



Python 101

Basic python in industry

Hendri Karisma
Tech Lead at Finantier



Hendri Karisma

- Tech Lead for Platform team at Finantier
- Working for Encryption, Credit Scoring, Authentication and Authorization, and billing system
- Part time lecturer at Telkom University for Software Construction Course
- Before working as Research and Development team at an ecommerce for Ai team and developer experience team



What is Python?

- Programming language
- Free and Open source
- The Python Software Foundation is the organization behind Python. Become a member of the PSF and help advance the software and our mission.
- PEP (Python Enhancement Proposal) 8 is Style Guide for Python Code
- PyPI (Python Package Index) is a repository of software for the Python programming language.
- Dynamic Typed Language
- Could use OOP, Functional Paradigm, Parallel programming, async process, etc.
- <https://pypi.org> , <https://www.python.org/dev/peps/>
- Has a lot of good packages to support data science activities.



Setup python

- Direct use installed python on top operating system
- Virtual env
- Anaconda



Python for Software and AI Engineering

- Object Oriented Programming
- Functional Programming
- Rest api
- Web App

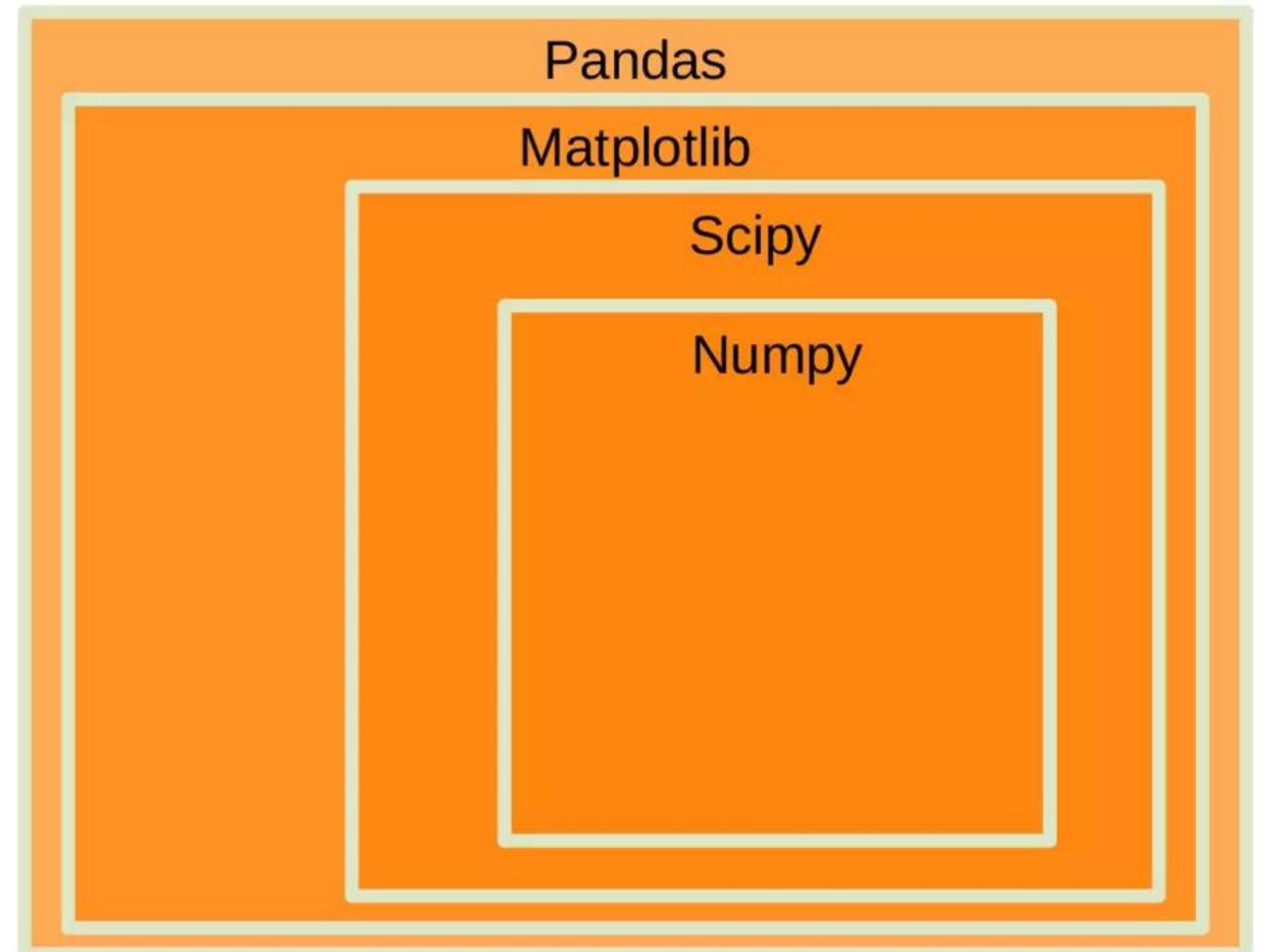
- Flask or FastApi
- db connector lib

For ML or AI Engineer :

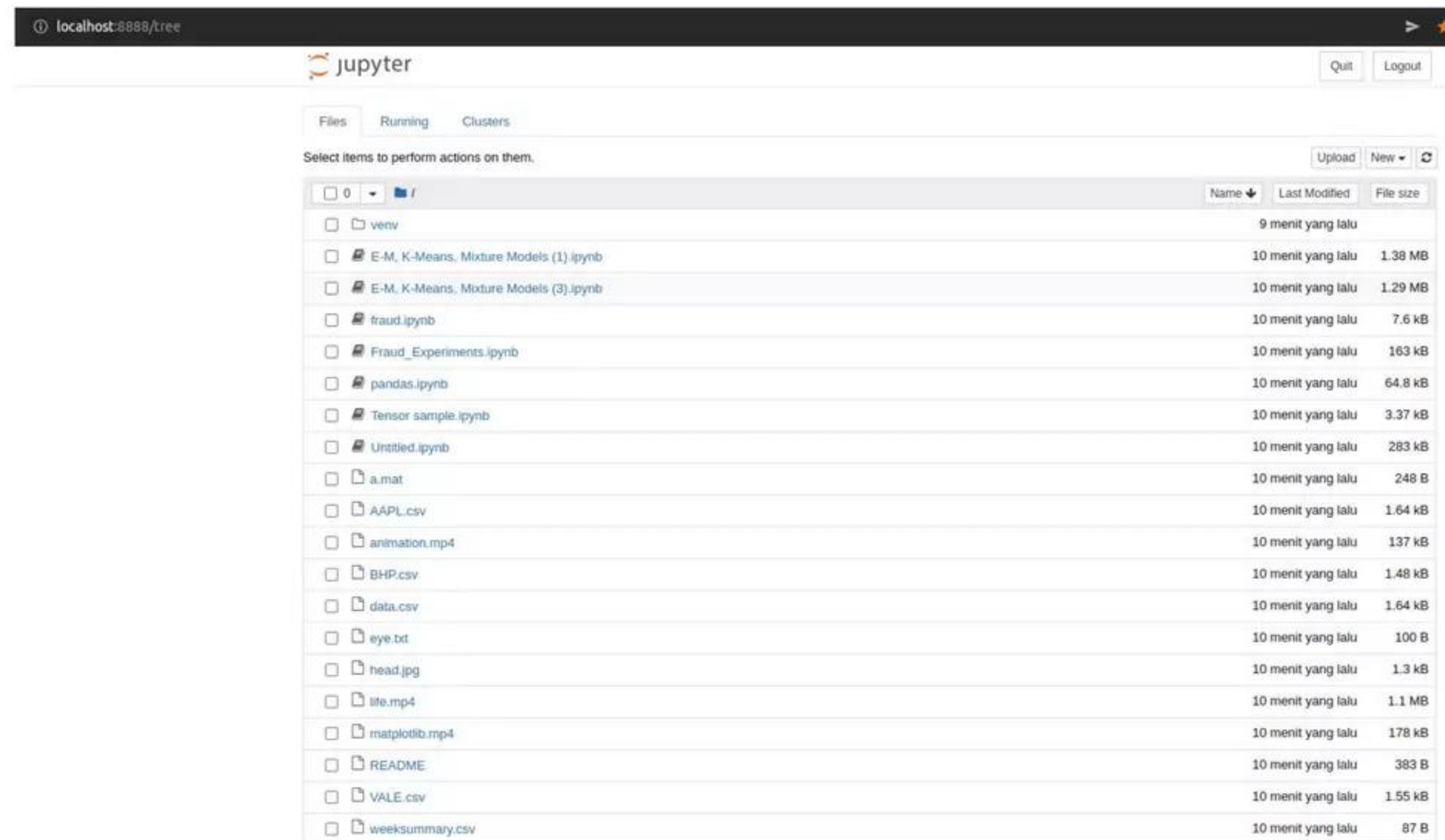
- pandas
- numpy
- scikit learn
- tensor
- pytorch

Python for Data science

- Jupyter notebook
- Numpy
- pandas
- Matplotlib / seaborn
- statsmodels
- scikit learn
- scrapy
- tensorflow / keras / pytorch

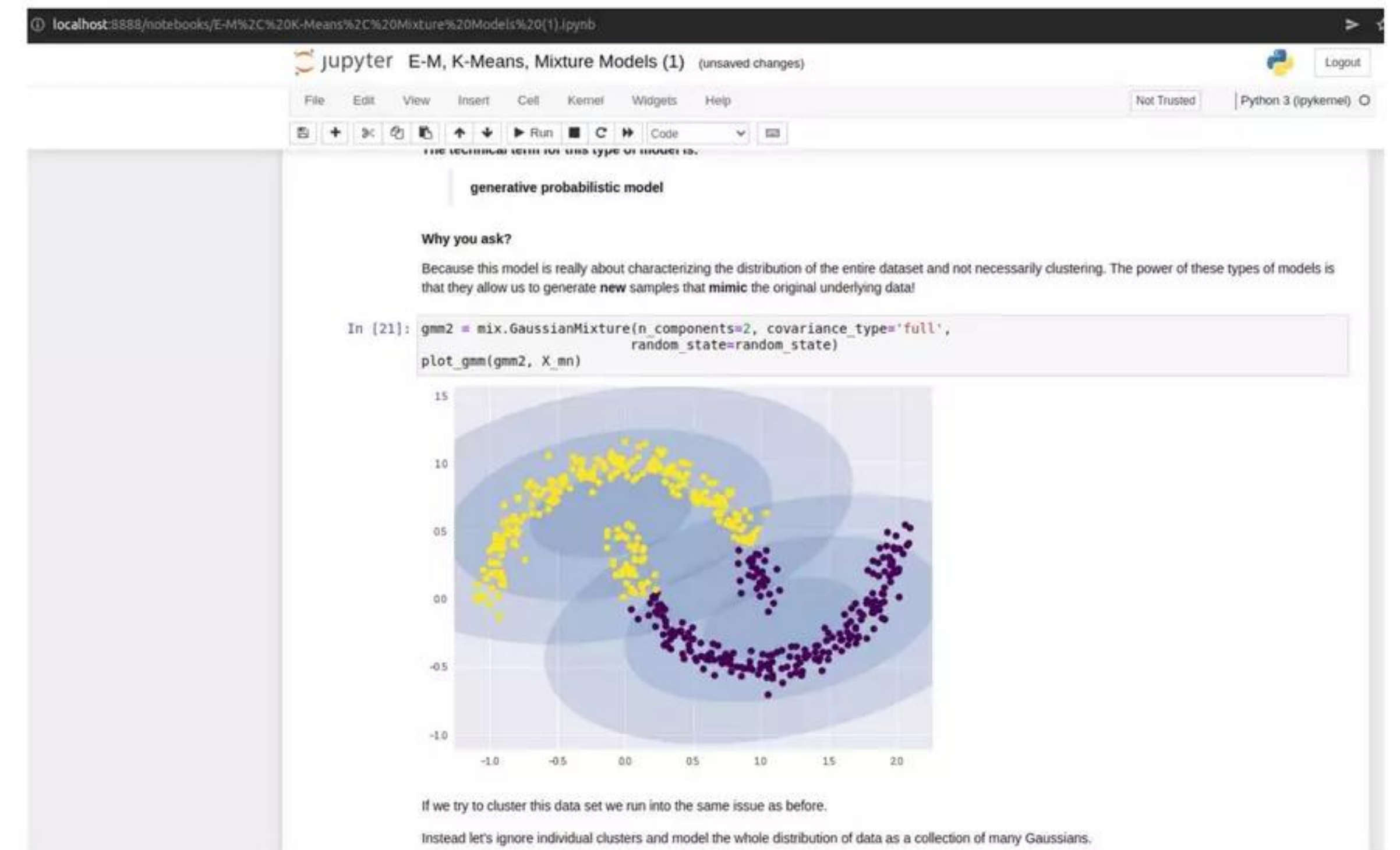


Jupyter notebook



The screenshot shows the JupyterLab file browser interface. The address bar indicates the current location is `localhost:8888/tree`. The interface includes a navigation bar with "Files", "Running", and "Clusters" tabs. Below the navigation bar, there is a prompt "Select items to perform actions on them." and buttons for "Upload" and "New". The main area displays a list of files and folders with columns for "Name", "Last Modified", and "File size".

Name	Last Modified	File size
venv	9 menit yang lalu	
E-M, K-Means, Mixture Models (1).ipynb	10 menit yang lalu	1.38 MB
E-M, K-Means, Mixture Models (3).ipynb	10 menit yang lalu	1.29 MB
fraud.ipynb	10 menit yang lalu	7.6 kB
Fraud_Experiments.ipynb	10 menit yang lalu	163 kB
pandas.ipynb	10 menit yang lalu	64.8 kB
Tensor sample.ipynb	10 menit yang lalu	3.37 kB
Untitled.ipynb	10 menit yang lalu	283 kB
a.mat	10 menit yang lalu	248 B
AAPL.csv	10 menit yang lalu	1.64 kB
animation.mp4	10 menit yang lalu	137 kB
BHP.csv	10 menit yang lalu	1.48 kB
data.csv	10 menit yang lalu	1.64 kB
eye.txt	10 menit yang lalu	100 B
head.jpg	10 menit yang lalu	1.3 kB
life.mp4	10 menit yang lalu	1.1 MB
matplotlib.mp4	10 menit yang lalu	178 kB
README	10 menit yang lalu	383 B
VALE.csv	10 menit yang lalu	1.55 kB
weeksummary.csv	10 menit yang lalu	87 B



The screenshot shows a Jupyter notebook cell titled "E-M, K-Means, Mixture Models (1) (unsaved changes)". The cell contains a code block and a plot. The code block defines a Gaussian Mixture Model (GMM) and plots the data points. The plot shows a scatter plot of data points with two clusters highlighted by semi-transparent ellipses. The x-axis ranges from -1.0 to 2.0, and the y-axis ranges from -1.0 to 1.5.

```
In [21]: gmm2 = mix.GaussianMixture(n_components=2, covariance_type='full',
random_state=random_state)
plot_gmm(gmm2, X_mn)
```

generative probabilistic model

Why you ask?

Because this model is really about characterizing the distribution of the entire dataset and not necessarily clustering. The power of these types of models is that they allow us to generate **new samples** that **mimic** the original underlying data!

If we try to cluster this data set we run into the same issue as before.
Instead let's ignore individual clusters and model the whole distribution of data as a collection of many Gaussians.



Term in Python

```
1 print('Hello World')
2
3 def power(x, y):
4     return x^y
5
6
7 if __name__ == '__main__':
8     print(f'result of 3 power 2 is {power(3,2)}')
9
10    is_odd_or_even = 'it\'s even number' if power(3,2) % 2 == 0 else 'it\'s odd number'
11
12    print(is_odd_or_even)
13
```



Collection

DataCollection

- List
- Dict

Data Collection Operations

- String Slicing and joining
- Sub list
- Reverse String
- String replace
- List sort
- List Filter
- Matrix
- List of Dict
- String formatting

```
>
2 first_list = [1, 2, 3, 4, 45, 56, 67, 7, 78, ]
3
4 second_list = ["hello", "gaes", "lets walking2"]
5
6 third_list = ["this is string", 50, True]
7
8 this_dict = {
9     "Name": "Tono",
10    "Age": 20,
11    "Address": "Pinggir Jalan"
12 }
13
14
15 print(first_list)
16
17 print(second_list)
18
19 print(third_list)
20
21 print(this_dict)
```

Result:

```
(venv) hendri@nvd-a11:~/python_projects/keynote$ python main.py
[1, 2, 3, 4, 45, 56, 67, 7, 78]
['hello', 'gaes', 'lets walking2']
['this is string', 50, True]
{'Name': 'Tono', 'Age': 20, 'Address': 'Pinggir Jalan'}
(venv) hendri@nvd-a11:~/python_projects/keynote$
```

Python function and procedure

```
>> def hitung_luas_segitiga(alas: int, tinggi: int) -> float:
3     return 0.5 * alas * tinggi
4
5
>> def hitung_luas_kotak(sisi):
7     return sisi * sisi
8
9
>> def procedure_print(nilai):
11    print("nilainya adalah : ", nilai)
12
13
>> luas_segitiga = hitung_luas_segitiga(10, 5)
15
16 print(luas_segitiga)
17
18 print(hitung_luas_kotak(10))
19
20
21 procedure_print(10000)
```

Result :

```
(venv) hendri@nvd-a11:~/python_projects/keynote$ python function.py
25.0
100
nilainya adalah : 10000
(venv) hendri@nvd-a11:~/python_projects/keynote$
```



Dasar OOP

- Class
 - Encapsulation
 - Inheritance
 - Polymorphism
- SOLID :
 - Single Responsibility Principle
 - Open-Close Principle
 - Liskov Substitution Principle
 - Interface Segregation Principle
 - Dependency Inversion Principle



Numpy

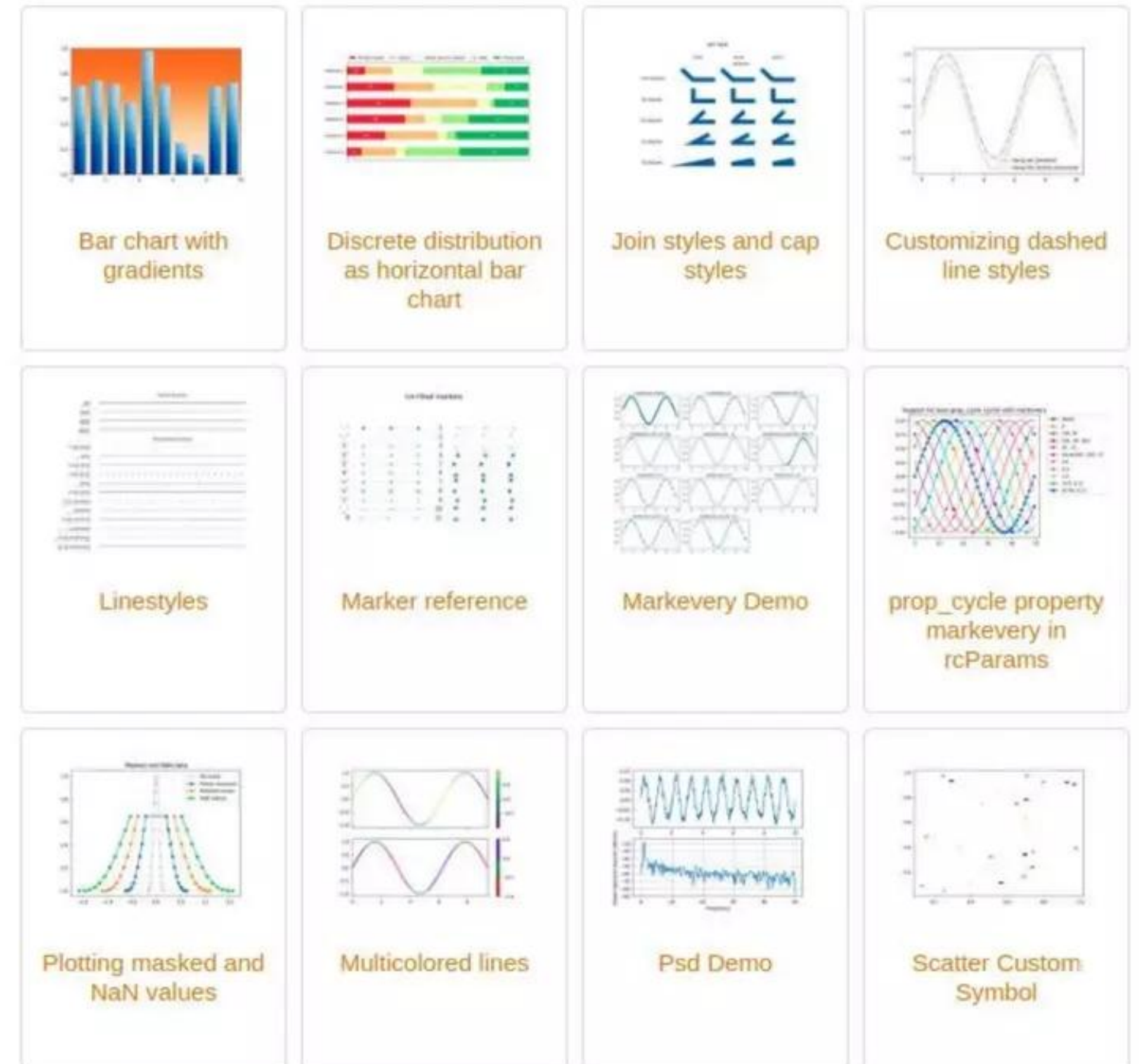
- Nddarray and matrix
- Operations : manipulation, arithmetic, math and
- statistical function
- visualization
- I/O Numpy
- scipy



Basic Matplotlib

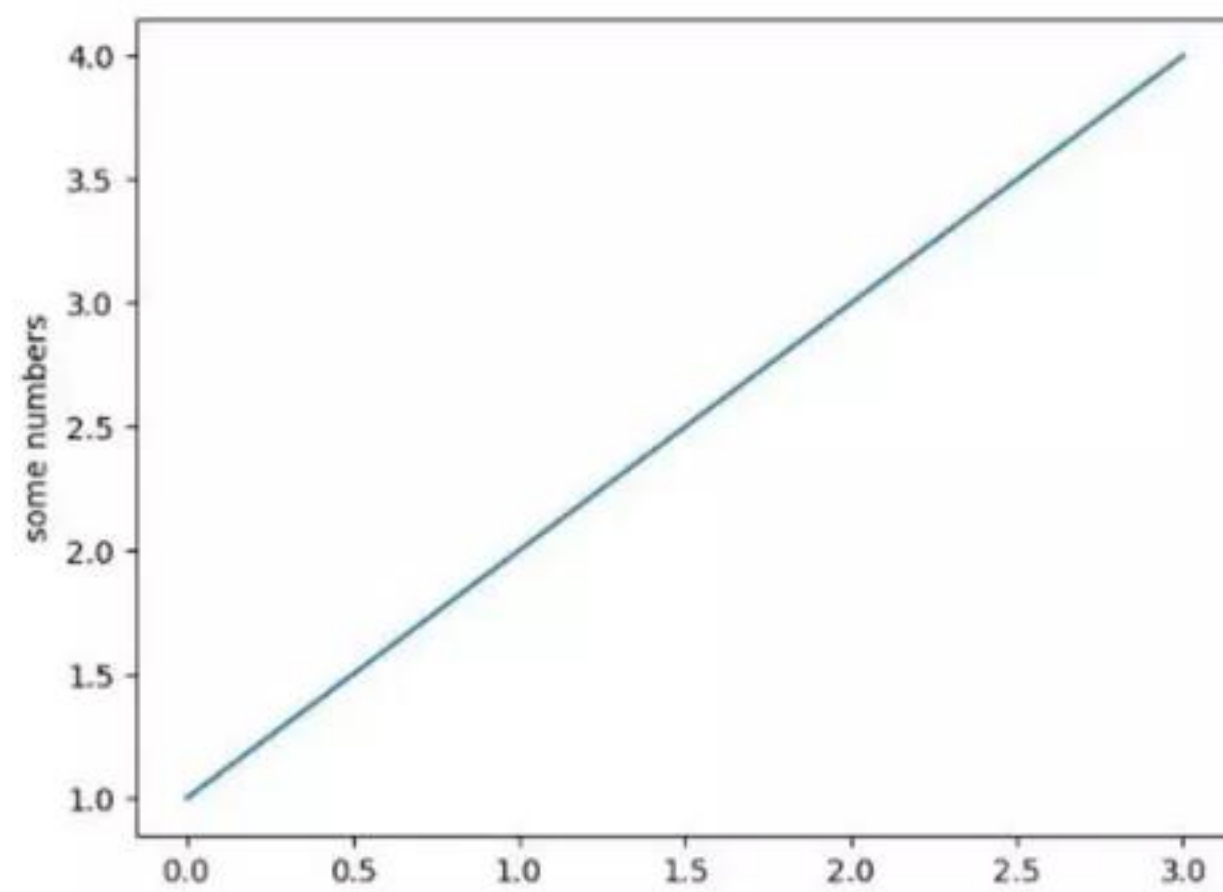
matplotlib.pyplot is a collection of functions that make matplotlib

work like MATLAB

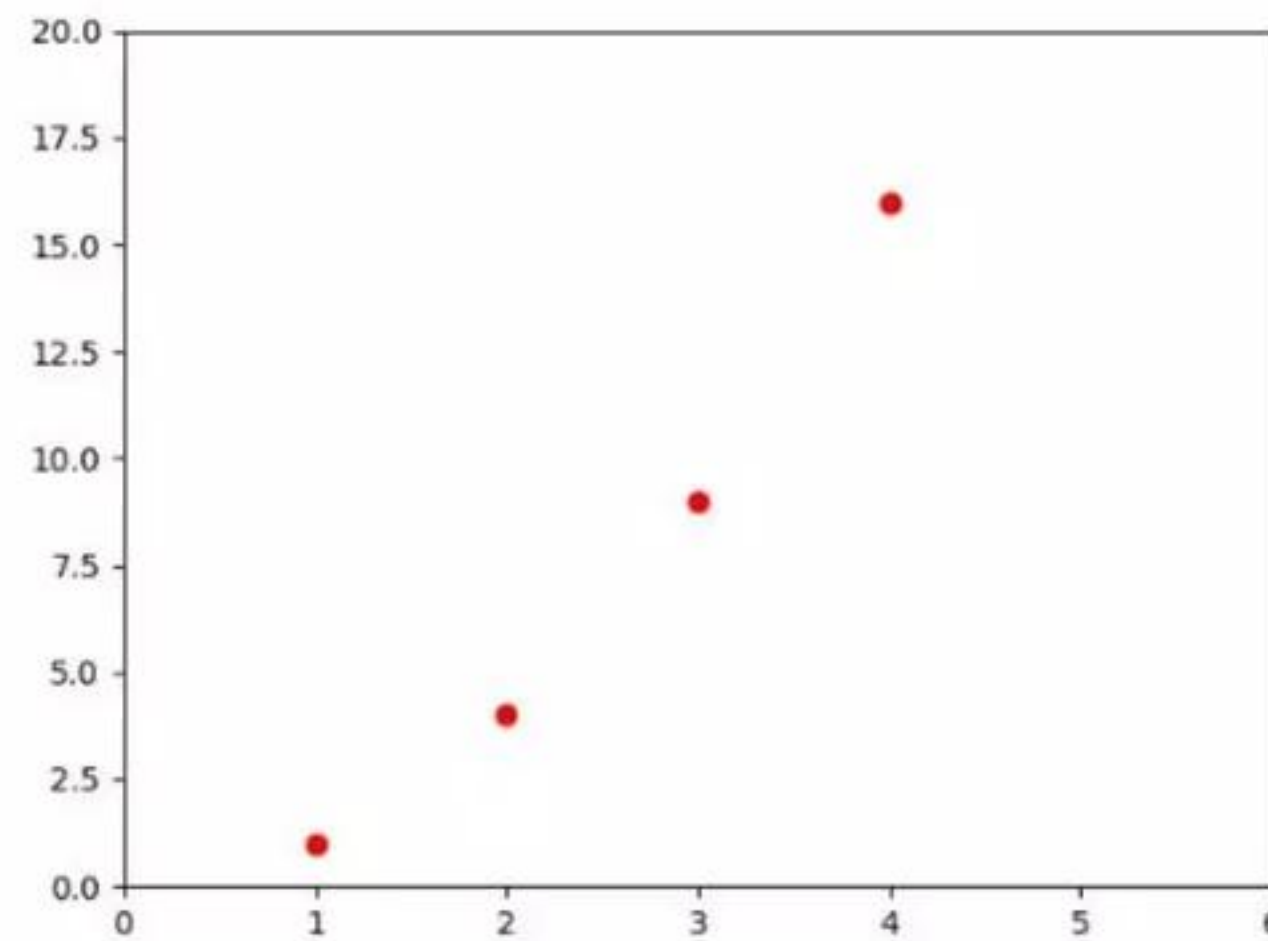


Basic Matplotlib #2

```
import matplotlib.pyplot as plt
plt.plot([1, 2, 3, 4])
plt.ylabel('some numbers')
plt.show()
```

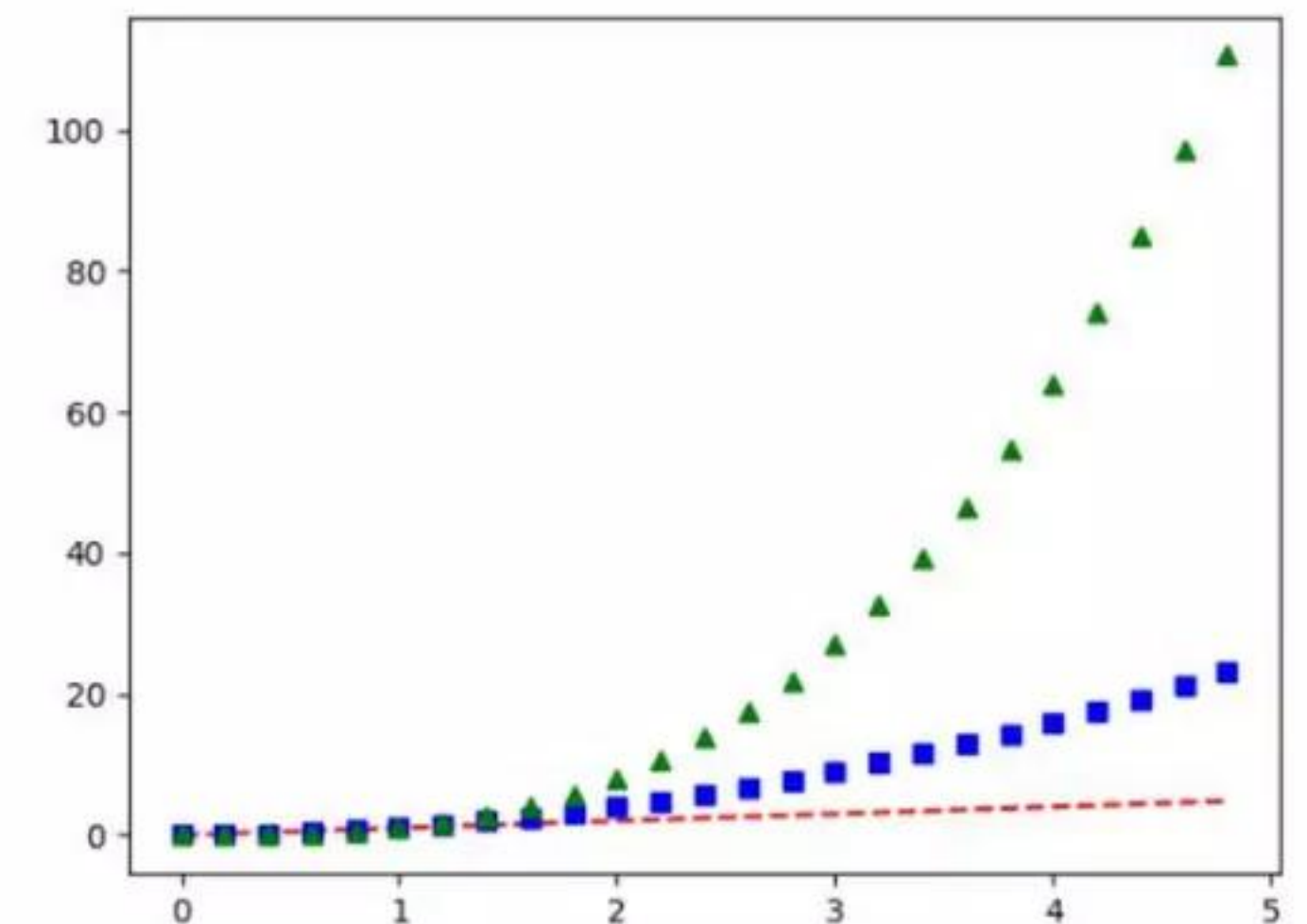


```
plt.plot([1, 2, 3, 4], [1, 4, 9, 16], 'ro')
plt.axis([0, 6, 0, 20])
plt.show()
```



```
import numpy as np
# evenly sampled time at 200ms intervals
t = np.arange(0., 5., 0.2)

# red dashes, blue squares and green triangles
plt.plot(t, t, 'r--', t, t**2, 'bs', t, t**3, 'g^')
plt.show()
```



Pandas

All Images Videos News Maps More Settings Tools

About 592,000,000 results (0.57 seconds)

pandas.pydata.org

pandas - Python Data Analysis Library

pandas. pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.

Documentation

User Guide - Getting started - API reference - ...

10 Minutes to Pandas

This is a short introduction to pandas, geared mainly for new ...

Installation

Installation. The easiest way to install pandas is to install it as ...

More results from pydata.org »

Getting started

Getting started. Installation instructions. The next steps ...

pandas.DataFrame

pandas.DataFrame. class pandas.DataFrame (data=None, index ...

User Guide

10 Minutes to Pandas - Intro to data structures - Visualization

People also ask

What is the use of pandas?

What is difference between NumPy and pandas?

How can I learn panda?

Is pandas hard to learn?



pandas

Software

In computer programming, pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. [Wikipedia](#)

Developer: Wes McKinney

License: New BSD License

Initial release: 11 January 2008; 12 years ago

Stable release: 1.1.5 / 7 December 2020; 9 days ago

Programming languages: Python, C, Cython

People also search for

View 15+ more



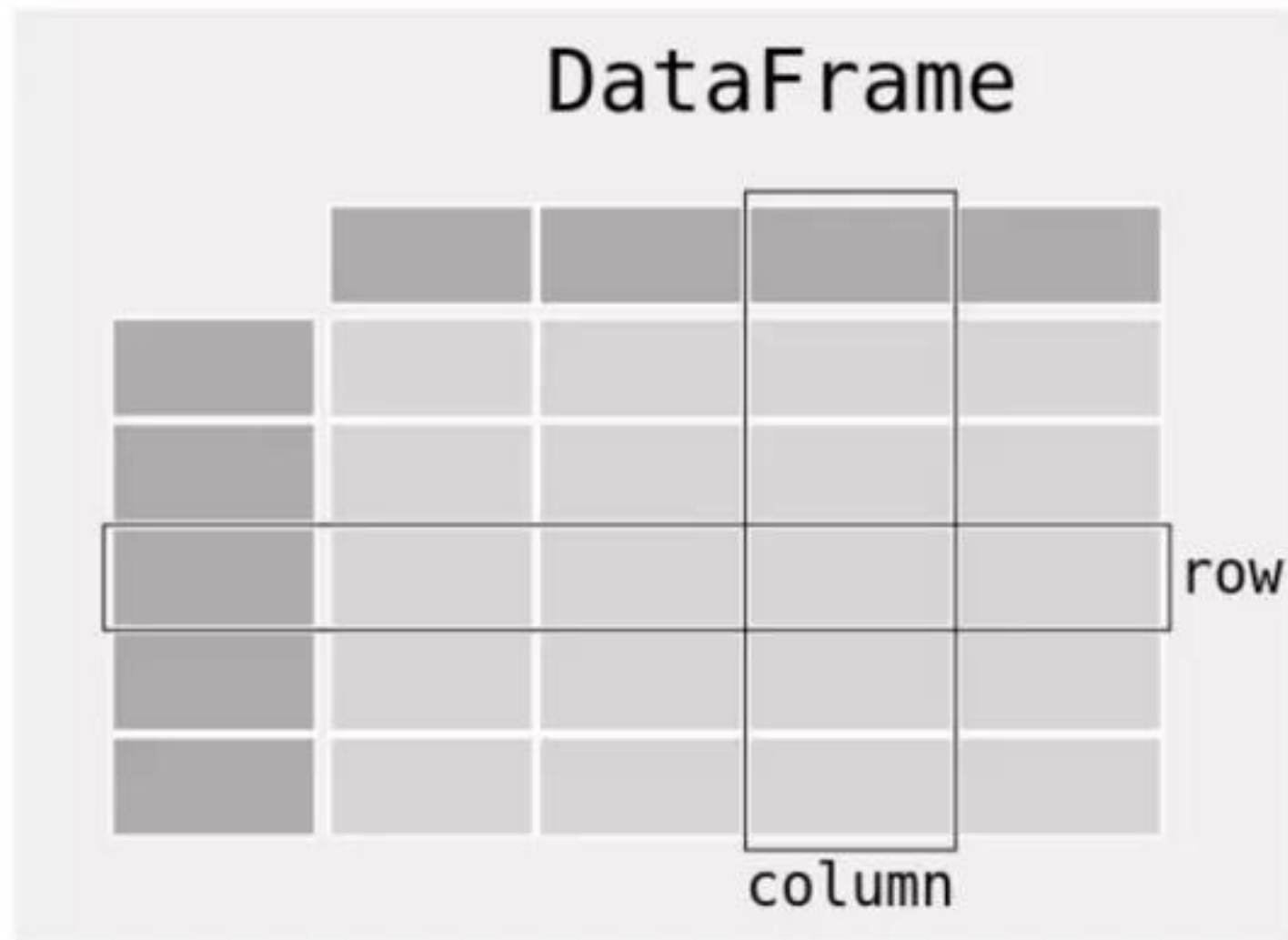
Feedback



Basic Pandas

- Intro and setup pandas
- Dataframe and Data structure
- Basic Function
- Data Duplicate, Join, Merge, and Subset
- Grouping
- Time Series
- Method Chaining
- Visualization

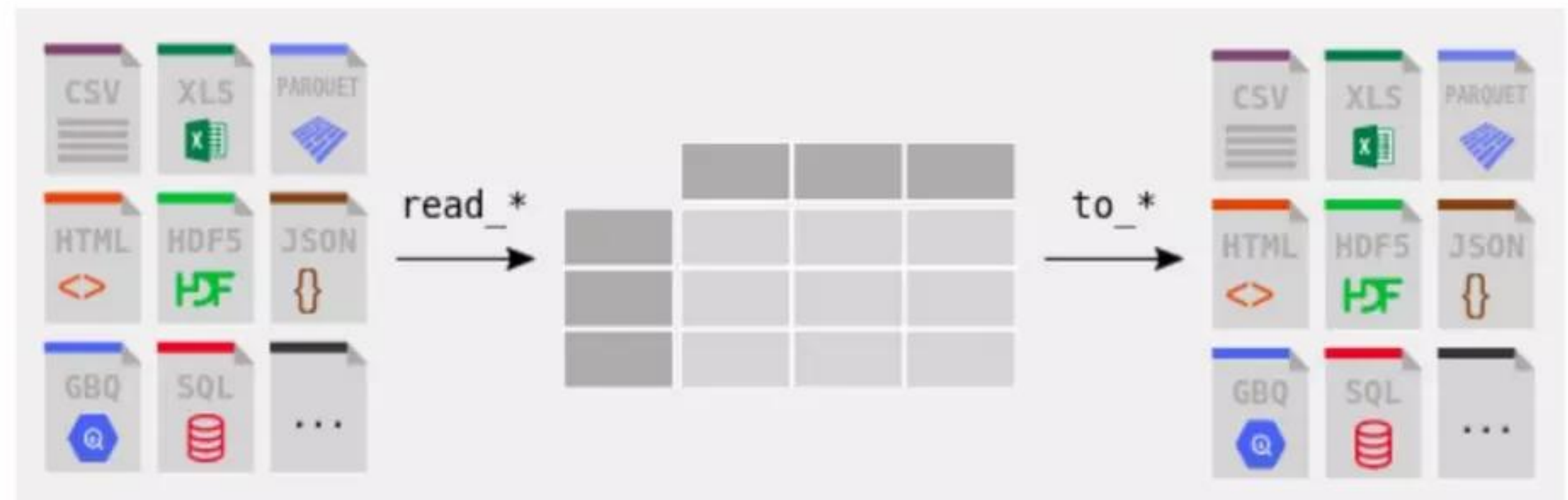
Basic Pandas



```
iris = pd.read_csv("data/iris.data")
```

Import pandas as pd

```
df = pd.DataFrame({  
    "name": ["a", "b", "c"],  
    "age": [20, 21, 22],  
    "sex": ["male", "female", "male"]  
})
```



Scikit-Learn

scikit-learn.org/stable/modules/classes.html



Prev Up Next

scikit-learn 0.24.0
Other versions

Please [cite us](#) if you use the software.

API Reference

- [sklearn.base](#): Base classes and utility functions
- [sklearn.calibration](#): Probability Calibration
- [sklearn.cluster](#): Clustering
- [sklearn.compose](#): Composite Estimators
- [sklearn.covariance](#): Covariance Estimators
- [sklearn.cross_decomposition](#): Cross decomposition
- [sklearn.datasets](#): Datasets
- [sklearn.decomposition](#): Matrix Decomposition
- [sklearn.discriminant_analysis](#): Discriminant Analysis
- [sklearn.dummy](#): Dummy estimators
- [sklearn.ensemble](#): Ensemble Methods
- [sklearn.exceptions](#): Exceptions and warnings
- [sklearn.experimental](#): Experimental
- [sklearn.feature_extraction](#): Feature Extraction
- [sklearn.feature_selection](#): Feature Selection

API Reference

This is the class and function reference of scikit-learn. Please refer to the [full user guide](#) for further details, as the class and function raw specifications may not be enough to give full guidelines on their uses. For reference on concepts repeated across the API, see [Glossary of Common Terms and API Elements](#).

sklearn.base: Base classes and utility functions

Base classes for all estimators.

Base classes

base.BaseEstimator	Base class for all estimators in scikit-learn.
base.BiclusterMixin	Mixin class for all bicluster estimators in scikit-learn.
base.ClassifierMixin	Mixin class for all classifiers in scikit-learn.
base.ClusterMixin	Mixin class for all cluster estimators in scikit-learn.
base.DensityMixin	Mixin class for all density estimators in scikit-learn.
base.RegressorMixin	Mixin class for all regression estimators in scikit-learn.
base.TransformerMixin	Mixin class for all transformers in scikit-learn.
feature_selection.SelectorMixin	Transformer mixin that performs feature selection given a support mask

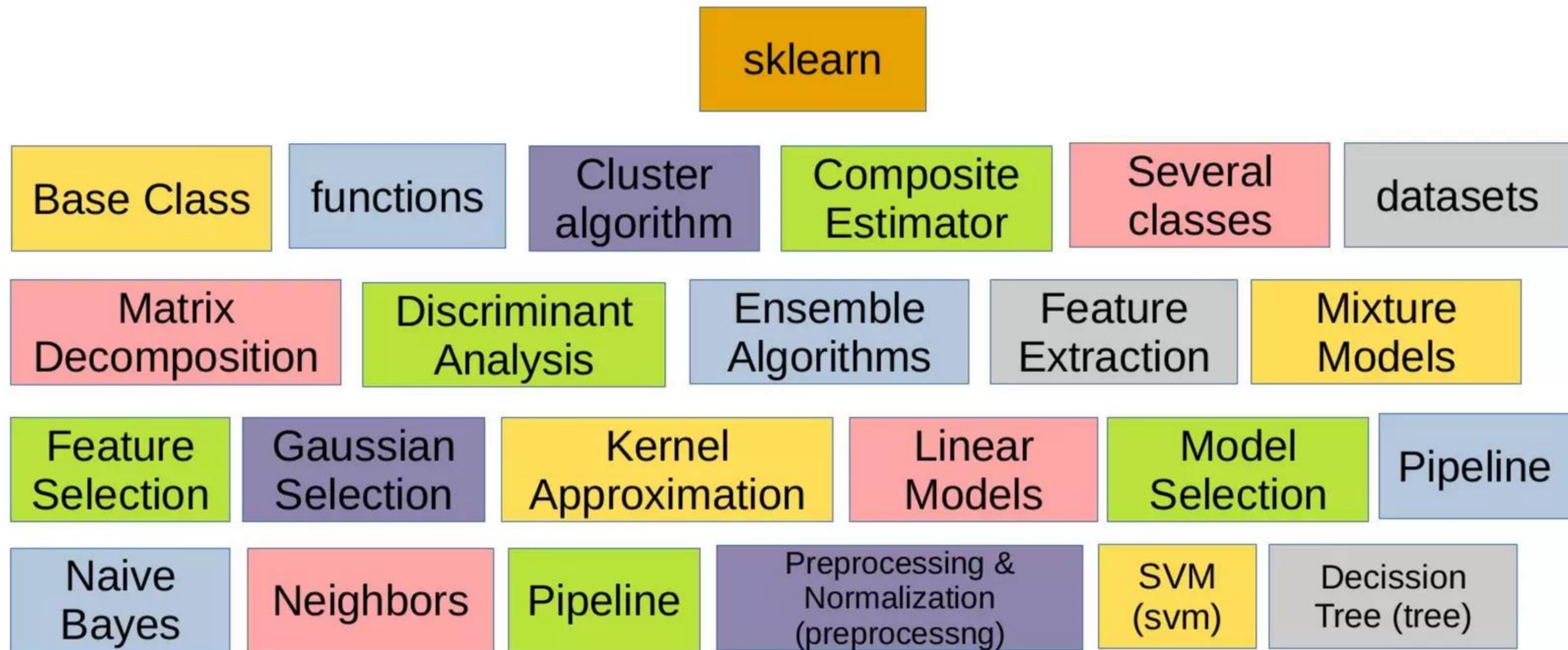
Functions

base.clone(estimator, *, safe)	Constructs a new unfitted estimator with the same parameters.
base.is_classifier(estimator)	Return True if the given estimator is (probably) a classifier.
base.is_regressor(estimator)	Return True if the given estimator is (probably) a regressor.
config_context(**new_config)	Context manager for global scikit-learn configuration
get_config()	Retrieve current values for configuration set by set_config
set_config(assume_finite, working_memory, ...)	Set global scikit-learn configuration
show_versions()	Print useful debugging information"

sklearn.calibration: Probability Calibration



Scikit-Learn





Sample Projects on my github

- Restful API : https://github.com/situkangsayur/staff_app
- Jupyter notebook <https://github.com/situkangsayur/samples-notebook>
- Jupyter notebook for presentation slide : <https://github.com/situkangsayur/python-outfit-slide>
- Rest API machine learning with sklearn : <https://github.com/situkangsayur/baper-ips>
- Opensource library for integrating with consul and vault :
<https://github.com/situkangsayur/python-outfit>



Python Indonesia

<https://t.me/pythonID/217588>

Python Indonesia
24.617 members

Pinned message
WAJIB BACA sampai habis pesan ini, termasuk apa yang ada di tautan-tautan (links). Selamat datang di group Telegram Python Indonesia! Mari bersama...

```
mpikan_data(result)
file "C:\Users\user\Desktop\Gempa_BMW\Gempa_BMW\lokasi\main.py", line 14, in mpikan_data
    print(f'Lokasi: {result["ls"]}, B* {result["B*"]}')
KeyError: 'ls'
```

[Time] exited with code 1 in 0.002 seconds

```
def ekstraksi_data():
    hasil = {}
    hasil['tanggal'] = '24 Agustus 2021'
    hasil['waktu'] = '12:05:52 WIB'
    hasil['magnitudo'] = 4.0
    hasil['lokasi'] = {'ls': 1.46, 'bt': 159.42}
    hasil['pusat'] = 'Pusat gempa berada di darat 18 km Barat Laut Ransiki'
    hasil['dirasakan'] = 'Dirasakan (Shala RW): II-III Manukwari, II-III Ransiki'

    return hasil

def tampilkan_data(result):
    print('Gempa Terakhir Berdasarkan BMKG')
    print(f'Tanggal {result["tanggal"]}')
    print(f'Waktu {result["waktu"]}')
    print(f'Magnitudo {result["magnitudo"]}')
    print(f'Lokasi: {result["ls"]}, B* {result["B*"]}')
    print(f'Pusat {result["pusat"]}')
    print(f'Dirasakan {result["dirasakan"]}')

if __name__ == '__main__':
    print('Aplikasi Utama')
    result = ekstraksi_data()
    tampilkan_data(result)
```

Selamat siang.
Untuk Traceback in <module> sama keyError ini gimana ya solusinya? Terimakasih 12.07

Alif Adwitiya Pratama
Anggy Ferdiawan
Selamat siang. Untuk Traceback in <module> sama keyE...
gak ada key ls di dictionary result. Coba di print dulu dictionary resultnya 12.13

klo ls yang dimaksud adalah ls dalam lokasi harusnya diakses dengan cara result["lokasi"]["ls"] 12.14

ancreatures
#ask
adakah function di python untuk membulatkan angka ke yang terkecil?
contoh misal:
- 1.5 di bulatkan menjadi 1
- 3.7 di bulatkan menjadi 3 13.13

Write a message...



Finantier

Open Finance. Di sini. Hari ini.


Presented by:

Hendri Karisma

Presenter Credential

 hendri@finantier.co

 +62

 [Hendri Karisma](#)

 finantier.co



Finantier is regulated by BI and OJK in Indonesia and SEC in the Philippines