

# Quickstart

This section is mainly intended for developers who are already accustomed to fundamentals of Python, as well as its common ML libraries and frameworks. If you're a beginner in ML Development, we recommend checking the [Tutorials](#) first.

We assume you have installed the giza-datasets library in your preferred environment, if not, check the [installation guide](#) .

1.
2. Import giza-datasets
3.

...

Copy fromgiza\_datasetsimportDatasetsHub,DatasetsLoader

...

Additionally, it might be required to run the following lines. See [DatasetsLoader](#) .

...

Copy importos importcertifi

os.environ['SSL\_CERT\_FILE']=certifi.where()

...

1.
2. Query the datasets using a DatasetsHub object
3.

...

Copy hub=DatasetsHub()

...

With theDatasetsHub() object, we can know query the DatasetsHub to find the perfect dataset for our ML model. See [DatasetsHub](#) for further instructions. Alternatively, you can check [DatasetsHub](#) pages to explore the available datasets from your browser.

Lets use thelist\_tags() function to list all the tags and thenget\_by\_tag() to query all the datasets with the "Yearn-v2" tag.

...

Copy print(hub.list\_tags())

...

[ 'Trade Volume', 'DeFi', 'Yearn-v2','Interest Rates','compound-v2',...]

Yearn-v2 looks interesting, lets search all the datasets that have the "Yearn-v2" tag.

...

Copy datasets=hub.get\_by\_tag("Yearn-v2")

fordatasetindatasets: hub.describe(dataset.name)

...

...

Copy Details for yearn-individual-deposits			Attribute	Value
			Path	gs://datasets-giza/Yearn/Yearn_Individual_Deposits.parquet
Description	Individual Yearn Vault deposits			
Tags	DeFi, Yield, Yearn-v2, Ethereum, Deposits			
Documentation	https://datasets.gizatech.xyz/hub/yearn/individual-vault-deposits			

...

yearn-individual-deposits looks great!

1.
2. Load a dataset using DatasetLoader
3.

...

Copy loader=DatasetsLoader()

...

Having instantiated theDatasetsLoader() , all we need to do is load the dataset using the name we have queried usingDatasetsHub() .

...

Copy df=loader.load("yearn-individual-deposits")

df.head()

...

shape: (5, 7)

evt\_block\_time evt\_block\_number vaults token\_contract\_address token\_symbol token\_decimals value datetime[ns] i64 str str i64 i64 2023-06-07 09:50:35 17427717 "0x3b27f92c0e21... "0xdac17f958d2e... "USDT" 6 14174.301085 2022-08-25 13:53:28 15409462 "0x3b27f92c0e21... "0xdac17f958d2e... "USDT" 6 38.046614 2022-08-25 07:13:02 15407745 "0x3b27f92c0e21... "0xdac17f958d2e... "USDT" 6 4620.369198 2022-11-19 03:41:35 16001443 "0x3b27f92c0e21... "0xdac17f958d2e... "USDT" 6 969.687071 2022-12-30 18:34:11 16299403 "0x3b27f92c0e21... "0xdac17f958d2e... "USDT" 6 56.270566 Keep in mind that giza-datasets uses Polars (and not Pandas) as the underlying DataFrame library.

Perfect, the Dataset is loaded correctly and ready to go! Now we can use our preferred ML Framework and start building.

[Previous Installation](#) [Next Overview](#)

