

Practical Neural Networks From Scratch

Concepts, Techniques and Tools to Build Intelligent Systems

Module 2

Beyond Normal Python:

Python For Data Science and Machine Learning

Ali Samanipour

May. 2023

1

Setup Your Development Environment

2

NUMPY & Pandas

3

MATPLOTLIB

4

SEABOARN

5

SCIPY

6

SCIKIT-LEARN

Development Environment Setup- Step 1

Download and install Python based on your OS.

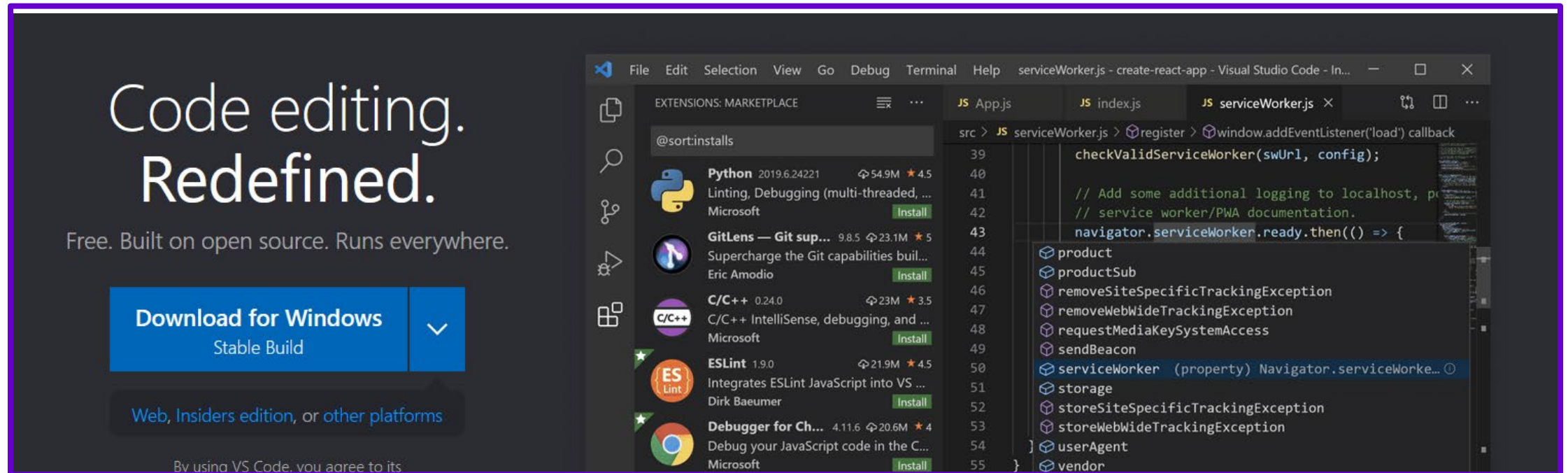
<https://www.python.org/downloads/>



Development Environment Setup- Step 2

Download and install VSCode based on your OS.

<https://code.visualstudio.com/>



The image shows the Visual Studio Code website on the left and the VS Code application interface on the right. The website features the text "Code editing. Redefined." and "Free. Built on open source. Runs everywhere." with a "Download for Windows" button. The application interface shows the "EXTENSIONS: MARKETPLACE" sidebar with a list of extensions including Python, GitLens, C/C++, ESLint, and Debugger for Chrome. The main editor area displays JavaScript code for a service worker.

Code editing.
Redefined.

Free. Built on open source. Runs everywhere.

Download for Windows
Stable Build

Web, Insiders edition, or other platforms

By using VS Code, you agree to its

EXTENSIONS: MARKETPLACE

@sortinstalls

- Python** 2019.6.24221 54.9M ★ 4.5
Linting, Debugging (multi-threaded, ...
Microsoft **Install**
- GitLens — Git sup...** 9.8.5 23.1M ★ 5
Supercharge the Git capabilities built into VS Code
Eric Amodio **Install**
- C/C++** 0.24.0 23M ★ 3.5
C/C++ IntelliSense, debugging, and more
Microsoft **Install**
- ESLint** 1.9.0 21.9M ★ 4.5
Integrates ESLint JavaScript into VS Code
Dirk Baeumer **Install**
- Debugger for Chrome** 4.11.6 20.6M ★ 4
Debug your JavaScript code in the Chrome DevTools
Microsoft **Install**

src > JS serviceWorker.js > register > window.addEventListener('load') callback

```
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55
```

checkValidServiceWorker(swUrl, config);

// Add some additional logging to localhost, pointing
// service worker/PWA documentation.

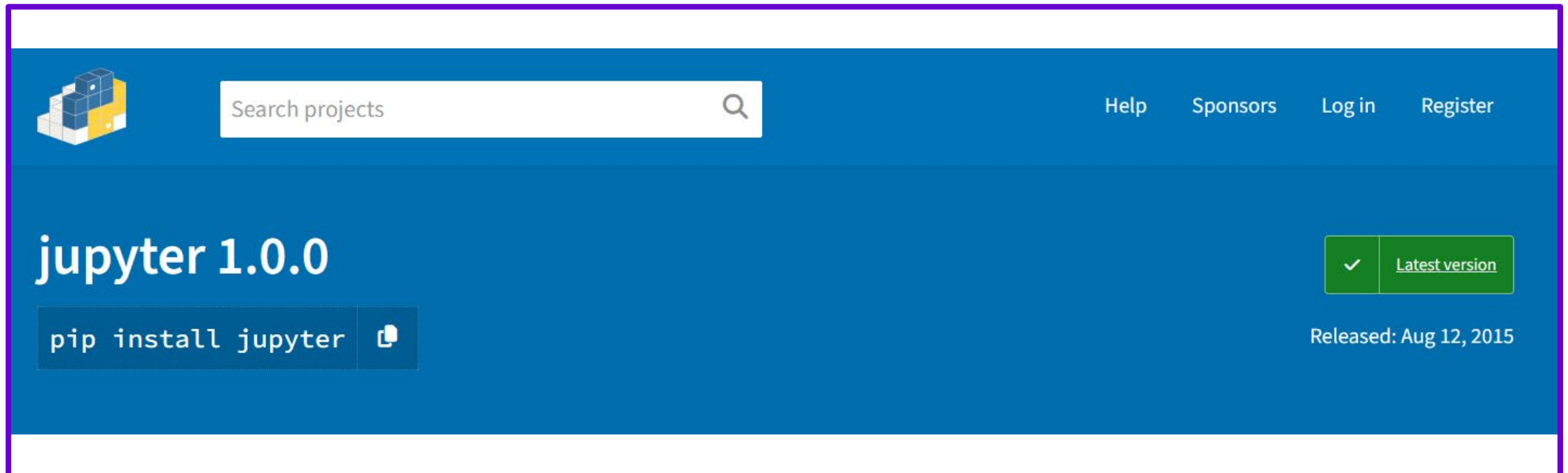
navigator.serviceWorker.ready.then(() => {

- product
- productSub
- removeSiteSpecificTrackingException
- removeWebWideTrackingException
- requestMediaKeySystemAccess
- sendBeacon
- serviceWorker (property) Navigator.serviceWorker...
- storage
- storeSiteSpecificTrackingException
- storeWebWideTrackingException
- userAgent
- vendor

Development Environment Setup- Step 3

Download and install Jupyter package.

<https://pypi.org/project/jupyter>

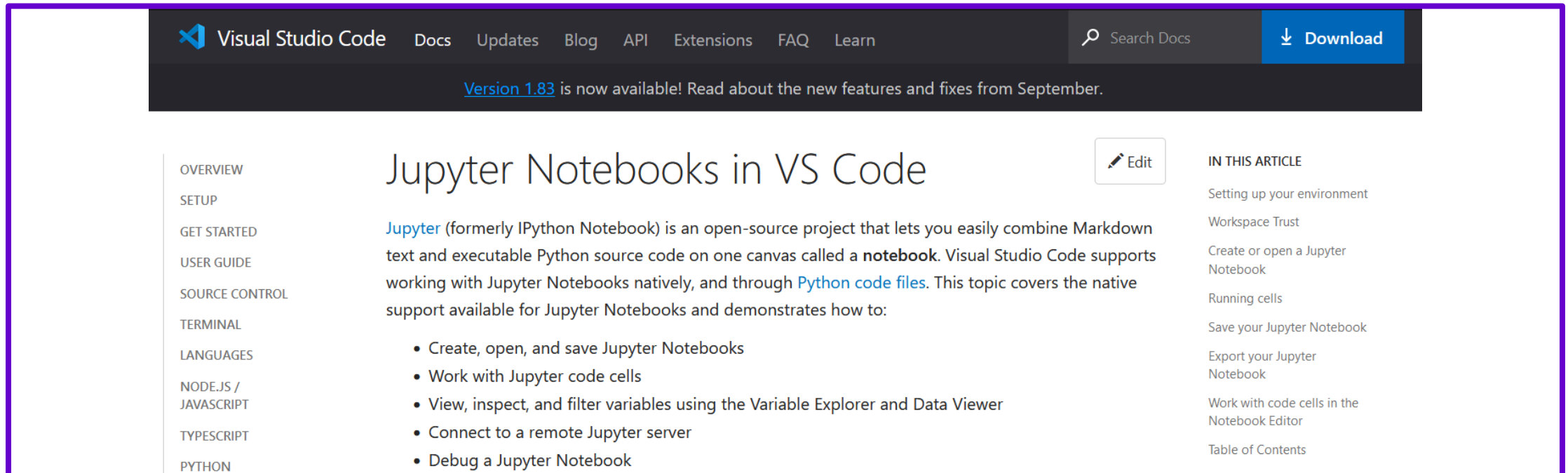


The screenshot shows the PyPI page for the Jupyter project. At the top left is the Jupyter logo (a stack of colored cubes). Next to it is a search bar with the text "Search projects" and a magnifying glass icon. To the right of the search bar are links for "Help", "Sponsors", "Log in", and "Register". Below the search bar, the text "jupyter 1.0.0" is displayed in a large font. To the right of this text is a green button with a checkmark and the text "Latest version". Below the version text, there is a dark blue button with the text "pip install jupyter" and a copy icon. To the right of this button, the text "Released: Aug 12, 2015" is displayed.

Development Environment Setup- Step 4

Download and install Jupyter Extension on VSCode.

<https://code.visualstudio.com/docs/datascience/jupyter-notebooks>



The screenshot shows the Visual Studio Code documentation website. At the top, there's a navigation bar with links for Visual Studio Code, Docs, Updates, Blog, API, Extensions, FAQ, and Learn. A search bar and a 'Download' button are also present. Below the navigation bar, a banner announces 'Version 1.83 is now available! Read about the new features and fixes from September.' The main content area is titled 'Jupyter Notebooks in VS Code' and includes an 'Edit' button. The text describes Jupyter as an open-source project for combining Markdown and Python code. A list of topics covered is provided: creating, opening, and saving notebooks; working with code cells; viewing and filtering variables; connecting to a remote server; and debugging notebooks. A sidebar on the left lists various documentation topics, and a right sidebar titled 'IN THIS ARTICLE' lists the specific topics covered in this document.

Visual Studio Code Docs Updates Blog API Extensions FAQ Learn

Search Docs Download

Version 1.83 is now available! Read about the new features and fixes from September.

Jupyter Notebooks in VS Code

[Jupyter](#) (formerly IPython Notebook) is an open-source project that lets you easily combine Markdown text and executable Python source code on one canvas called a **notebook**. Visual Studio Code supports working with Jupyter Notebooks natively, and through [Python code files](#). This topic covers the native support available for Jupyter Notebooks and demonstrates how to:

- Create, open, and save Jupyter Notebooks
- Work with Jupyter code cells
- View, inspect, and filter variables using the Variable Explorer and Data Viewer
- Connect to a remote Jupyter server
- Debug a Jupyter Notebook

IN THIS ARTICLE

- Setting up your environment
- Workspace Trust
- Create or open a Jupyter Notebook
- Running cells
- Save your Jupyter Notebook
- Export your Jupyter Notebook
- Work with code cells in the Notebook Editor
- Table of Contents

Alternative Development Environments

You could use anaconda, Jupyter, Google Colab, etc.



1

Setup Your Development Environment

2

NUMPY & Pandas

3

MATPLOTLIB

4

SEABOARN

5

SCIPY

6

SCIKIT-LEARN



NUMPY

NumPy is a Python library for **numerical computing**. It **provides powerful data structures**, such as n-dimensional arrays or "ndarrays", and a wide range of mathematical **functions for working with these arrays efficiently**.

Command

```
np.array([1,2,3])
```



NumPy Array

1
2
3

NUMPY

Pros

- Efficient and fast
- Powerful data structures
- Comprehensive mathematical functions
- Integration with other Python libraries

Cons

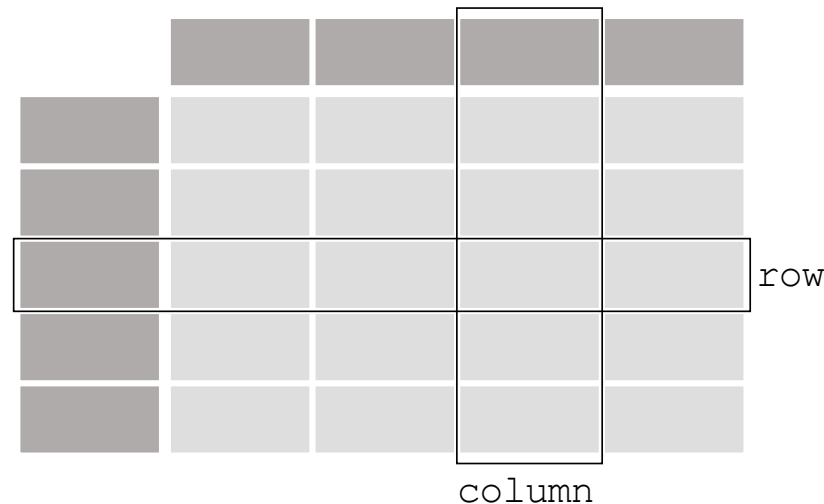
- Steep learning curve
- Memory usage
- Lack of flexibility



Pandas

Python Pandas is an open-source **data manipulation** and **analysis** library for the Python programming language. It provides a set of data structures for efficiently storing and manipulating large data sets, as well as a variety of tools for **data analysis**, **cleaning**, and **preprocessing**.

DataFrame



Pandas

Pros

- Easy-to-use and highly versatile
- Provides powerful tools for handling large datasets
- Supports a wide range of input and output formats
- Offers a rich set of data visualization tools
- Has a large and active community
- Can be used in conjunction with other popular data science libraries

Cons

- Pandas can be memory-intensive
- Some of the functions and methods can be complex
- Can be slow when performing certain operations
- May not always produce the desired results
- Some users have reported issues with compatibility and portability between different versions of Pandas

1

Setup Your Development Environment

2

NUMPY & Pandas

3

MATPLOTLIB

4

SEABOARN

5

SCIPY

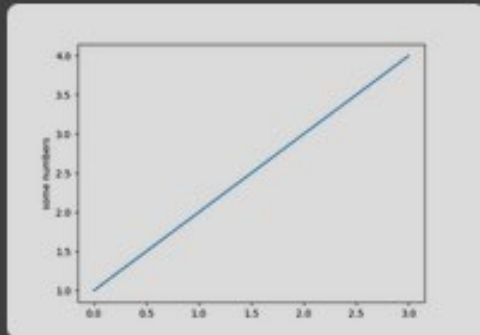
6

SCIKIT-LEARN



MATPLOTLIB

Matplotlib is a popular **data visualization** library for the Python programming language. It provides a way to create a wide range of **static, animated, and interactive visualizations** in Python.



Pyplot tutorial

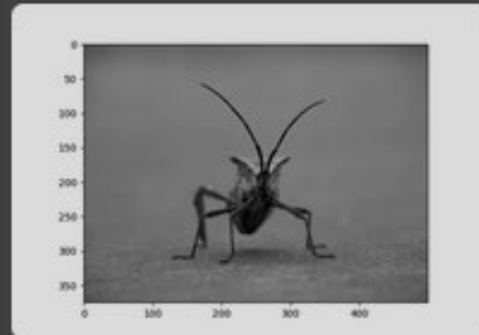
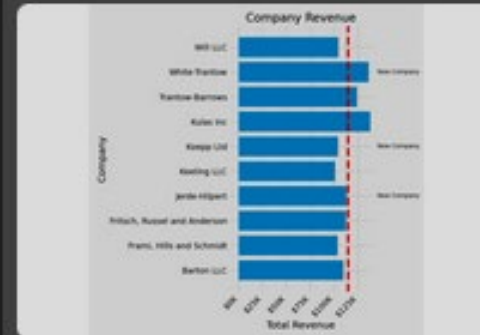
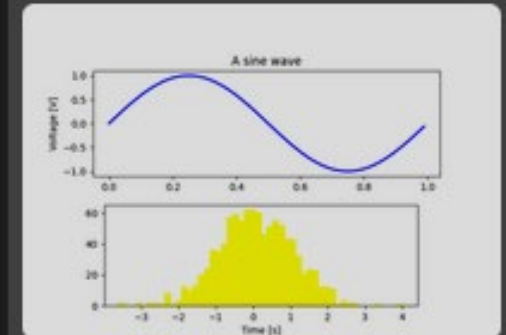


Image tutorial



The Lifecycle of a
Plot



Artist tutorial

MATPLOTLIB

Pros

- Matplotlib is a widely used
- It is highly customizable
- It provides a wide range of plotting functionality
- Can produce high-quality plots
- It is well integrated with other Python libraries
- It is easy to use and learn

Cons

- The syntax can be verbose and complex
- The default settings for plots may not always be aesthetically pleasing
- Does not provide as much interactivity
- It can be slower for generating complex
- Creating complex or advanced plots may require more coding

1

Setup Your Development Environment

2

NUMPY & Pandas

3

MATPLOTLIB

4

SEABOARN

5

SCIPY

6

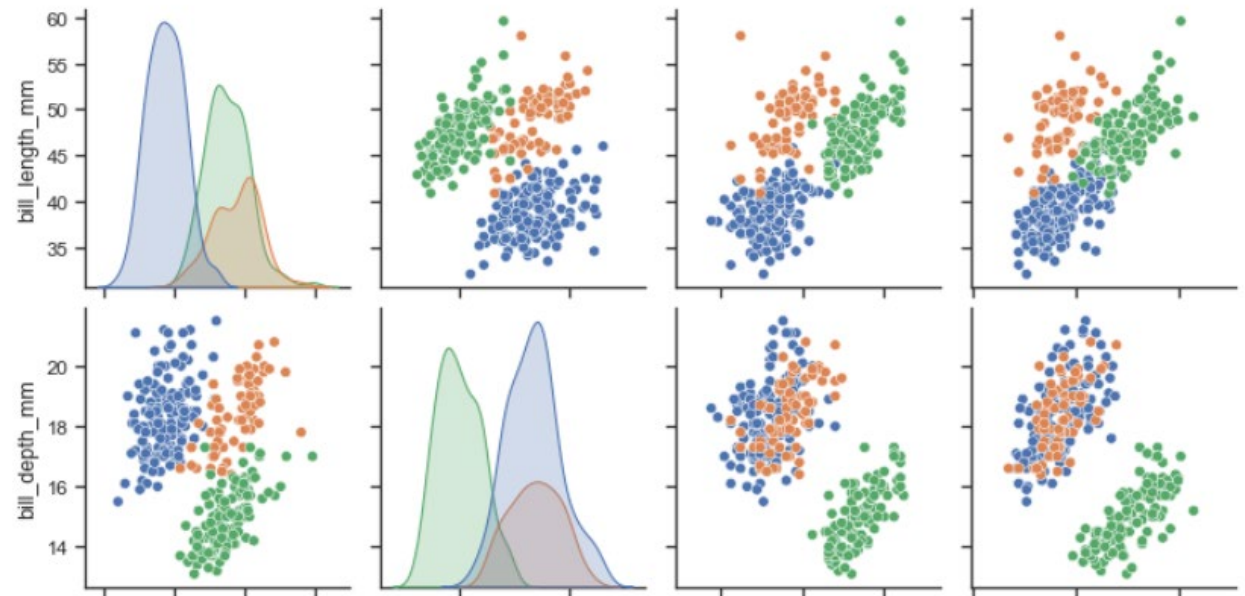
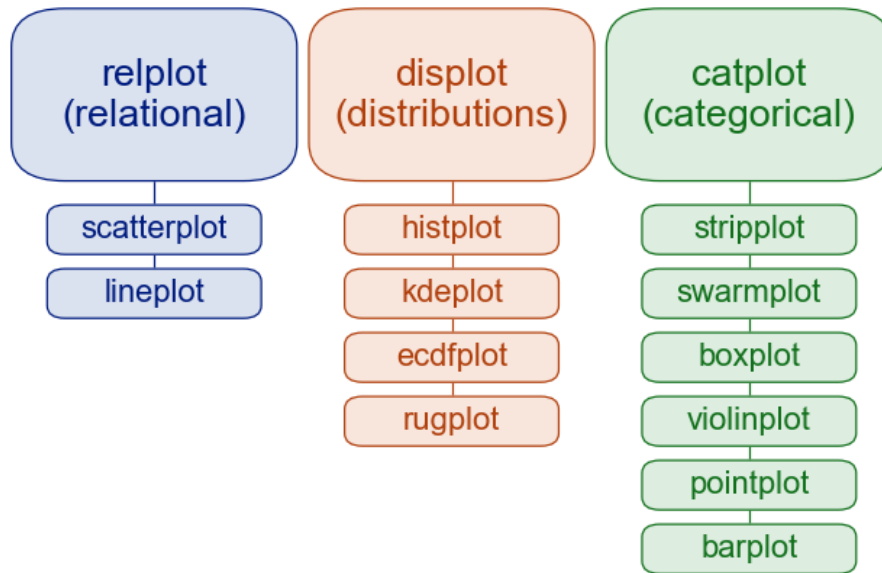
SCIKIT-LEARN



seaborn

SEABOARN

Seaborn is a Python **data visualization** library built **on top of Matplotlib**. It provides a high-level interface **for creating informative** and attractive **statistical graphics** in Python



SEABOARN

Pros

- Attractive and informative visualizations
- User-friendly interface
- Integration with Pandas
- Versatility: Seaborn offers a wide range of visualization techniques,

Cons

- Limited scope: Seaborn is focused on statistical data visualization
- Steep learning curve
- Limited customization options

1

Setup Your Development Environment

2

NUMPY & Pandas

3

MATPLOTLIB

4

SEABOARN

5

SCIPY

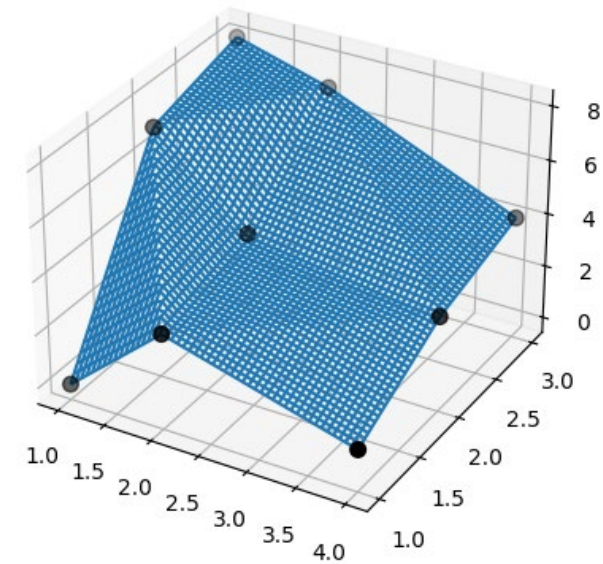
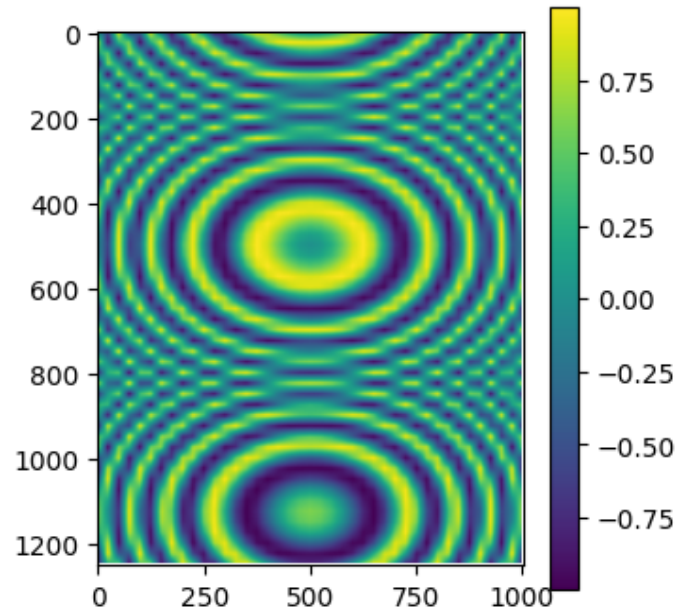
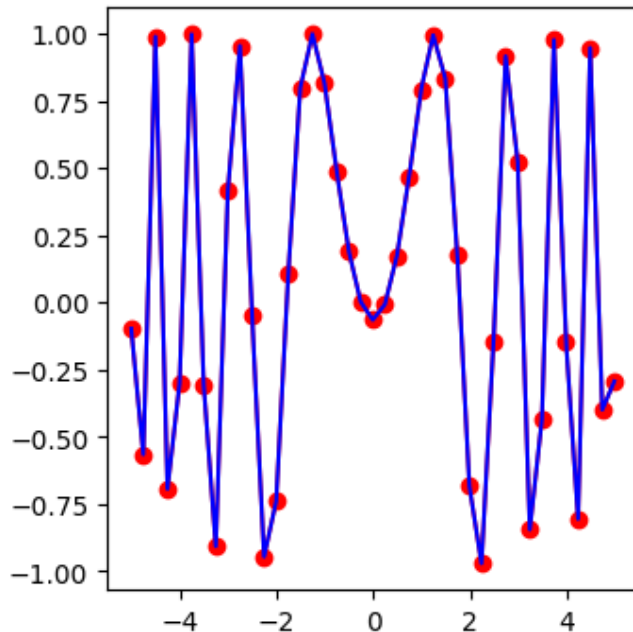
6

SCIKIT-LEARN



SCIPY

Scipy is an open-source **scientific computing** library for Python that provides a collection of **functions for mathematics, science, and engineering**



SCIPLY

Pros

- Provides a comprehensive set of tools for scientific computing and numerical analysis
- Built on top of NumPy
- efficient numerical operations
- Large and active community of users
- Well-documented with many examples
- Portable and cross-platform

Cons

- Can be complex and difficult to learn for beginners
- Some functions may be computationally intensive
- Some functions may have limitations
- Requires careful consideration of precision
- Some functions may not be as fast as optimized

1

Setup Your Development Environment

2

NUMPY & Pandas

3

MATPLOTLIB

4

SEABOARN

5

SCIPY

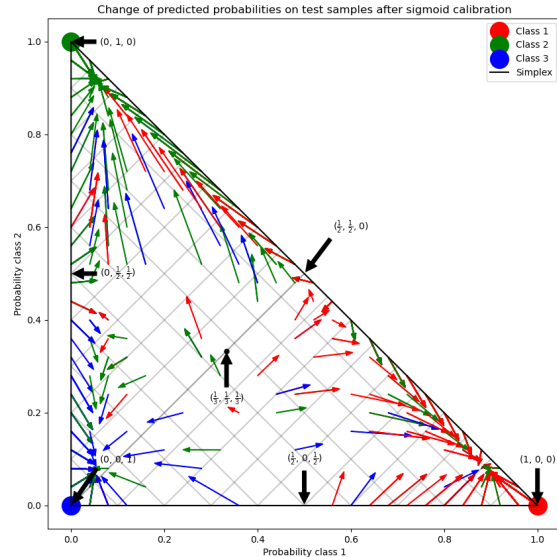
6

SCIKIT-LEARN

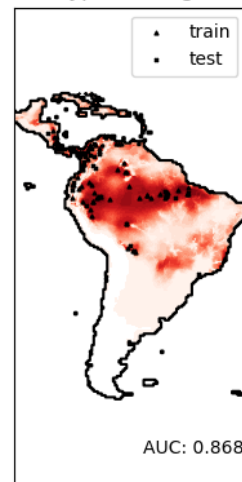


SCIKIT-LEARN

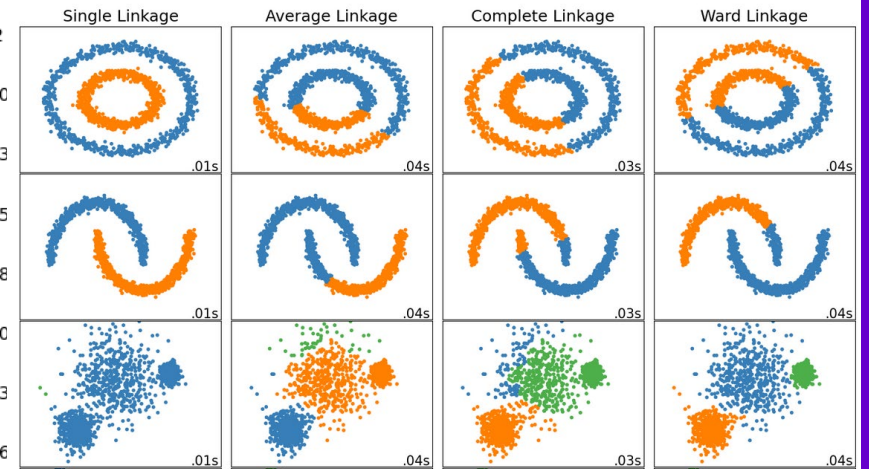
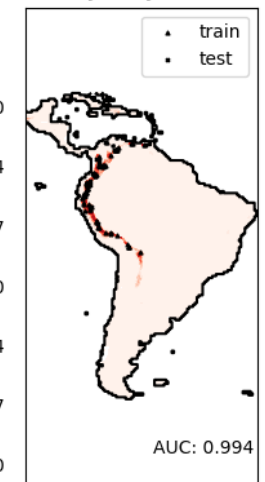
Python scikit-learn (also known as sklearn) is a **popular machine learning library** for the Python programming language. It provides a range of **supervised** and **unsupervised** learning algorithms



bradypus variegatus



microryzomys minutus



SCIKIT-LEARN

Pros

- It's a powerful and comprehensive machine learning library
- Scikit-learn is easy to use
- It is built on top of other popular scientific computing libraries for Python such as NumPy, SciPy, and matplotlib
- It provides a range of tools for data preprocessing
- Scikit-learn is well-documented
- It is open-source and free to use

Cons

- it may not be the best choice for some specific tasks or datasets that require more specialized algorithms or models
- It may not be the most efficient library for large-scale or complex machine learning tasks
- It does not include some newer or more advanced machine learning techniques
- Scikit-learn does not include built-in support for some popular machine learning frameworks such as TensorFlow or PyTorch

Course References

- [1] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Pearson, 2021.
- [2] T. Ghosh and S. K. B. Math, *Practical Mathematics for AI and Deep Learning: A Concise yet In-Depth Guide on Fundamentals of Computer Vision, NLP, Complex Deep Neural Networks and Machine Learning (English Edition)*. BPB Publications, 2022.
- [3] M. P. Deisenroth, A. A. Faisal, and C. S. Ong, *Mathematics for Machine Learning*. Cambridge University Press, 2020.
- [4] T. V. Geetha and S. Sendhilkumar, *Machine Learning: Concepts, Techniques and Applications*. CRC Press LLC, 2023.
- [5] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, 2023.
- [6] O. Theobald, *Machine Learning for Absolute Beginners: A Plain English Introduction (Third Edition)*. Scatterplot Press, 2021.

Accessing Course Resource



[linkedin.com/in/Samanipour](https://www.linkedin.com/in/Samanipour)



t.me/SamaniGroup



github.com/Samanipour