

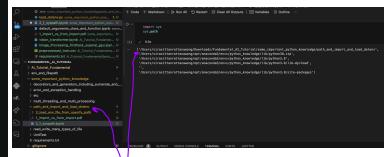
# 1. sys.path

## sys.path in Python

Sys is a built-in Python module that contains parameters specific to the system i.e. it contains variables and methods that interact with the interpreter and are also governed by it.

### sys.path

syspath is a built-in variable within the sys module. It contains a list of directories that the interpreter will search in for the required module.



- **DEFAULT-** By default, the interpreter looks for a module **within the current directory**. To make the interpreter search in some other directory you just simply have to change the current directory. The following example depicts a default path taken by the interpreter:

A screenshot of a Jupyter Notebook cell. The code imports 'sys' and prints 'sys.path'. The output shows the path: '/Users/rohit/PycharmProjects/Python\_Lecture\_1'. A blue arrow points from the text 'current path is ...' to the output. Another blue arrow points from 'current path is ...' to the line 'import sys'. A blue circle highlights the output text. A blue bracket highlights the entire code block.

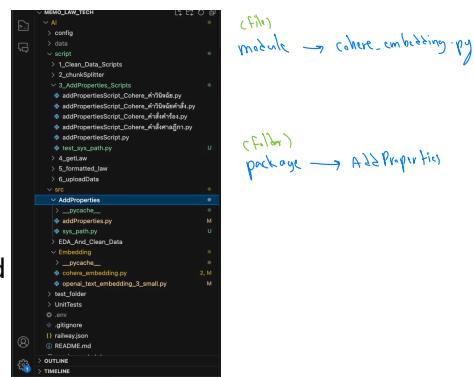
```
import sys
print(sys.path)
```

Output:

```
['/Users/rohit/PycharmProjects/Python_Lecture_1']
```

A module is a Python file that's intended to be imported into scripts or other modules.

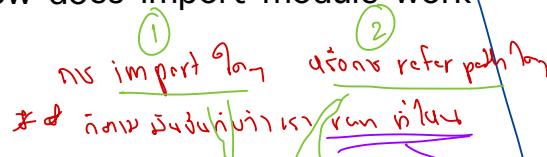
A package is a collection of python files. These files are specifically placed within a folder. Ex, python class files dog.py, cat.py, and rat.py would be placed under folder animal, then folder animal would be the package.



## 2. How to set folder structure and how does import module work

Firstly, we need to understand that every path would relate to how(where) we run the script

I would use these 3 files as an example



module1 (as a script)

```
module1
├── config
│   ├── __init__.py
│   └── config.py
├── data
│   ├── __init__.py
│   └── data.py
└── models
    ├── __init__.py
    └── memo_low.py
```

module2 (in src) (call module1)

```
module2
├── config
│   ├── __init__.py
│   └── config.py
├── data
│   ├── __init__.py
│   └── data.py
└── models
    ├── __init__.py
    └── memo_low.py
```

module3

```
module3
├── config
│   ├── __init__.py
│   └── config.py
├── data
│   ├── __init__.py
│   └── data.py
└── models
    ├── __init__.py
    └── memo_low.py
```

When we run we would run at directory of module1 (scripts)

```
# (memo_low) sirasittamrattanawong@irisits-air ~_AddProperties_Scripts % python test_sys_path
project root is : ../../.
before sys.path is ['~/Users/sirasittamrattanawong/Downloads/memo_low/tech/ai/script/_AddProperties_Scripts', '/Users/sirasittamrattanawong/opt/anaconda3/envs/memo_low/tech/lib/python3.8.zip', '/Users/sirasittamrattanawong/opt/anaconda3/envs/memo_low/tech/lib/python3.8', '/Users/sirasittamrattanawong/opt/anaconda3/envs/memo_low/tech/lib/python3.8/lib-dynload', '/Users/sirasittamrattanawong/opt/anaconda3/envs/memo_low/tech/lib/python3.8/site-packages']
Index 0 of sys.path (current path/dir) is : /Users/sirasittamrattanawong/Downloads/memo_low/tech/ai/script/_AddProperties_Scripts
After sys.path is ['../../', '/Users/sirasittamrattanawong/Downloads/memo_low/tech/ai/script/_AddProperties_Scripts', '/Users/sirasittamrattanawong/opt/anaconda3/envs/memo_low/tech/lib/python3.8.zip', '/Users/sirasittamrattanawong/opt/anaconda3/envs/memo_low/tech/lib/python3.8', '/Users/sirasittamrattanawong/opt/anaconda3/envs/memo_low/tech/lib/python3.8/lib-dynload', '/Users/sirasittamrattanawong/opt/anaconda3/envs/memo_low/tech/lib/python3.8/site-packages']

[0.0197408, 0.0362488, -0.00022957, 0.038223267, -0.000350652, 0.01549255, 0.0002063675, -0.037322996, -0.035951416, -0.02005398, -0.04647236, -0.0026607513, -0.0026607513]
```

```
# (memo_low) sirasittamrattanawong@irisits-air ~_AddProperties_Scripts % python test_sys_path
project root is : ../../.
before sys.path is ['~/Users/sirasittamrattanawong/Downloads/memo_low/tech/ai/script/_AddProperties_Scripts', '/Users/sirasittamrattanawong/opt/anaconda3/envs/memo_low/tech/lib/python3.8.zip', '/Users/sirasittamrattanawong/opt/anaconda3/envs/memo_low/tech/lib/python3.8', '/Users/sirasittamrattanawong/opt/anaconda3/envs/memo_low/tech/lib/python3.8/lib-dynload', '/Users/sirasittamrattanawong/opt/anaconda3/envs/memo_low/tech/lib/python3.8/site-packages']
Index 0 of sys.path (current path/dir) is : /Users/sirasittamrattanawong/Downloads/memo_low/tech/ai/script/_AddProperties_Scripts
After sys.path is ['../../', '/Users/sirasittamrattanawong/Downloads/memo_low/tech/ai/script/_AddProperties_Scripts', '/Users/sirasittamrattanawong/opt/anaconda3/envs/memo_low/tech/lib/python3.8.zip', '/Users/sirasittamrattanawong/opt/anaconda3/envs/memo_low/tech/lib/python3.8', '/Users/sirasittamrattanawong/opt/anaconda3/envs/memo_low/tech/lib/python3.8/lib-dynload', '/Users/sirasittamrattanawong/opt/anaconda3/envs/memo_low/tech/lib/python3.8/site-packages']

[0.0197408, 0.0362488, -0.00022957, 0.038223267, -0.000350652, 0.01549255, 0.0002063675, -0.037322996, -0.035951416, -0.02005398, -0.04647236, -0.0026607513, -0.0026607513]
```

after run in folder AI\_Scripts  
with path injection  
when we run the script  
not path from that module.

now run's add —  
first execution path is code

ກີ່ຈະເຮັດຕິດການຂົບສົກເກມ

1. ໃນ code ອາຍຸ module 1 ໂດຍ run ຮຽນ run ອັນ folder 3-A22 —

ໃຫ້ຈາກລາຍກື sys.path.insert ມີຄວາມ search ອີ່ memo\_low\_tech ດ້ວຍ

2. ; ໃນ module 2 ມີ import module 3 ປໍ່ນາງ

From AZ.svc.Embedding coherence-embedding

ໃຊ້ໄດ້ພູມໄຫວ່າພຽງວ່າ ລາຍລະອຽດ ມີຄວາມ search ອີ່ໂທ່ອນບັນຍາ

memo\_low\_tech ວິທີອຸ່ນຍຸ່ນຕັ້ງ run script  
ດີຫຼັມເນັ້ນ 3-A22 — ໃນ  
ຖຸກຕາມຕຳຫຼາດໃຫຍ່  
path ໃຫຍ່ —>

3. In modules

```
dotenv_path = Path('.../../.env')
config = load_dotenv(dotenv_path=dotenv_path)
```

ກ່າວົກສອງ ພົມທີ່ໃນ run script ດີກົດ

ໃນ folder 3-A22 —

ກ່າວົກສອງ ພົມທີ່ໃນ modules ໄດ້ປັບປຸງກົດ

ஏக்ஸிள்கூ டுமெல் இது வீரப்பதை  
இல்லாத sys.path.insert நை தெய்ய

### 3. Questions

1. டுமெல்

இல்லாத sys.path.insert நை என்று சொல்ல ஒன்று

இம்மேதுக்கூறு இல்லை

நிர்ணயம் கிடைக்கிறோம்

The terminal window displays the following Python code:

```
MEMO_LAW_TECH
├── AI
│   ├── __init__.py
│   ├── config
│   │   └── data
│   └── sort
├── __init__.py
└── 3_AdProperties_Scripts
    ├── __init__.py
    ├── addPropertiesScript_Cohere_AIRIndia.py
    ├── addPropertiesScript_Cohere_AirIndiaFinsys.py
    ├── addPropertiesScript_Cohere_AirIndiaFinsys.py
    └── addPropertiesScript.py

    4_getLaw
    5_formatted_law
    6_uploadData
    test_folder
    UnitTests
    7_main.py
    README.md
    .gitignore
    .env
    .env.local
    .gitkeep
    requirements.txt
    backend
    data
    frontend
```

PROBLEMS: ① OUTPUT DEBUG CONSOLE TERMINAL POINTS

① [memo\_tech] sirasittiratanaengirilts-air memo\_law\_tech \$ python AI/script/3\_AdProperties\_Scripts/test\_sys\_path.py  
Traceback (most recent call last):  
 File "AI/script/3\_AdProperties\_Scripts/test\_sys\_path.py", line 38, in <module>  
 from AI.xi.AdProperties import test  
ModuleNotFoundError: No module named 'AI.xi.AdProperties'  
② [memo\_tech] sirasittiratanaengirilts-air memo\_law\_tech %

2. நை கட்டும் விதம் என்னவே என்ன?

3. ஓரெண்டல் setup.py க்கு

4. ດ້ວຍເຫັນ script ຮັດໃຫຍ່ໄປຂອງ ໄລັດ

ການ

ອີງຕີ ການທີ່ມີຄົນດີ ຮັດໃຫຍ່ໄປ  
ໄຕເຈົ້າໃຫຍ່ໄປ

ພົບເລືອດໂສ ສcript ຮັດໃຫຍ່ໄປ

ເພີ້ມມາ import ພົບເລືອດ input

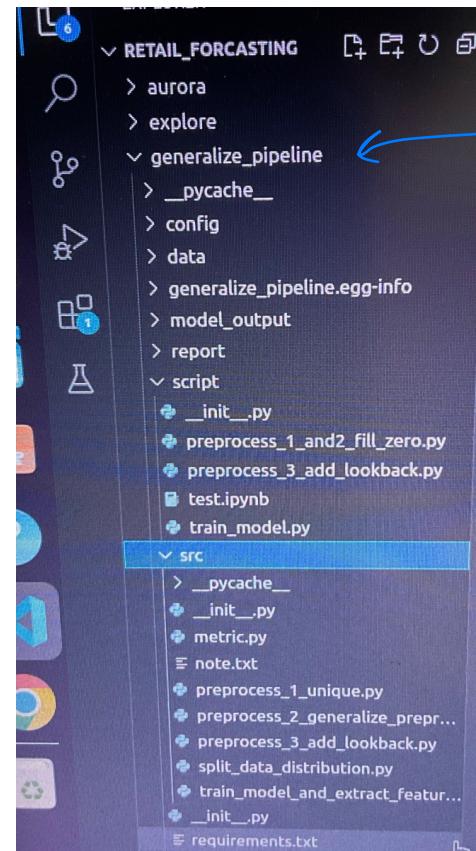
ການ



សារព័ត៌មាន និង project នៃគេង ដែលត្រូវការចាប់ផ្តើម

an internship based on individual's vision

→ impact 17 Mo



Folder 2

monks n' r

import src

میلاد

until now!!

```
> explore  
  > generalize_pipeline  
    > __pycache__  
    > config  
    > generalize_pipeline.egg-info  
      > dependency_links  
      > model_output  
      > report  
    > script  
      > __init__.py  
      > preprocessors_1_and2_fill_zero.py  
      > preprocessors_1_and3_loadcheck.py  
      > test.ipynb  
    > train_model.py  
  > src  
    > __init__.py  
    > __main__.py  
    > metrics.py  
    > notebook.py  
    > preprocessors_1_unique.py  
    > preprocessors_2_and3_loadcheck.py  
    > split_data_distribution.py  
    > train_and_extract_feature...  
    > ...  
  > requirements.txt  
  
1   from generalize_pipeline.src.train_model_and_extract_fea...  
2   from generalize_pipeline.src.preprocess_1_unique import...  
3  
4   def main():  
5       #import preprocessing = UniqueProcessing()  
6       #load path = f'{generalize_pipeline.data_group}/train...  
7       group_df_train_with_lookback_all_data_before_split = ...  
8  
9  
10      average_total_months_back_baseline = 3  
11      save_path_baseline = f'{generalize_pipeline.data_group}/base...  
12      overwritten_baseline = False  
13  
14  
15  
16  
17      train_model_and_extract_feature = TrainModelAndExtractFea...  
18      average_list, y_test_prediction, model_output = train_m...  
19      print(f'mean(average_list), len(y_test_prediction)')  
20  
21  
22      if __name__ == '__main__':  
23          main()  
24
```

# Solve1. Set python anaconda path (from chatgpt first)

How Python search for imported module/package

<https://medium.com/@sachinsoni600517/how-python-search-for-imported-module-package-76cf0da5f690>

How to set environment variables in a conda virtual environment

<https://guillaume-martin.github.io/saving-environment-variables-in-conda.html>

set anaconda python path

<https://www.geeksforgeeks.org/how-to-setup-anaconda-path-to-environment-variable/>

1. Create conda env like before.
2. Conda env list to find path of conda env

```
1 day   ● (retail2) ( .venv ) loolootech@loolootech:~/Downloads/retail_forcasting$ conda env list
2 days
# conda environments:
#
base          /home/loolootech/anaconda3
italasia      /home/loolootech/anaconda3/envs/italasia
retail2        * /home/loolootech/anaconda3/envs/retail2
retail_forcast_pipeline /home/loolootech/anaconda3/envs/retail_forcast_pipeline
```

3. set path → *will reference to the previous step*

2. Navigate to the Activation Directory:

Conda environments have a directory for activation scripts. We will create one if it doesn't exist.

```
sh
mkdir -p /home/loolootech/anaconda3/envs/retail_forcast_pipeline/etc/conda/activate.d
```

3. Create the Activation Script:

Create a new script file named 'env\_vars.sh' in the 'activate.d' directory.

```
sh
vim /home/loolootech/anaconda3/envs/retail_forcast_pipeline/etc/conda/activate.d/env_vars.sh
```

4. Add the Following Content to Set the 'PYTHONPATH':

Press 'i' to enter insert mode, then add the following line:

```
sh
export PYTHONPATH=/home/loolootech/Downloads/retail_forecasting:$PYTHONPATH
```

5. Save and Close the File:

- Press Esc to exit insert mode.
- Type :wq and press Enter to save and quit vim.

3. Create Deactivation Script (Optional but Recommended)

To clean up the environment variable when you deactivate the Conda environment, you can create a deactivation script.

1. Navigate to the Deactivation Directory:

```
sh
mkdir -p /home/loolootech/anaconda3/envs/retail_forcast_pipeline/etc/conda/deactivate.d
```

2. Create the Deactivation Script:

Create a new script file named 'env\_vars.sh' in the 'deactivate.d' directory.

```
sh
vim /home/loolootech/anaconda3/envs/retail_forcast_pipeline/etc/conda/deactivate.d/env_vars.sh
```

3. Add the Following Content to Unset the 'PYTHONPATH':

Press 'i' to enter insert mode, then add the following line:

```
sh
unset PYTHONPATH
```

4. Save and Close the File:

  - Press Esc to exit insert mode.
  - Type :wq and press Enter to save and quit vim.

Running Your Script

Now, whenever you activate your Conda environment, the 'PYTHONPATH' will be set automatically.

1. Activate Your Conda Environment:

```
sh
conda activate retail_forcast_pipeline
```

2. Navigate to Your Project Directory:

```
sh
cd /home/loolootech/Downloads/retail_forecasting
```

3. Run Your Script:

```
sh
python generalize_pipeline/script/preprocess.py
```

*reference*

Now, we need to create scripts that will set our environment variables whenever we activate our virtual environment and unset them when we deactivate. The path to our environment directory is set as \$CONDA\_PREFIX. We cd to that directory and create a /etc/conda/activate.d and a /etc/conda/deactivate.d directory:

```
(my_environment)$ cd $CONDA_PREFIX
(my_environment)$ mkdir -p ./etc/conda/activate.d
(my_environment)$ mkdir -p ./etc/conda/deactivate.d
```

In each of those directories, we create a env\_var.sh script where we'll write the commands to set and unset our variables:

```
(my_environment)$ touch ./etc/conda/activate.d/env_vars.sh
(my_environment)$ touch ./etc/conda/deactivate.d/env_vars.sh
```

or run pip install -r requirements.txt

```
RETAIL_FORCASTING
    > aurora
    > explore
    > generalize_pipeline
        > __pycache__
        > config
        > data
        > generalize_pipeline.egg-info
        > model_output
        > report
    > script
        > __init__.py
        > preprocess_1_and2_fill_zero.py
        > preprocess_3_add_lookback.py
        > test.ipynb
        > train_model.py
    > src
        > __pycache__
        > __init__.py
        > metric.py
        > note.txt
        > preprocess_1_unique.py
        > preprocess_2_generalize_prep...
        > preprocess_3_add_lookback.py
        > split_data_distribution.py
        > train_model_and_extract_featur...
        > __init__.py
        > requirements.txt
    > generalize_pipeline.egg-info
    > italiasa
```

```
generalize_pipeline > script > train_model.py > main
1   from generalize_pipeline.src.train_model_and_extract_feature import TrainModelAndExtractFeature
2   from generalize_pipeline.src.preprocess_1_unique import UniqueProcessing
3
4   def main():
5       unique_processing = UniqueProcessing()
6       read_path = "./generalize_pipeline/data/group_df_train_with_lookback_all_data_before_split.csv"
7       group_df_train_with_lookback_all_data_before_split = unique_processing.read_file(read_path)
8
9
10      average_total_months_back_baseline0 = 3
11      save_path_baseline0 = "./generalize_pipeline/data/baseline_0_{average_total_months_back_baseline0}_months_
12      overwritten_baseline0 = False
13
14
15      train_model_and_extract_feature = TrainModelAndExtractFeature()
16      average_list, y_test_prediction, model_output = train_model_and_extract_feature.runall(group_df_train_with_lo
17
18      print(len(average_list), len(y_test_prediction))
19
20
21
22
23  if __name__ == "__main__":
24      main()
```

:, run script

ஒவ்வொரு போக்குவரத்து  
இன் சிரமங்களைப் போக்குவரத்து

கீழ் run command பதில் கிடைத்துவது

```
(retail2) (.venv) loolootech@loolootech:~/Downloads/retail_forcasting$ pwd
/home/loolootech/Downloads/retail_forcasting
(retail2) (.venv) loolootech@loolootech:~/Downloads/retail_forcasting$ python generalize_pipeline/script/train_model.py
```

Important key

1. Set anaconda python path to retail\_store

2.1 when we want to run anything we would run at retail\_store folder ex. python generalize\_pipeline/script/train\_model.py

2.2 same as before path would relate to where we run the script.

2.3 we set anaconda python path and run the script at retail\_store folder when we import we would think that that is our folder path ex. from generalize\_pipeline.src.train\_model\_and\_extract\_feature import TrainModelAndExtractFeature

```
RETAIL_FORCASTING
  aurora
  explore
  generalize_pipeline
    > .pycache_
    > config
    > data
    > generalize_pipeline.egg-info
    > model_output
    > report
  script
    > __init__.py
    > preprocess_1_and2_fill_zero.py
    > preprocess_3_add_lookback.py
    > test.ipynb
  train_model.py
  src
    > .pycache_
    > __init__.py
    > metric.py
  note.txt
  preprocess_1_unique.py
  preprocess_2_generalize_prep...
  preprocess_3_add_lookback.py
  split_data_distribution.py
  train_model_and_extract_featu...
  __init__.py
  requirements.txt
  generalize_pipeline.egg-info
  setup.py

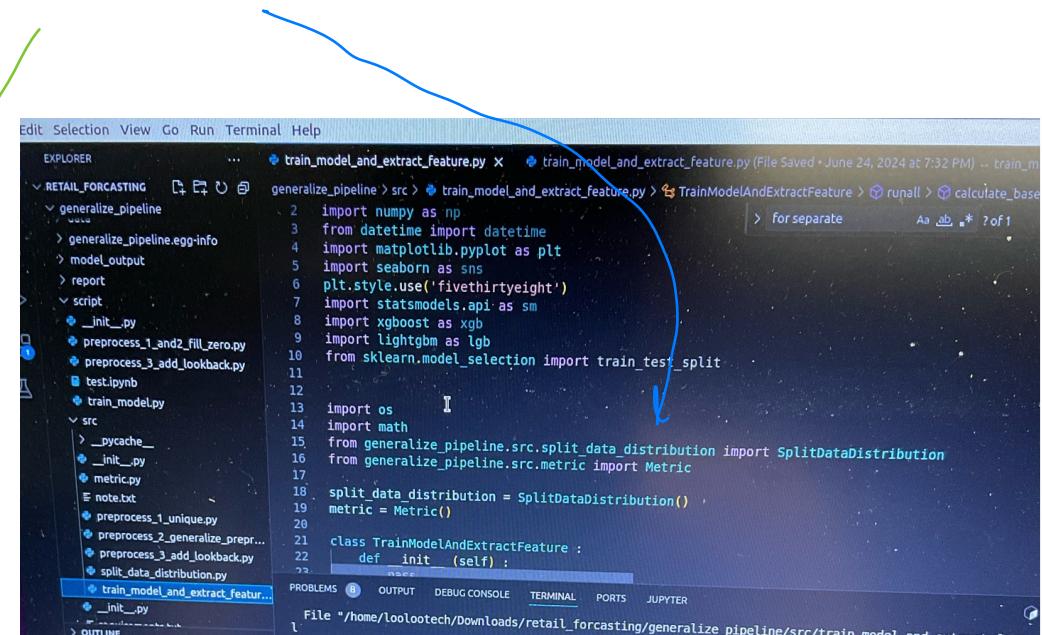
generalize_pipeline > script > train_model.py > main
main.py

1 from generalize_pipeline.src.train_model_and_extract_feature import TrainModelAndExtractFeature
2
3 def main():
4     unique_processing = UniqueProcessing()
5     read_path = ".generalize_pipeline/data/group_df_train_with_lookback_all_data_before_split.csv"
6     group_df_train_with_lookback_all_data_before_split = unique_processing.read_file(read_path)
7
8     average_total_months_back_baseline0 = 3
9     save_path_baseline0 = "./generalize_pipeline/data/baseline_0_{average_total_months_back_baseline0}_months"
10    overwritten_baseline0 = False
11
12    train_model_and_extract_feature = TrainModelAndExtractFeature()
13    average_list, y_test_prediction, model_output = train_model_and_extract_feature.runall(group_df_train_with_lo
14
15
16
17
18
19
20
21
22
23 if __name__ == "__main__":
24     main()
```

```
RETAIL_FORCASTING
  aurora
  explore
  generalize_pipeline
    > .pycache_
    > config
    > data
    > generalize_pipeline.egg-info
    > model_output
    > report
  script
    > __init__.py
    > preprocess_1_and2_fill_zero.py
    > preprocess_3_add_lookback.py
    > test.ipynb
  train_model.py
  src
    > .pycache_
    > __init__.py
    > metric.py
  note.txt
  preprocess_1_unique.py
  preprocess_2_generalize_prep...
  preprocess_3_add_lookback.py
  split_data_distribution.py
  train_model_and_extract_featu...
  __init__.py
  requirements.txt
  generalize_pipeline.egg-info
  setup.py

generalize_pipeline > script > preprocess_3_add_lookback.py > main
main.py

1 from generalize_pipeline.src.preprocess_3_add_lookback import LookBack
2 from generalize_pipeline.src.preprocess_1_unique import UniqueProcessing
3
4 def main():
5     lookback = LookBack()
6     unique_processing = UniqueProcessing()
7     read_path = ".generalize_pipeline/data/encoded_df_fill_zero.csv"
8     encoded_df_fill_zero = unique_processing.read_file(read_path)
9
10    # this should be in config
11    save_path = ".generalize_pipeline/data/group_df_train_with_lookback_all_data_before_split.csv"
12    group_df_train_with_lookback_all_data_before_split = lookback.add_look_back(encoded_df_fill_zero, save
13
14
15
16
17
18
19 if __name__ == "__main__":
20     main()
```



file path

Initialisation → import path

Iteration → file path

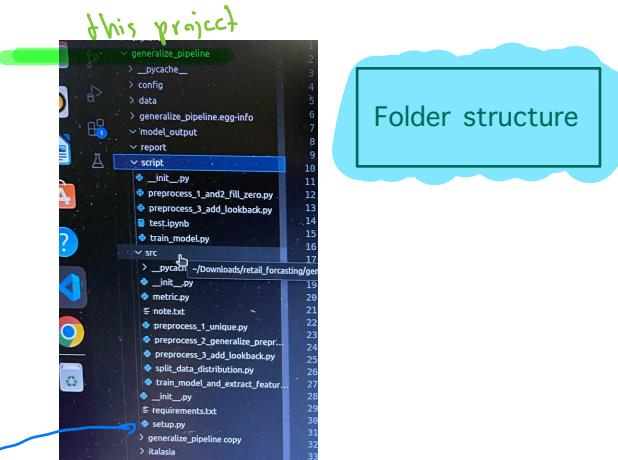
```
● (retail2) (.venv) loolootech@loolootech:~/Downloads/retail_forcasting$ pwd  
/home/loolootech/Downloads/retail_forcasting  
○ (retail2) (.venv) loolootech@loolootech:~/Downloads/retail_forcasting$ python generalize_pipeline/script/train_model.py
```



## Solve2 setup.py

setup.py file python tutorial

<https://xebia.com/blog/a-practical-guide-to-using-setup-py/>



This project

Folder structure

```
1 from setuptools import setup, find_packages
2
3 setup(
4     name='generalize_pipeline',
5     version='0.1',
6     packages=find_packages(), # Finds all packages under the current directory
7     python_requires='>=3.10', # Specify the required Python version
8     install_requires=[
9         'pandas==2.2.2',
10        'pyarrow==16.1.0',
11        'fastparquet==2024.5.0',
12        'matplotlib==3.9.0',
13        'plotly==5.22.0',
14        'nbformat==5.10.4',
15        'seaborn==0.13.2',
16        'statsmodels==0.14.2',
17        'xgboost==2.0.3',
18        'lightgbm==4.3.0',
19        'pyproject-toml==0.8.10',
20        'scikit-learn==1.5.0',
21    ],
22)
23
```

0. crack contd env.

1. Create setup.py file at root of the project (in generalize\_pipeline) (1 step inside compared to before)
2. pip install -e . at the folder that consists of setup.py
3. Now path would be generalize\_pipeline not retail \_store so you need to change import code and path compared to above

In scripts folder example (one step inside compare to solve)

preprocess\_1\_and2\_fill\_zero.py

```
from src.preprocess_1.unique import UniqueProcessing
from src.preprocess_2.generalize_preprocess_fill_zero import GeneralizePreprocessingFillZero

def main():
    unique_processing = UniqueProcessing()
    generalize_processing_fill_zero = GeneralizePreprocessingFillZero()

    file_path = "./data/ITALASIA SHOP DATA 2023.parquet" # this should be in config
    original_df = unique_processing.read_file(file_path)
    rename_dict = [
        {"ชื่อคอลัมน์": "date",
         "ภาษาไทย": "price",
         "ภาษาไทย": "volume",
         "ภาษาไทย": "total_price",
         "ภาษาไทย": "total_revenue",
         "ภาษาไทย": "unit",
         "ภาษาไทย": "outlet",
         "ภาษาไทย": "outlet_code",
         "ภาษาไทย": "discount",
         "ภาษาไทย": "is_return",
         "SKU": "SKU ID",
         "SKU Name": "SKU Name", # ต้องบันทึกชื่อบริษัท (brand)
         "Product Group": "Brand", }
    ]
    df = unique_processing.unique_preprocess(original_df, rename_dict)

    # should be in config
    selected_cols = ["outlet_code", "Brand", "Product Type", "SKU ID", "unit", "volume", "date"]
    categorical_cols = ['SKU ID', 'Brand', 'Product Type', 'unit', 'outlet_code']
    feature_cols = [col for col in selected_cols if col not in ['outlet_code', 'SKU_ID', 'volume', 'date']]
    cols_to_be_grouped_by_first = ['Brand', 'Product Type', 'unit']
    cols_to_be_grouped_by_sum = ['volume']
    save_path = "./data/encoded_df_fill_zero.csv"

    generalize_processing_fill_zero.run_all(df, selected_cols, categorical_cols, feature_cols, cols_to_be_grouped_by_sum)
```

che requirements.txt setup.py  
tech@loolotech:~/Downloads/retail\_forcasting/generalize\_pipeline\$ python script/preprocess\_1\_and2\_fill\_zero.py

run at

Config file

# Method1

## 1. Write yaml file and load

```
RETAIL_FORECASTING
  > aurora
  > explore
  > generalize_pipeline
    > __pycache__
    > config
      ! preprocess1_config.yaml
      ! preprocess2_config.yaml
    > data
    > generalize_pipeline.egg-info
    > mode_output
    > outputs
    > report
    > report.txt
    > script
      > __init__.py
      > preprocess_1_unique_script.py
      > preprocess_2_fill_zero.py
      > preprocess_3_add_lookback.py
      test.ipynb
      train_model.py
    > src
    > temp
    > report.txt
    > __init__.py
    > .gitignore
  README.md
```

```
generalize_pipeline > config > ! preprocess2_config.yaml
1 # generalize_pipeline/config/preprocess2_config.yaml
2
3 group_outlet_code: true
4 file_path: "./data/preprocessed_data.csv"
5 save_path: "./data/encoded_df_fill_zero.csv"
6 selected_cols:
7   - SKU_ID
8   - year
9   - month
10  - day
11  - target
12  - Brand
13  - Product Type
14  - unit
15 categorical_cols:
16   - SKU_ID
17   - Brand
18   - Product Type
19   - unit
20 feature_cols:
21   - Brand
22   - Product Type
23   - unit
24 cols_to_be_grouped_by_first:
25   - Brand
26   - Product Type
27   - unit
28 cols_to_be_grouped_by_sum:
29   - target
```

relative to where you run  
(from setup.py file)

search command : yaml file tutorial medium

link : <https://medium.com/buildpiper/all-you-need-to-know-about-yaml-files-8fa319b1f26f>

search command : how to load yaml file in python

link : <https://python.land/data-processing/python-yaml>

```
RETAIL_FORECASTING
  > aurora
  > explore
  > generalize_pipeline
    > __pycache__
    > config
      ! preprocess1_config.yaml
      ! preprocess2_config.yaml
    > data
    > generalize_pipeline.egg-info
    > mode_output
    > outputs
    > report
    > report.txt
    > script
      > __init__.py
      > preprocess_1_unique_script.py
      > preprocess_2_fill_zero.py
      > preprocess_3_add_lookback.py
      test.ipynb
      train_model.py
    > src
    > temp
    > report.txt
    > __init__.py
    > .gitignore
  README.md
```

```
generalize_pipeline > config > ! preprocess2_config.yaml
1 import yaml
2 from src.preprocessing_1.unique import UniqueProcessing
3 from src.preprocessing_2.fill_zero import GeneralizePreprocessingFillZero
4
5 def load_config(config_path):
6     with open(config_path, 'r') as file:
7         config = yaml.safe_load(file)
8     return config
9
10 def main():
11     # Main function to preprocess and transform data.
12
13     # This function reads a parquet file, renames its columns, and applies preprocessing steps.
14     # Finally, it runs a general preprocessing step to fill missing values and saves the transformed
15     # data to a new parquet file.
16
17     Arguments
18     None
19
20     Returns
21     None
22
23     # Load configuration
24     config = load_config('./config/preprocess2_config.yaml')
25
26     unique_processing = UniqueProcessing()
27     generalize_processing_fill_zero = GeneralizePreprocessingFillZero()
28
29     # Get configuration values
30     group_outlet_code = config['group_outlet_code']
31     file_path = config['file_path']
32
33     selected_cols = config['selected_cols']
34     categorical_cols = config['categorical_cols']
35     feature_cols = config['feature_cols']
36
37     cols_to_be_grouped_by_first = config['cols_to_be_grouped_by_first']
38     cols_to_be_grouped_by_sum = config['cols_to_be_grouped_by_sum']
39
40     # Add 'outlet_code' if group_outlet_code is True
41     if group_outlet_code:
42         required_cols.append('outlet_code')
43
44     required_cols.append('year')
45     required_cols.append('month')
46     required_cols.append('day')
47
48     if group_outlet_code:
49         required_cols.append('outlet_code')
50
51     # Read the original data file
52     df = unique_processing.read_file(file_path)
53     generalize_processing_fill_zero.run_all(df,
54                                             required_cols,
55                                             selected_cols,
56                                             categorical_cols,
57                                             feature_cols,
```

how to load

load and use

# Method2 hydra (a little different from openthaigpt)

Link : <https://medium.com/@jh.baek.sd/mastering-configuration-management-in-python-with-hydra-with-omegacnf-a-comprehensive-guide-5cf1d38e01f7>

```
RETAIL_FORECASTING
aurora
explore
generalize_pipeline
> generalize_pipeline
  config
    preprocess1_config.yaml
    preprocess2_config.yaml
  data
  generalize_pipeline.egg-info
  model_output
  outputs
  report
    report.txt
  script
    __init__.py
    preprocess_1_unique_script.py
    preprocess_2_fill_zero.py
    preprocess_3_add_lookback.py
    test.ipynb
    train_model.py
  src
  temp
    report.txt
  __init__.py
  .gitignore
  README.md
  reference.txt
  requirements.txt
  setup.py
  > generalize_pipeline_copy
  italias

> OUTLINE
< TIMELINE preprocess_2_fill_zero.py
  o File Saved now
  o Undo / Redo 1 min
  o File Saved 20 mins
  o File Saved 23 mins
  o File Saved 30 mins
  o File Saved 31 mins
  o File Saved 32 mins
  o File Saved 33 mins
  o File Saved 37 mins
  o File Saved 46 mins
  o File Saved 2 days

@hydra.main(config_path='../../config', config_name='preprocess2_config', version_base=None)
def main(cfg: DictConfig) -> None:
    """
    Main function to preprocess and transform data.

    This function reads a parquet file, renames its columns, and applies preprocessing steps.
    Finally, it runs a general preprocessing step to fill missing values and saves the transformed data.

    Arguments
    -----
    None

    Returns
    -----
    None
    """

    unique_processing = UniqueProcessing()
    generalize_processing_fill_zero = GeneralizePreprocessingFillZero()

    # Get configuration values
    group_outlet_code = config['group_outlet_code']
    group_outlet_code = cfg.group_outlet_code
    file_path = cfg.file_path
    save_path = cfg.save_path
    selected_cols = cfg.selected_cols
    categorical_cols = cfg.categorical_cols
    feature_cols = cfg.feature_cols
    cols_to_be_grouped_by_first = cfg.cols_to_be_grouped_by_first
    cols_to_be_grouped_by_sum = cfg.cols_to_be_grouped_by_sum

    # Add 'outlet_code' if group_outlet_code is True
    required_cols = ['SKU_ID', 'year', 'month', 'day', 'target']
    if group_outlet_code:
        required_cols.append("outlet_code")
        selected_cols.append("outlet_code")
        categorical_cols.append("outlet_code")

    group_level = ['SKU_ID']
    if group_outlet_code:
        group_level.append("outlet_code")

    # Read the original data file
    df = unique_processing.read_file(file_path)
    generalize_processing_fill_zero.run_all(
        df,
        selected_cols,
        categorical_cols,
        feature_cols,
        cols_to_be_grouped_by_first)
```

\*\*\* So, generally, path should always be used relate to where we run the scripts