MATPLOTLIB

Matplotlib is a plotting library for the Python programming language and its numerical math ematics extension NumPy. It provides an object-oriented API for embedding plots into applic ations using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+. There is als o a procedural "pylab" interface based on a state machine (like OpenGL), designed to closel y resemble that of MATLAB, though its use is discouraged.[3] SciPy makes use of matplotlib.

Matplotlib was originally written by John D. Hunter, has an active development community,[4] and is distributed under a BSD-style license. Michael Droettboom was nominated as matplot lib's lead developer shortly before John Hunter's death in August 2012,[5] and further join ed by Thomas Caswell[6][7]

As of 23 June 2017, matplotlib 2.0.x supports Python versions 2.7 through 3.6. Matplotlib 1 .2 is the first version of matplotlib to support Python 3.x. Matplotlib 1.4 is the last ver sion of matplotlib to support Python 2.6.[8]

Matplotlib has pledged to not support Python 2 past 2020 by signing the Python 3 Statement. [9]

Comparison with MATLAB
Pyplot is a matplotlib module which provides a MATLAB-like interface.[10] Matplotlib is des igned to be as usable as MATLAB, with the ability to use Python, and the advantage of being free and open-source.

********* EXAPMPLES *********
Line Plot

```
>>> import matplotlib.pyplot as plt
>>> import numpy as np
>>> a = np.linspace(0, 10, 100)
>>> b = np.exp(-a)
>>> plt.plot(a, b)
>>> plt.show()
```

Histogram

```
>>> import matplotlib.pyplot as plt
>>> from numpy.random import normal,rand
>>> x = normal(size=200)
>>> plt.hist(x, bins=30)
>>> plt.show()
```

Scatter plot

```
>>> import matplotlib.pyplot as plt
>>> from numpy.random import rand
>>> a = rand(100)
>>> b = rand(100)
>>> plt.scatter(a, b)
>>> plt.show()
```

3D plot

```
>>> from matplotlib import cm
>>> from mpl_toolkits.mplot3d import Axes3D
>>> import matplotlib.pyplot as plt
>>> import numpy as np
>>> fig = plt.figure()
>>> ax = fig.gca(projection='3d')
>>> X = np.arange(-5, 5, 0.25)
>>> Y = np.arange(-5, 5, 0.25)
>>> X, Y = np.meshgrid(X, Y)
>>> R = np.sqrt(X**2 + Y**2)
>>> Z = np.sin(R)
>>> surf = ax.plot_surface(X, Y, Z, rstride=1, cstride=1, cmap=cm.coolwarm)
>>> plt.show()
```

********** SCIPY **********

SciPy is a free and open-source Python library used for scientific computing and technical computing.

SciPy contains modules for optimization, linear algebra, integration, interpolation, specia
l functions, FFT, signal and image processing, ODE solvers and other tasks common in scienc
e and engineering.

SciPy builds on the NumPy array object and is part of the NumPy stack which includes tools
like Matplotlib, pandas and SymPy, and an expanding set of scientific computing libraries.
This NumPy stack has similar users to other applications such as MATLAB, GNU Octave, and Sc
ilab. The NumPy stack is also sometimes referred to as the SciPy stack.[3]

SciPy is also a family of conferences for users and developers of these tools: SciPy (in th
e United States), EuroSciPy (in Europe) and SciPy.in (in India).[4] Enthought originated th
e SciPy conference in the United States and continues to sponsor many of the international
conferences as well as host the SciPy website.

The SciPy library is currently distributed under the BSD license, and its development is sp
onsored and supported by an open community of developers. It is also supported by Numfocus
which is a community foundation for supporting reproducible and accessible science.

Scipy library

The SciPy package of key algorithms and functions core to Python's scientific computing cap
abilities. Available sub-packages include:

constants: physical constants and conversion factors (since version 0.7.0[5])
cluster: hierarchical clustering, vector quantization, K-means
fftpack: Discrete Fourier Transform algorithms
integrate: numerical integration routines
interpolate: interpolation tools
io: data input and output
lib: Python wrappers to external libraries
linalg: linear algebra routines
misc: miscellaneous utilities (e.g. image reading/writing)
ndimage: various functions for multi-dimensional image processing
optimize: optimization algorithms including linear programming
signal: signal processing tools
sparse: sparse matrix and related algorithms
spatial: KD-trees, nearest neighbors, distance functions
special: special functions
stats: statistical functions
weave: tool for writing C/C++ code as Python multiline strings

********** SCIKIT-LEARN **********

Scikit-learn (formerly scikits.learn) is a free software machine learning library for the P
ython programming language.[3] It features various classification, regression and clusterin
g algorithms including support vector machines, random forests, gradient boosting, k-means
and DBSCAN, and is designed to interoperate with the Python numerical and scientific librar
ies NumPy and SciPy.

The scikit-learn project started as scikits.learn, a Google Summer of Code project by David
Cournapeau. Its name stems from the notion that it is a "SciKit" (SciPy Toolkit), a separa
tely-developed and distributed third-party extension to SciPy.[4] The original codebase was
later rewritten by other developers. In 2010 Fabian Pedregosa, Gael Varoquaux, Alexandre G
ramfort and Vincent Michel, all from INRIA took leadership of the project and made the firs
t public release on February the 1st 2010.[5] Of the various scikits, scikit-learn as well
as scikit-image were described as "well-maintained and popular" in November 2012.[6]

As of 2018, scikit-learn is under active development.
Scikit-learn is largely written in Python, with some core algorithms written in Cython to a
chieve performance. Support vector machines are implemented by a Cython wrapper around LIBS
VM; logistic regression and linear support vector machines by a similar wrapper around LIBL
INEAR.