focuses on combining different algorithms and pre-processing methods to find the best model for a given dataset. Auto-sklearn employs Bayesian optimization, meta-learning, and ensemble methods to achieve high performance.

- MLBox is an AutoML library that offers pre-processing, optimization, and prediction capabilities. It is designed for efficiency, capable of handling large datasets and performing feature selection automatically.

```
!pip install mlbox
```

- TPOT, short for Tree-based Pipeline Optimization Tool, leverages genetic algorithms to automate the design of ML pipelines. TPOT is built on top of scikit-learn, so all of the code it generates should look familiar.
- AutoKeras is an AutoML system based on Keras. It simplifies deep learning by automating the design and tuning of neural networks. It utilizes neural architecture search (NAS) to find the best model architecture for a specific problem.
- AutoGluon automates feature engineering, model selection, and hyperparameter tuning, enabling it to handle tabular, image, and text data.
- Auto_ViML , short for Automated Variant Interpretable Machine Learning, focuses on creating interpretable ML models.
- EvalML is an AutoML library which builds, optimizes, and evaluates ML pipelines using domain-specific objective functions.
- PyCaret is an open-source, low-code ML library in Python that automates ML workflows.
- dabl tries to help make supervised ML more accessible for beginners, and reduce boiler plate for common ML tasks.
- FLAML, short for Fast and Lightweight AutoML, supports AutoML and Hyperparameter Tuning in Microsoft Fabric Data Science. FLAML's

efficiency makes it useful for scenarios where computational resources are limited.

ML Interpretation

- Scikit-Plot provides visualizations for many ML metrics related to regression, classification, and clustering. Scikit-Plot is built on top of Matplotlib.

```
!pip install scikit-plot
```

- Yellowbrick is a suite of visual analysis and diagnostic tools for ML regression and classification tasks.

```
!pip install yellowbrick
```

- interpret-text is the Microsoft Interpret Text SDK for Python. It builds on Interpret, an open source Python package for training interpretable models and helping to explain black box ML systems.

```
!pip install interpret-text
```

- InterpretML is an open-source package that incorporates state-of-the-art machine learning interpretability techniques under one roof. With this package, you can train interpretable glassbox models and explain

blackbox systems. InterpretML helps you understand your model's global behavior, or understand the reasons behind individual predictions.

```
!pip install interpret
```

- LIME (Local Interpretable Model-Agnostic Explanations for ML classifiers) is able to explain any black box classifier, with two or more classes. All they require is that the classifier implements a function that takes in raw text or a numpy array and outputs a probability for each class. Support for scikit-learn classifiers is built-in.

```
!pip install lime
```

- ELI5 is a Python package which helps to debug ML classifiers and explain their predictions.

```
!pip install eli5
```

- SHAP stands for SHapley Additive exPlanations. SHAP is a unified game theoretic approach approach to explain the output of any ML model.

```
!pip install shap
```

- TreeInterpreter is a package for interpreting scikit-learn's decision tree and random forest predictions. Allows decomposing each prediction into bias and feature contribution components as described in http://blog.datadive.net/interpreting-random-forests/.

```
!pip install treeinterpreter
```

- Diverse Counterfactual Explanations (DiCE) for ML. Exploring "what-if" scenarios is an important way to inspect a ML model. The DiCE library helps you to understand an ML model by generating "what-if" data points that lead to the desired model output.

```
!pip install dice-ml
```

- Captum is a model interpretability and understanding library for PyTorch. Captum contains general purpose implementations of integrated gradients, saliency maps, smoothgrad, vargrad and others for PyTorch models.
- Requirements: Python >= 3.9, PyTorch >= 1.10

```
conda install captum -c pytorch
```

Computer Vision (CV)

- OpenCV is a library is useful for real-time CV programs. It's able to process a variety of visual inputs from image and video data and identify objects, faces, and handwriting.

```
!pip install opencv-python

import cv2
```

- GluonCV provides implementations of the state-of-the-art deep learning models in CV.

```
!pip install gluoncv mxnet>=1.6.0 --upgrade
```

- PyTorch + TorchVision — A framework for deep learning and CV tasks.

Image Processing

- Pillow is based on the Python Imaging Library (PIL). This library provides extensive file format support, an efficient internal representation, and fairly powerful image processing capabilities. It encompasses several image processing activities, including point operations, filtering, manipulating, etc.

```
!pip install pillow
```

- scikit-image: Image processing in Python

```
conda install -c conda-forge scikit-image
```

- SciPy is mainly used for scientific computations, but it can also be used for image processing and CV by importing relevant modules of the library.
- GraphicsMagick is the swiss army knife of image processing.
- pgmagick is a yet another boost.python based wrapper for GraphicsMagick .

```
!pip install pgmagick
```

- SimpleITK is an open-source library that offers multi-dimensional image analysis.

```
!pip install SimpleITK
```

NLP Applications

- NLTK is a widely used library for developing Python applications that engage with natural human language data. NLTK requires Python 3.8, 3.9, 3.10, 3.11 or 3.12.

```
!pip install nltk
```

- Gensim is a Python library for topic modelling, document indexing and similarity retrieval with large corpora. Gensim's user-friendly interfaces enable effective multi-process implementations of well-known techniques, such as word2vec deep learning, online Latent Semantic Analysis (LSA/LSI/SVD), Latent Dirichlet Allocation (LDA), Random Projections (RP), and Hierarchical Dirichlet Process (HDP).

```
!pip install gensim
```

- spaCy is an open-source NL library designed for efficient and scalable processing of textual data. spaCy comes with pretrained pipelines and currently supports tokenization and training for 70+ languages.

```
!pip install spacy
```

- *TextBlob* is a Python library for processing textual data. It provides a simple API for diving into common NLP tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, and more.

```
from textblob import TextBlob
```

- PyNLPl, pronounced as 'pineapple', is a Python library for NLP. It contains various modules useful for common, and less common, NLP tasks. PyNLPl can be used for basic tasks such as the extraction of n-

grams and frequency lists, and to build simple language models. There are also more complex data types and algorithms.

```
!pip install pynlpl
```

OpenAI/LLMs

- The OpenAI Python library provides convenient access to the OpenAI REST API from any Python 3.8+ application.

```
# install from PyPI
!pip install openai
```

- llm 0.19.1 is a CLI utility and Python library for interacting with Large Language Models (LLMs), including OpenAI, PaLM and local models installed on your own machine.

Chatbots

- ChatterBot is a Python library designed to make it easy to create software that can engage in conversation. It uses a selection of machine learning algorithms to produce different types of responses. This helps create chatbots and automate conversations with users. Click here to install.
- DeepPavlov is an open-source conversational AI library built on TensorFlow and Keras. It has comprehensive and flexible tools that let developers and NLP researchers create production-ready conversational skills and complex multi-skill conversational assistants. Click here to install.

- Microsoft Bot Framework (MBF) offers an open-source platform for building bots. The MBF offers an impressive number of tools to aid the process of making a chatbot. It can also integrate with Luis, its natural language understanding engine.
- Botkit is just one part of a bigger set of developer tools and SDKs that encompass the MBF. The Bot Framework SDK provides the base upon which Botkit is built. It is available in multiple programming languages!
- The **Dialogflow** platform is behind the **Google Assistant** of your phone or smart speaker, you often use it. The entity is a product of Google that uses the same **NLP technology** that powers Google Assistant — which is the company's own virtual assistant. This feature makes it a powerful tool that is capable of understanding complex user queries and conversation contexts.

Geolocation

- **GeoPandas** extends the functionality of Pandas. It provides a high-level interface for working with vector data, such as points, lines, and polygons.
- **Folium** is a library for interactive mapping visualizations, allowing users to create and manipulate interactive maps using Python and Leaflet.js.
- **Streamlit** is a Python framework for developing web applications with geospatial functionality. Its a popular web framework, and provides tools for managing geospatial data and performing spatial queries.
- **GeoPy** is a library for geocoding and reverse geocoding, which allows you to convert addresses to geographic coordinates and vice versa. A library for geocoding and distance calculations.
- **Cartopy** is a Python library for geospatial data processing and visualization. It provides a simple and intuitive interface for creating maps and working with geospatial datasets.

```
!pip install geopy, streamlit, folium, geopandas, Cartopy
```

Much more Geo-Python Libraries can be explored here.

GUI Frameworks

1. PyQt5 is a Python library for creating GUI applications using the Qt toolkit.
2. Tkinter is Python's standard GUI framework.
3. Kivy is an open-source Python library that allows you to develop multi-platform graphical user interface applications on Windows, macOS, Android, iOS, Linux, and Raspberry-Pi.
4. wxPython is a blend of wxWidgets and Python programming library.
5. Libavg is a high-level development platform for media-centric applications. It allows programmers, media artists and designers to quickly develop media applications, uses Python as scripting language and is written in high-speed C++.

```
git clone https://github.com/libavg/libavg.git
```

```
!pip install PyQt5, tk-tools, Kivy, wxPython==4.0.0
```

Web Frameworks

- Django is a free, open-source Python framework that enables rapid development of complicated code and applications by programmers.

Python web developers can use it to create high-quality web apps.

- CherryPy is a lightweight, quick, and stable Python web development framework. It is open-source and can run on any Python-compatible framework.
- Pyramid web development platform is popular because of its adaptability and simplicity. Pyramid is written in Python 3.
- Grok is a web framework based on the technologies of the Zope toolkit (ZPK).
- TurboGears is a Python framework for data-driven, full-stack web applications.
- Web2Py includes a debugger, a code editor, and a deployment instrument for testing and maintaining web applications.
- Flask is a Python framework available under the BSD license. The Werkzeug WSGI toolkit and Jinja2 template are utilized by Flask.
- Bottle is one of the top Python web frameworks that belongs to the category of small-scale frameworks. It was first designed for constructing web APIs.

```
!pip install bottle, Flask, web2py, TurboGears2, grok, pyramid, CherryPy, Django
```

DevOps

- ansible — A radically simple IT automation platform.
- cloudinit — A multi-distribution package that handles early initialization of a cloud instance.
- openstack — Open source software for building private and public clouds.

- pyinfra — A versatile CLI tools and python libraries to automate infrastructure.
- saltstack — Infrastructure automation and management system.
- SSH-style Deployment
- cuisine — Chef-like functionality for Fabric.
- fabric — A simple, Pythonic tool for remote execution and deployment.
- supervisor — Supervisor process control system for UNIX.
- psutil — A cross-platform process and system utilities module.
- borg — A deduplicating archiver with compression and encryption.

Industry-Specific Applications of Data Science

- Below I rounded up 22 use-case examples of DS business applications seen today in various sectors ranging from FinTech to Government (read more here).

1. *FinTech*
2. *HealthTech*
3. *E-Commerce*
4. *Cybersecurity*
5. *Transportation*
6. *Disaster Risk Reduction*
7. *Digital Marketing & CRM*
8. *Retail Sales*
9. *Hospitality*

10. *Real Estate*
11. *Energy*
12. *Biotech & Pharma*
13. *Climate Change*
14. *Sports*
15. *Gaming*
16. *Social Media*
17. *Content Marketing*
18. *Automotive Industry*
19. *Smart Agriculture*
20. *LegalTech*
21. *Education*
22. *Government*

- From Data Analysis to Automation, Python is Transforming How Governments Operate
- Python prioritizes data security and privacy, critical aspects of government operations.
- USGS API is a Python module for interfacing with the US Geological Survey's API. It provides submodules to interact with various endpoints, and command line utilities, helpful for building out large pipelines.
- Python for Hydrology Self Study Curriculum: This repository contains study materials for python programming for hydrologic applications and a focus on groundwater modeling with flopy.

- datagovindia 1.0.2 Python API wrapper for Government of India Open Government Data (OGD) platform data.gov.in

The Final Word

- Python is a general-purpose and open source computer programming language. It is commonly used for both standalone programs and scripting applications in a wide variety of domains, by hundreds of thousands of developers worldwide.
- In this article, we have discussed the Comprehensive Repository of Awesome Python Libraries with Tutorials & Use-Case DS Examples across Industries.
- The Repository serves as a concise collection of DS projects to be a companion to other books, tutorials and other learning materials for aspiring data scientists.

Resources

- Awesome Python
- 30 Data Science Applications and Examples
- The most comprehensive Repository of Python Libraries for Data Science
- Top 26 Python Libraries for Data Science in 2024
- 10 Python Libraries Every Data Scientist Should Know
- Top 25 Python Libraries for Data Science in 2025
- Essential Python Libraries for Data Manipulation
- Top 15 Python Libraries for Data Science and Machine Learning
- 10 Automated EDA Tools That Will Save You Hours Of Work
- Essential Python Libraries for Statistics: A Practical Guide

- Python Statistics Fundamentals: How to Describe Your Data
- Python Web Scraping Tutorial
- Top 8 Most Important Unsupervised Machine Learning Algorithms With Python Code References
- 7 Best Libraries for Machine Learning Explained
- AutoML Python
- Explainable AI — Understanding and Trusting Machine Learning Models
- 7 Top NLP Libraries For NLP Development
- 70 Geospatial Python Libraries
- Top 12 Python GUI Frameworks for Developers
- Top 10 Python Web Development Frameworks in 2024

Explore More

- Basic Python Programming
- 100 Basic Python Codes
- DigHiSci Posts

Contacts

- Website
- GitHub
- Facebook
- X/Twitter
- Pinterest
- Mastodon

- Tumblr

Disclaimer

- I declare that this article is written by me and not with any AI content creation tools.
- I declare that no data privacy policy is breached, and that any data associated with the contents here are obtained legitimately to the best of my knowledge.

Python | Data Science | Libraries | Guides And Tutorials | Industry

Written by Alexzap

1.95K Followers · 1.96K Following

Data scientist, shareholder, investor, fintech, passionate about ML/AI, Python, and open-source knowledge sharing https://newdigitals.org/

Following

Responses (1)

What are your thoughts?

Respond

Larrimer Prestosa You
4 minutes ago

Greatly appreciated by curating this list.

Reply

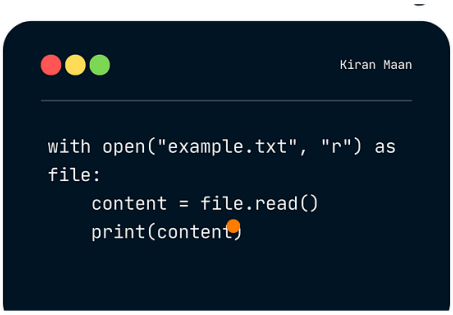
More from Alexzap

In Python in Plain English by Alexzap

Create 60+ Mind-Blowing Graphs, Charts, Maps & Animations with...

Part-of-Whole Interactive Data Visualization in Python

Dec 12 506 5



In Python in Plain English by Kiran Maan

10 Python Features That Seem Confusing (But Are Actually...

Python features that seem hard but are mind-blowing once you get them. Ready to master...

Dec 11 657 3

In Python in Plain English by Kiran Maan

The Python Module That Saved Me Hours of Work

Spoiler: It's not what you are thinking.

Nov 24 506 6

Alexzap

Using Skforecast for Multiple-ML Time Series Forecasting (TSF)...

Zero to Hero Use-Case Analysis of Skforecast TSF, Backtesting, Hyperparameter...

Nov 30 79 1

See all from Alexzap

Recommended from Medium

Terrill Toe

Quant Method: Cluster Correlation Analysis of Time Series Data

Analyzing the relationship between stocks and economic indicators using a novel...

Dec 20 218 8

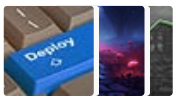
In Towards Data Science by Sergei Savvov

Your Company Needs Small Language Models

When specialized models outperform general-purpose models

3d ago 1.7K 18

Lists



Predictive Modeling w/ Python

20 stories · 1743 saves

Practical Guides to Machine Learning

10 stories · 2115 saves

Coding & Development

11 stories · 951 saves

ChatGPT prompts

51 stories · 2400 saves

Unicorn Day

The Hidden Patterns: How AI Reveals the True Groups in the...

Have you ever wondered how the biggest names in the stock market are secretly...

Dec 20 · 33 · 3

Robert McDermott

Saying Goodbye to Anaconda?

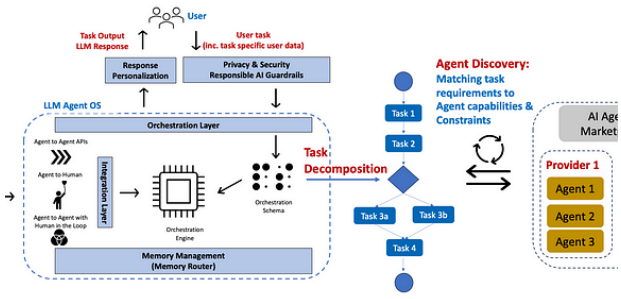
Finding a replacement for Conda

Oct 6 · 505 · 13

In Generative AI by Human×

I Let 1,000 Bots Try to Turn $20 into $52,000—Here's What Happened

A Trading Experiment



In AI Advances by Debmalya Biswas

AI Agents Marketplace & Discovery for Multi-agent Systems

3d ago 224 5

Why LLMs as a run-time execution engine for Agentic AI systems do not Scale?

2d ago 416 9

See more recommendations