

# Machine Learning Libraries in 2023

[BEGINNER](#)[DEEP LEARNING](#)[LEARNING PATH](#)[LIBRARIES](#)[LISTICLE](#)[MACHINE LEARNING](#)[PYTHON](#)[PYTORCH](#)[R](#)

## Introduction

With growing digitization, data is the lifeblood of the majority of organizations. As the existence of data-driven companies is expanding, the amount of data generated and accumulated by these companies is also expanding exponentially. Organizations are adapting to the newly emerging subset of artificial intelligence called machine learning to keep up with this data and continue making data-driven decisions. This emerging sub-field can unlock the undiscovered potential of organizations and their company data, enhancing business decisions. In this article, you will get to know about various machine learning libraries that you can use and optimise your work.

## Table of Contents

1. [What is Machine Learning?](#)
2. [Top 3 Languages Used in Machine Learning](#)
3. [Python](#)
4. [R Programming Language](#)
5. [MATLAB Programming Language](#)
6. [Top Python Libraries for ML and DL You Should Know in 2023](#)
7. [Fastai](#)
8. [OpenCV](#)
9. [Transformers](#)
10. [cuML](#)
11. [Scikit-Learn](#)
12. [PyTorch](#)
13. [TensorFlow](#)
14. [Keras](#)
15. [CNTK \(Microsoft Cognitive Toolkit\)](#)
16. [PyCaret](#)
17. [Conclusion](#)
18. [Frequently Asked Questions](#)
  - [Q1. Which library is used for machine learning?](#)
  - [Q2. Is Pandas a machine learning library?](#)
  - [Q3. What are AI ML libraries?](#)

## What is Machine Learning?

[Machine learning \(ML\)](#) is a field of computer science and artificial intelligence focusing on developing algorithms and statistical models that enable computers to learn from and make predictions or decisions based on data. The primary aim of machine learning is to enable computers to automatically improve performance on a specific task as more data is fed into the system.

Machine learning frameworks utilize numerous ML algorithms to perform tasks categorized into three main categories: supervised, unsupervised, and reinforcement learning tasks. The classification is based on how much the data is labeled/categorized. In supervised learning, algorithms are trained on labeled data; unlabeled data is used in unsupervised learning.

## Top 3 Languages Used in Machine Learning

Many programming languages can be used in machine learning. However, some are known to provide better efficiency and are more convenient to work with:-

### Python

[Python](#) is a high-level, all-purpose programming language. With a sizable development community and a broad range of applications, it has become one of the most well-liked languages in the world for novices. Because of its extensive library and framework ecosystem, Python makes it simple to create sophisticated applications rapidly. NumPy, Pandas, matplotlib, Django, Flask, TensorFlow, and PyTorch are well-known libraries and frameworks. Web development, data mining, machine learning, scientific computing, scripting, time series analysis, and data pretreatment and analysis.

### R Programming Language

Another programming language significantly used for statistical computing and machine learning is the R programming language. Developed in the 1990s, the programming language is mainly used in data analysis, visualization, and manipulation. It also has a large and active community of users and developers who contribute to its development and share their work through packages, which are collections of functions and data sets designed for specific tasks. With a large and active community of developers and users, its source code is freely available to everyone as an open-source language.

### MATLAB Programming Language

MATLAB is a proficient programming language and a computing environment for numerical, scientific, engineering, and machine-learning projects. It was developed in 1970 and is widely used in data modeling, analysis, and simulation. It has a comprehensive library of mathematical functions covering linear algebra, numerical analysis, matrix operations, and data visualization. It has a user-friendly interface and a suite of tools that helps developers in signal and image processing, control systems, and financial modeling. It is an excellent language with proprietary rights, implying its source code is not freely accessible.

## Top Python Libraries for ML and DL You Should Know in 2023

While many programming languages are useful in machine learning, Python programming language is the most widely used because it supports many frameworks, modules, neural networks, and multi-dimensional arrays.

**Some of the Best Python Libraries are listed below:**

Look at these Machine Learning Libraries and see where you can utilize them.

## Fastai

Fastai is a PyTorch-based open-source machine learning framework that offers high-level abstractions for deep learning model training. Various features, including data preprocessing, data augmentation, data manipulation, training, and inference using cutting-edge deep learning models, are available through the library.

It is highly recommended because

- **Robust Data Augmentation:** The library extensively generates more training data, improving model performance.
- **User-friendly Interface:** Fastai presents an intuitive API to ensure users can quickly build and train complex ML models.
- **Integration:** It is highly integrable with other libraries like PyTorch (its base) for facilitating the building and training of deep learning models.

While the FastAI library has many advantages, there are also some potential drawbacks.

- It is challenging for beginners because of a high-level abstraction layer.
- Offers limited customization.
- It has many dependencies.

### [Machine Learning Libraries in 2023](#)

```
%reload_ext autoreload
%autoreload 2
# Display images in the notebook
%matplotlib inline
# Import the vision library from fastai
from fastai.vision import *

#Helper functions from fastai docs
def get_ex(): return open_image('../input/licenseplate/111.PNG')

def plots_f(rows, cols, width, height, **kwargs):
    [get_ex().apply_tfms(tfms[0], **kwargs).show(ax=ax) for 1, ax in enumerate(plt.subplots(
        rows, cols, figsize=(width, height))[1].flatten())]
```

Source: Fast.ai forums

## OpenCV

OpenCV (Open Source Computer Vision) is an extensible, open-source computer vision and machine learning library that provides various tools and techniques for image and video analysis. It is a fantastic option for both beginning and expert machine learning developers due to its cross-platform compatibility, sizable community, and user-friendly UI.

Other benefits that OpenCV offers:

- - **A Suite of Tools and Techniques:** It provides various tools and techniques for image and video analysis, including image processing, object detection, face recognition, and optical character

recognition (OCR).

- **Free and Open-source:** OpenCV is a free and open-source library, meaning it can be used and modified by anyone without any licensing fees.

- **Integrable:** OpenCV is easily integrated with other Python libraries like TensorFlow and PyTorch.

Some disadvantages of working with OpenCV:

- Restricted deep learning support due to traditional algorithms.
- Suitable only for processing images and videos, limiting efficacy with other data types.
- Steep learning curve for beginners.



## Transformers

Hugging Face created the open-source Transformers library for machine learning. Modern natural language processing (NLP) models are provided that are simple to train and fine-tune for various NLP tasks, including text classification, question answering, and machine translation.

Transformers library offers

- **Colossal Community:** The Transformers library has a sizable and vibrant developer community that actively contributes to its development and offers users tools and support.
- **Highly Integrable:** The Transformers library can be easily integrated with popular machine learning libraries like PyTorch and TensorFlow.
- **Pre-trained Models:** Many pre-trained models in the Transformers library can be customized for different NLP needs. This saves much time and money compared to building models from scratch.

Although the Transformers library has many benefits, there are a few potential downsides to take into account as well:

- Although the Transformers library offers solid tools for natural language processing, it might not be as suitable for other sorts of data.
- Limited support for unsupervised learning.
- Extensive computational requirements.

```

from transformers import AutoModelForQuestionAnswering
from neural_compressor.config import PostTrainingQuantConfig
from optimum.intel.neural_compressor import INCQuantizer

model_name = "distilbert-base-cased-distilled-squad"
model = AutoModelForQuestionAnswering.from_pretrained(model_name)
save_dir = "quantized_model"
# Load the quantization configuration detailing the quantization we wish to apply
quantization_config = PostTrainingQuantConfig(approach="dynamic")
quantizer = INCQuantizer.from_pretrained(model)
# Apply dynamic quantization and save the resulting model
quantizer.quantize(quantization_config, save_directory=save_dir)

```

Source: Hugging Face

## cuML

NVIDIA created the open-source cuML library for machine learning. It offers GPU-accelerated techniques for various machine-learning tasks like classification, regression, clustering, and dimensionality reduction. Some of the key advantages of using the cuML library include

- **Processing considerable amounts of data:** The cuML library offers capabilities for processing massive amounts of data that would be challenging to process on CPU-based computers.
- **GPU acceleration:** The cuML library is designed to run on NVIDIA GPUs, providing significant speedups compared to CPU-based machine learning libraries.
- **Integration with other libraries:** The major machine learning libraries Scikit-learn, PyTorch, and TensorFlow can all be quickly connected with the cuML library.

Some disadvantages:

- Optimized for NVIDIA GPUs, it may be less efficient on non-NVIDIA hardware.
- Limited community support.
- Limited scalability.

```

import cudf
import cuml

# Load data into a cuDF DataFrame
X = cudf.read_csv("data.csv")

# Split data into features and labels
y = X['target_column']
X.drop(['target_column'], axis=1, inplace=True)

# Create a cuML Random Forest classifier
rf_classifier = cuml.ensemble.RandomForestClassifier()

# Train the classifier on the data
rf_classifier.fit(X, y)

# Make predictions on new data
new_data = cudf.read_csv("new_data.csv")
predicted_labels = rf_classifier.predict(new_data)

```

## Scikit-Learn

```
import time
from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

# Set training and validation sets
X, y = make_classification(n_samples=1000000, n_features=1000, n_classes = 2)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=10000)

# Solvers
solvers = ['liblinear', 'saga']

for sol in solvers:
    start = time.time()
    logreg = LogisticRegression(solver=sol)
    logreg.fit(X_train, y_train)
    end = time.time()
    print(sol + " Fit Time: ", end-start)

liblinear Fit Time:  2424.526051044464
saga Fit Time:  133.299968957901
```

[Scikit-learn](#) is one of the most popular machine learning libraries. It provides tools for building predictive models and performing data analysis.

Here are some of the critical features of scikit-learn and its application in machine learning:

- **Preprocessing and Feature Extraction:** Scikit-learn provides many tools for preprocessing data and extracting features from datasets.
- **Model Evaluation:** Scikit-learn offers a range of metrics for performance evaluation or various ML models, like predictive models, including accuracy, precision, and F1 score.
- **Supervised Learning:** Scikit-learn provides various algorithms for building predictive models from labeled data, including linear regression, logistic regression, decision trees, random forests, support vector machines (SVMs), and neural networks.

While scikit-learn is a powerful and widely used machine learning library, there are also some potential drawbacks:

- **Limited Support for Big Data:** Scikit-learn is designed to work with data that can fit into memory, which may need to be revised for extensive datasets.
- **Limited Support for Deep Learning:** Scikit-learn has limited support for deep learning algorithms compared to other libraries such as TensorFlow or PyTorch.

## PyTorch

Pytorch and most other deep learning frameworks do things a little differently than traditional linear algebra. It maps the rows of the input instead of the columns. That is, the  $i$ 'th row of the output below is the mapping of the  $i$ 'th row of the input under  $A$ , plus the bias term. Look at the example below.

```
# Author: Robert Guthrie

import torch
import torch.autograd as autograd
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim

torch.manual_seed(1)
```

```
lin = nn.Linear(5, 3) # maps from  $R^5$  to  $R^3$ , parameters  $A, b$ 
# data is 2x5.  $A$  maps from 5 to 3... can we map "data" under  $A$ ?
data = autograd.Variable(torch.randn(2, 5))
print(lin(data)) # yes
```

Out: Variable containing:  
 0.3130 0.2576 1.3546  
 1.8007 0.6433 0.4951  
 [torch.FloatTensor of size 2x3]

Torch is the foundation of the open-source, Python-based machine learning package known as PyTorch. In the subject of [deep learning](#), it is commonly employed. Using a straightforward and understandable API, PyTorch's dynamic computational graph enables developers to create and train neural networks. It is beneficial for producing:

- **Dynamic Computation Graphs:** PyTorch uses a dynamic computation graph that enables programmers to change the graph in real time while the program runs.
- **GPU Acceleration** is supported by PyTorch and can dramatically shorten training times for complex models.
- PyTorch offers automatic differentiation, simplifying the computation of gradients and optimizing model parameters during training.

However, there are also some potential drawbacks to consider:

- PyTorch has a steep learning curve, especially for users new to deep learning or neural networks.
- It may not scale as effectively as other deep learning libraries for large datasets.
- Limited model portability.

## TensorFlow



Source: tensorflow.org

One of the most well-known open-source machine learning libraries created by Google is called TensorFlow. The TensorFlow package provides the following:

- GPU acceleration.
- Automatic differentiation for computing gradients.
- A Hub for reusable machine-learning models.

It is helpful in deep learning models since it enables developers to build and train deep neural networks for numerous applications.

Tensors have broad applications:

- **Natural Language Processing (NLP):** TensorFlow may be used for NLP tasks like sentiment analysis and language translation.

- TensorFlow can create generative models like generative adversarial networks and variational autoencoders.
- **Computer Vision** can also benefit from this Python library.

Although TensorFlow is a solid and popular deep-learning library, there are a few potential downsides to take into account:

- Limited support for traditional machine-learning algorithms.
- It needs to be more scalable for distributed systems.
- Limited flexibility.

## Keras



Keras is a popular open-source deep learning library that provides a high-level API for building and training deep neural networks. It was made with an emphasis on rapid prototyping and experimentation. It was intended to be user-friendly and straightforward to use.

The following are some advantages of utilizing Keras:

- **Easy to Use:** Developers can rapidly and easily design and train deep neural networks using Keras thanks to its user-friendly API, eliminating the need for an in-depth understanding of the underlying mathematics.
- **Flexibility:** Keras supports many network topologies, including autoencoders, recurrent neural networks, and convolutional neural networks.
- **Portability:** TensorFlow, Microsoft Cognitive Toolkit, and Theano are just a few of the backends that Keras is compatible with. As a result, switching between backends is simple based on your unique use case.

While Keras is an easy-to-use deep learning library, there are also some potential drawbacks.

- It may provide less support than other libraries for specific specialized models, such as neural graph networks.
- Offers lesser advanced customization compared to PyTorch or TensorFlow.
- Limited research support.

## CNTK (Microsoft Cognitive Toolkit)



Microsoft developed the well-known open-source Microsoft Cognitive Toolkit for deep learning (CNTK). It is designed to handle both CPU and GPU processing. Deep neural network training is delivered with exceptional performance and scalability.

The following are some of the main advantages of CNTK in machine learning:

- **High Performance:** Using parallel computing architectures, it is performance-optimized and effectively handles massive datasets and intricate, deep neural networks.
- **Flexibility:** Deep learning models are supported for various applications, including object identification, picture classification, and natural language processing.
- It supports distributed training, which divides the training process among several machines.

While it has several advantages, there are also some disadvantages to consider:

- CNTK is highly optimized for specific use cases, such as image recognition, but may need to be more flexible.
- Microsoft has stated that it will no longer develop CNTK after 2020; therefore, it may not get any updates or new features.

```
import cntk as C
import numpy as np

# Define input and output dimensions
input_dim = 784
output_dim = 10

# Define input and output variables
input_var = C.input_variable(input_dim, np.float32)
output_var = C.input_variable(output_dim, np.float32)

# Define the neural network architecture
hidden_layer = C.layers.Dense(64, activation=C.relu)(input_var)
output_layer = C.layers.Dense(output_dim, activation=None)(hidden_layer)

# Define the loss function and evaluation metric
loss = C.cross_entropy_with_softmax(output_layer, output_var)
metric = C.classification_error(output_layer, output_var)

# Define the trainer and optimizer
learning_rate = 0.01
learner = C.sgd(output_layer.parameters, lr=learning_rate)
trainer = C.Trainer(output_layer, (loss, metric), [learner])
```

## PyCaret

In [26]:

```
#import the hepatitis dataset from PyCaret repository
from pycaret.datasets import get_data
data = get_data('hepatitis')
```

	Class	AGE	SEX	STEROID	ANTIVIRALS	FATIGUE	MALAISE	ANOREXIA	LIVER BIG	LIVER FIRM	S PAL
0	0	30	2	1.0	2	2	2	2	1.0	2.0	
1	0	50	1	1.0	2	1	2	2	1.0	2.0	
2	0	78	1	2.0	2	1	2	2	2.0	2.0	
3	0	31	1	NaN	1	2	2	2	2.0	2.0	
4	0	34	1	2.0	2	2	2	2	2.0	2.0	

PyCaret is an open-source, low-code machine learning library in Python that allows users to quickly

prototype, experiment, and deploy machine learning models.

Here are some key features and benefits of PyCaret:

- **Streamlined Machine Learning Workflow:** It provides a streamlined workflow for building, training, evaluating, and deploying machine learning models.
- **Low-code Interface:** It offers a low-code interface for machine learning, making it accessible to users with little or no programming experience.
- **Extensive Model Library:** [PyCaret](#) provides a comprehensive library of machine learning models, including regression, classification, clustering, and anomaly detection.

However, there are some disadvantages to consider:

- **Data Type Restrictions:** PyCaret is designed to handle shared data types and formats but may not provide for more complex data types.
- Offers little hyperparameter tuning.
- PyCaret automates many aspects of the machine learning processes. It can also make it more challenging to interpret the underlying models and algorithms.

## Conclusion

In conclusion, several solid machine-learning libraries for Python can make creating and deploying machine-learning models much more straightforward. These machine learning libraries include many functions, including model selection, hyperparameter tuning, data visualization, and data preprocessing. By utilizing these libraries, developers may speed up the machine learning process, save time and effort, and get better results.

For anyone interested in studying, sharing, and collaborating on a range of data science and analytics-related topics, Analytics Vidhya (AV) is a preeminent platform. Now that you know the best machine learning libraries in Python and want to learn more about them practically, you can check out the many tutorials and courses available at Analytics Vidhya. You can find a variety of materials there that can instruct you on how to reverse a string in Python, including articles, tutorials, courses, and contests. Overall, Analytics Vidhya can be a valuable resource for anyone looking to learn Python, Machine Learning, and Data Science, whether you're a beginner or an experienced professional.

## Frequently Asked Questions

### Q1. Which library is used for machine learning?

A. Numerous libraries are widely used in machine learning, and each of them offers a unique set of features and capabilities. Some of the most popular machine learning libraries include Keras, Scikit-Learn, PyTorch, TensorFlow, Matplotlib, NumPy, etc.

### Q2. Is Pandas a machine learning library?

A. Pandas is a prominent open-source library widely used for data science and machine learning tasks involving data manipulation and analysis. It is a flexible and versatile Python package that supports several data structures and mathematical operations.

### Q3. What are AI ML libraries?

A. AI/ML libraries are a framework comprising a set of routines and pre-defined functions written in commonly used programming languages. These libraries offer end-to-end software and application development technologies featuring artificial intelligence and machine learning for commercial uses.

---

Article Url - <https://www.analyticsvidhya.com/blog/2023/03/machine-learning-libraries-in-2023/>



**Swati Sharma**