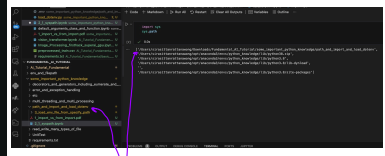
1. sys.path

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.

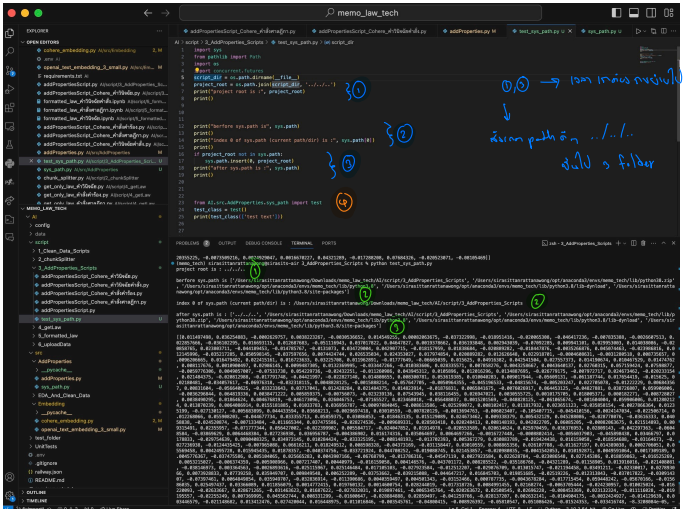
sys.path

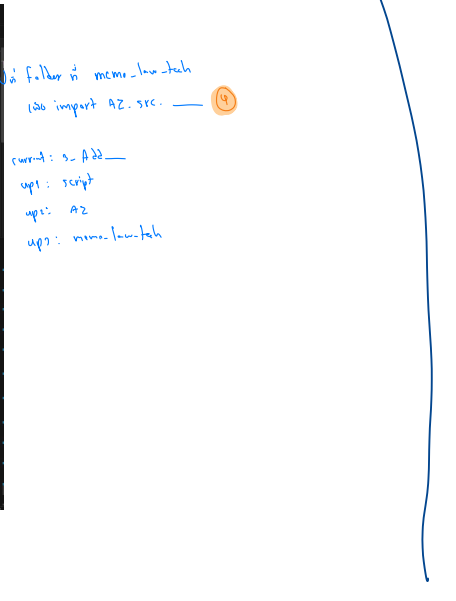
sys.path is a built-in variable within the sys module. It contains a list of directories that the interpreter will search in for the required module.



current dir

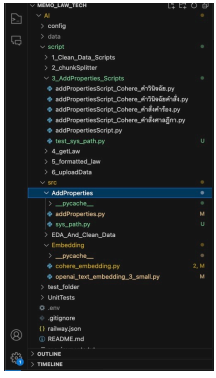
- **DEFAULT-** By default, the interpreter looks for a module within the current directory. To make the interpreter search in some other directory you just simply have to change the current directory. The following example depicts a default path taken by the interpreter:





A 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.



(file)
module → cohere_embedding.py

(folder)
package → AddProperties

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

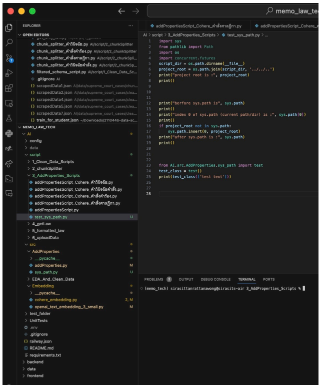
Firstly, we need to understand that every path would relate to how(where) we run the script

I would use these 3 files as an example

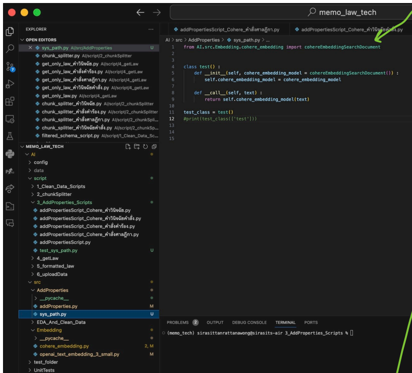
(1) การ import ไฟล์ (2) หรือการ refer path ไฟล์

** ตาม มันขึ้นกับว่าเรา run ที่ไหน

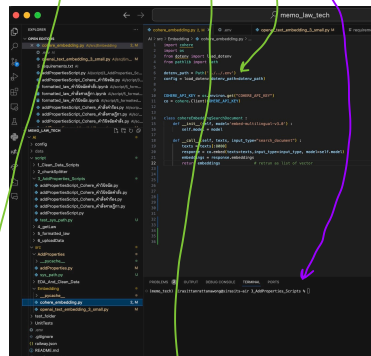
module 1 (as a script)



module2 (in src) (call module3)



module 3



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

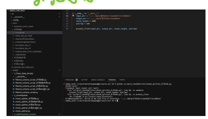
```
(memo_tech) sirasittanrattanawong@sirasits-air 3_AddProperties_Scripts % python test_sys_path.py
project root is : ../../.

before sys.path is ['/Users/sirasittanrattanawong/Downloads/memo_law_tech/AI/script/3_AddProperties_Scripts', '/Users/sirasittanrattanawong/opt/anaconda3/envs/memo_tech/lib/python38.zip', '/Users/sirasittanrattanawong/opt/anaconda3/envs/memo_tech/lib/python3.8', '/Users/sirasittanrattanawong/opt/anaconda3/envs/memo_tech/lib/python3.8/lib-dynload', '/Users/sirasittanrattanawong/opt/anaconda3/envs/memo_tech/lib/python3.8/site-packages']

index 0 of sys.path (current path/dir) is : /Users/sirasittanrattanawong/Downloads/memo_law_tech/AI/script/3_AddProperties_Scripts

after sys.path is : ['../../..', '/Users/sirasittanrattanawong/Downloads/memo_law_tech/AI/script/3_AddProperties_Scripts', '/Users/sirasittanrattanawong/opt/anaconda3/envs/memo_tech/lib/python38.zip', '/Users/sirasittanrattanawong/opt/anaconda3/envs/memo_tech/lib/python3.8', '/Users/sirasittanrattanawong/opt/anaconda3/envs/memo_tech/lib/python3.8/lib-dynload', '/Users/sirasittanrattanawong/opt/anaconda3/envs/memo_tech/lib/python3.8/site-packages']
[[0.011497498, 0.036254883, -0.0002629757, 0.0038223267, -0.0030536652, 0.014549255, 0.00082063675, -0.037322998, -0.018951416, -0.020065308, -0.046417236, -0.087835388, -0.0026607513, 0.
```

ตัวอย่าง



ถ้าเรา run ที่ folder AI มันจะ error
เพราะ path มันเริ่มจาก
when we run the script
not path from that module.

เราต้อง run ที่ 3_Add___
ก็จะสัมพันธ์กับ path ใน code

ตัวอย่าง

มี 3 เรื่องที่ต้องการจะให้สังเกต

1. จาก code ใน module 1 เวลา run เราจะ run ที่ folder 3_Add___

แต่จากการใช้ sys.path.insert มันจะ search ที่ memo_law_tech ด้วย

2. ∴ ใน module2 การ import module 3 เข้ามา

from AZ.src.Embedding.cohere_embedding

เราหา มันเจอเพราะว่า เหมือน มัน search จาก ที่เราอยู่

memo_law_tech

* แต่ที่อยู่ของการ run script
ก็ยังเป็น 3_Add___ นะ
ดูได้จาก ตัวอย่าง เรื่อง
path ในข้อ 3

3. ใน module3

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

สังเกตุว่า_ นับจาก ที่เรา run script ก็คือ

ที่ folder 3_Add___

ไม่ใช่ ตาม path ของ module3 แต่บังเอิญกรณีนี้มันตรงกัน

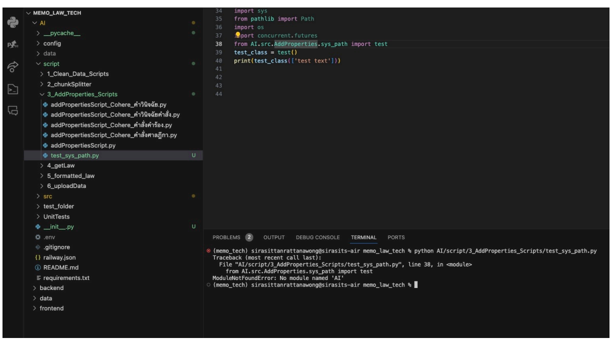
* อย่างไรก็ตาม ใน module1 เป็น bad practice เราไม่ควรใช้ sys.path.insert เวลา deploy

3. Questions

1. ใน module1

เราไม่ใช้ sys.path.insert แล้ว เราไป run ที่ memo_law_tech แทน

ทำไมมันยังหาไม่เจอ



2. แล้ว structure ของ มันควรเป็นยังไงกันแน่

3. อะไรคือ setup.py file

4. ถ้ามีหลายๆ script ควรเขียนไปในชุด แล้ว run

บอกว่า run ที่บรรทัดใด ควรเขียน code

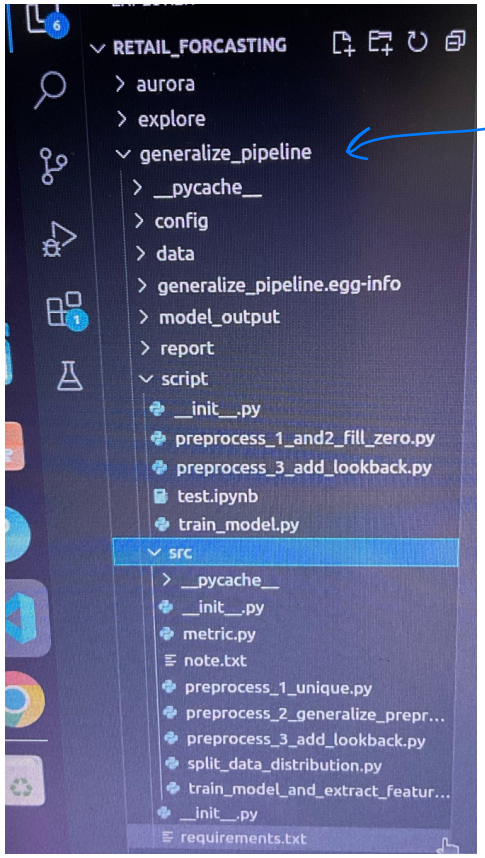
ให้เป็นแบบไหน

จุดที่เราบอกว่า script ควรอยู่บนสุด

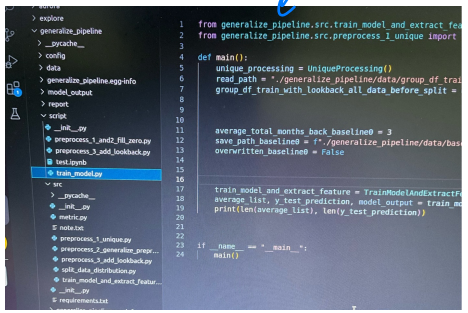
เพราะเวลา import มันก็จะ import ได้ทั้ง

ได้หมด

ปัญหาที่เจอตอนทำ project ตอนแรก เข้าใจดีในระดับนึงตอน
ทำ internship ในโลก เข้าใจแล้วว่าทำไมก่อนหน้านี้
ถึง import ไม่ได้



folder นี้
ตอนแรกก็
import src
ไม่ได้!!



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

set anaconda python path

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

How to set environment variables in a conda virtual environment

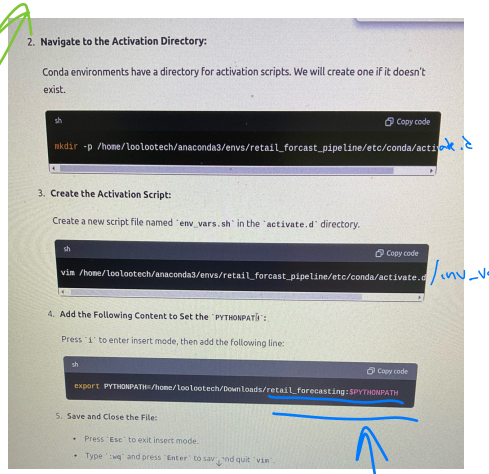
https://guillaume-martin.github.io/saving-environment-variables-in-conda.html

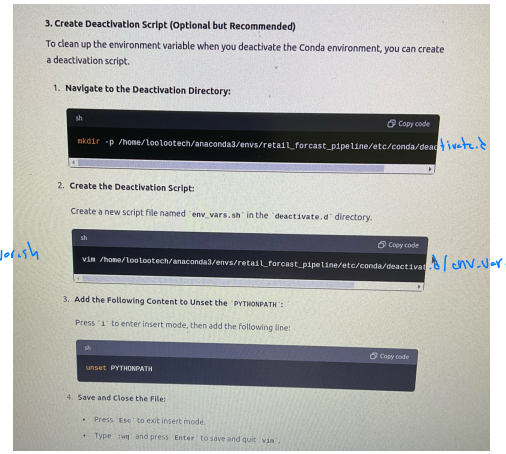
---

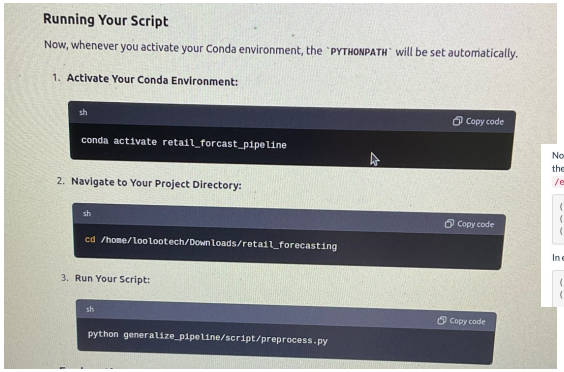
1. Create conda env like before.
2. Conda env list to find path of conda env

```
1 day   (retail2) (.venv) loolootech@loolootech:~/Downloads/retail_forcasting$ conda env list
        # conda environments:
        #
2 days  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 → ขั้นตอนการตั้งค่าตัวแปรสภาพแวดล้อมใน VS Code reference ได้







ex. reference

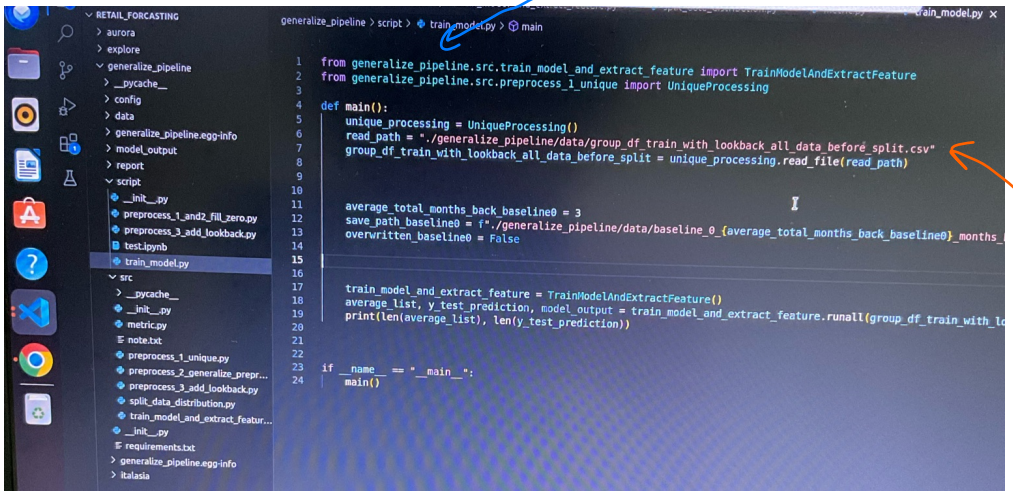
Now, we need to create scripts that will set our environment variables whenever we activate our virtual environment and unset them when we deactivate. The path to our environment directory is set as `$CONDA_PREFIX`. We `cd` to that directory and create a `/etc/conda/activate.d` and a `/etc/conda/deactivate.d`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

* ตรงนี้เวลาจะ import อะไรมันก็เลยมองว่าอยู่ใน conda env อยู่ที่ folder นี้

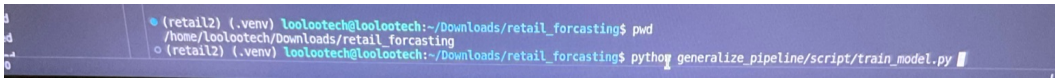


∴ ตรงนี้ก็จะนำเข้าได้เลยแล้ว

อันนี้ก็เหมือนเดิม ยังอยู่กับ

เรา run script ที่ไหน

* เวลา run อะไรก็ต้องดู path นี้



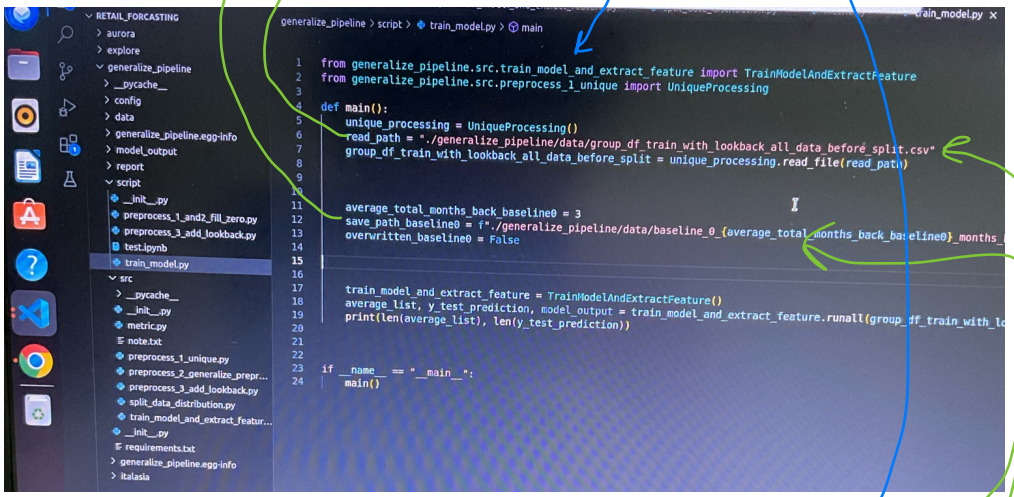
Important key

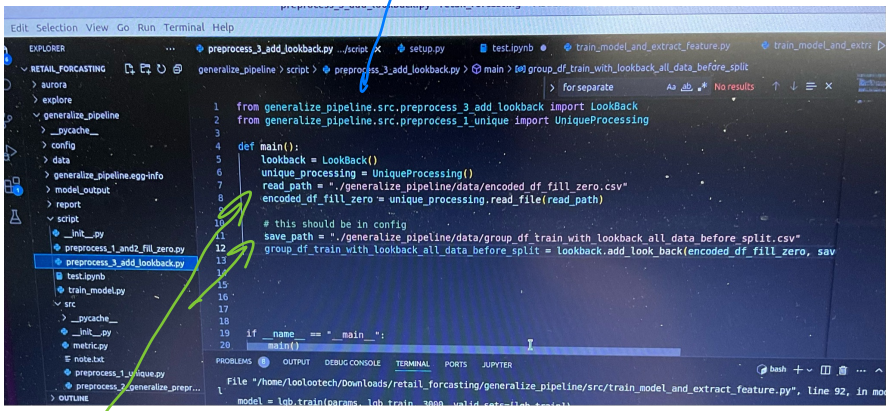
1. Set anaconda python path to retail_store

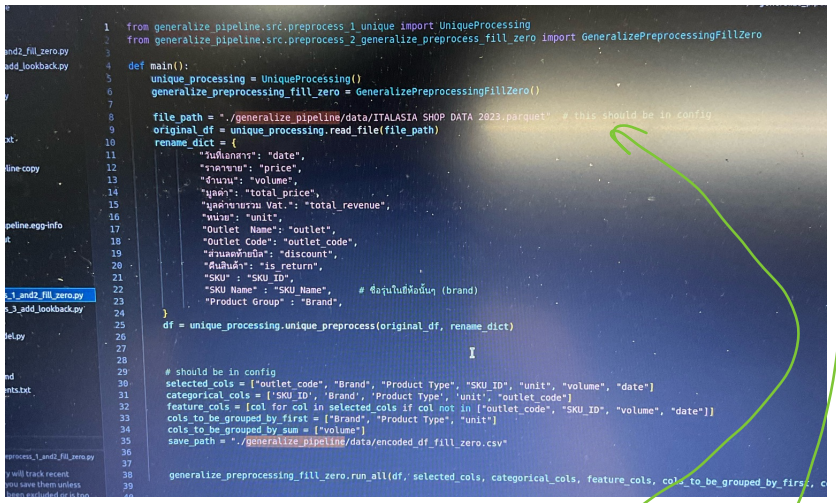
2.1 when we want to run anything we would run at retail_store folder ex. python generalize_pipeline/script/train_model.py

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







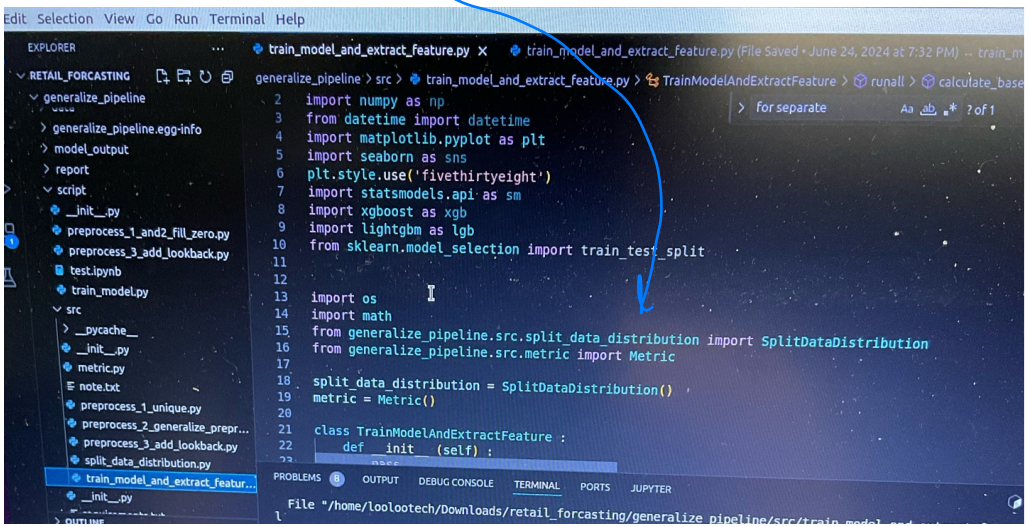
```python
from generalize_pipeline.src.preprocess_1_unique import UniqueProcessing
from generalize_pipeline.src.preprocess_2_generalize_preprocess_fill_zero import GeneralizePreprocessingFillZero

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

    file_path = "./generalize_pipeline/data/ITALASIA SHOP DATA 2023.parquet"  # this should be in config
    original_df = unique_processing.read_file(file_path)
    rename_dict = {
            "วันที่เอกสาร": "date",
            "ราคาขาย": "price",
            "จำนวน": "volume",
            "มูลค่า": "total_price",
            "มูลค่าขายรวม Vat.": "total_revenue",
            "หน่วย": "unit",
            "Outlet  Name": "outlet",
            "Outlet Code": "outlet_code",
            "ส่วนลดท้ายบิล": "discount",
            "คืนสินค้า": "is_return",
            "SKU" : "SKU_ID",
            "SKU Name" : "SKU_Name",     # ชื่อรุ่นในยี่ห้อนั้นๆ (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 = "./generalize_pipeline/data/encoded_df_fill_zero.csv"


    generalize_preprocessing_fill_zero.run_all(df, selected_cols, categorical_cols, feature_cols, cols_to_be_grouped_by_first, c
```



```
Edit  Selection  View  Go  Run  Terminal  Help

EXPLORER                                  train_model_and_extract_feature.py ×   train_model_and_extract_feature.py (File Saved • June 24, 2024 at 7:32 PM)
RETAIL_FORCASTING                         generalize_pipeline > src > train_model_and_extract_feature.py > TrainModelAndExtractFeature > runall > calculate_base
generalize_pipeline                       for separate
  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_prepr...
    preprocess_3_add_lookback.py
    split_data_distribution.py
    train_model_and_extract_featur...
    __init__.py

import numpy as np
from datetime import datetime
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('fivethirtyeight')
import statsmodels.api as sm
import xgboost as xgb
import lightgbm as lgb
from sklearn.model_selection import train_test_split


import os
import math
from generalize_pipeline.src.split_data_distribution import SplitDataDistribution
from generalize_pipeline.src.metric import Metric

split_data_distribution = SplitDataDistribution()
metric = Metric()

class TrainModelAndExtractFeature :
    def __init__(self) :

PROBLEMS 8   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   JUPYTER
File "/home/loolootech/Downloads/retail_forcasting/generalize_pipeline/src/train_model_and_extract_feature.py", line 1
```

file path

เหมือนกัน → import path

เหมือนกัน → 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
```

run at

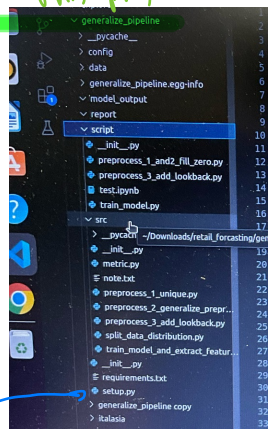
Solve2 setup.py

setup.py file python tutorial

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

อันไหนเป็น folder แล้วจะเรียกไฟล์ในนั้นต้องใส่ __init__.py (แค่ไฟล์เปล่านี่แหละ) ด้วย

this project



Folder structure

setup.py

```python
from setuptools import setup, find_packages

setup(
    name='generalize_pipeline',
    version='0.1',
    packages=find_packages(),  # Finds all packages under the current directory
    python_requires='>=3.10',  # Specify the required Python version
    install_requires=[
        'pandas==2.2.2',
        'pyarrow==16.1.0',
        'fastparquet==2024.5.0',
        'matplotlib==3.9.0',
        'plotly==5.22.0',
        'nbformat==5.10.4',
        'seaborn==0.13.2',
        'statsmodels==0.14.2',
        'xgboost==2.0.3',
        'lightgbm==4.3.0',
        'pyproject==1.3.1',
        'pyproject-toml==0.0.10',
        'scikit-learn==1.5.0',
    ],
)
```

0. crack conda env.

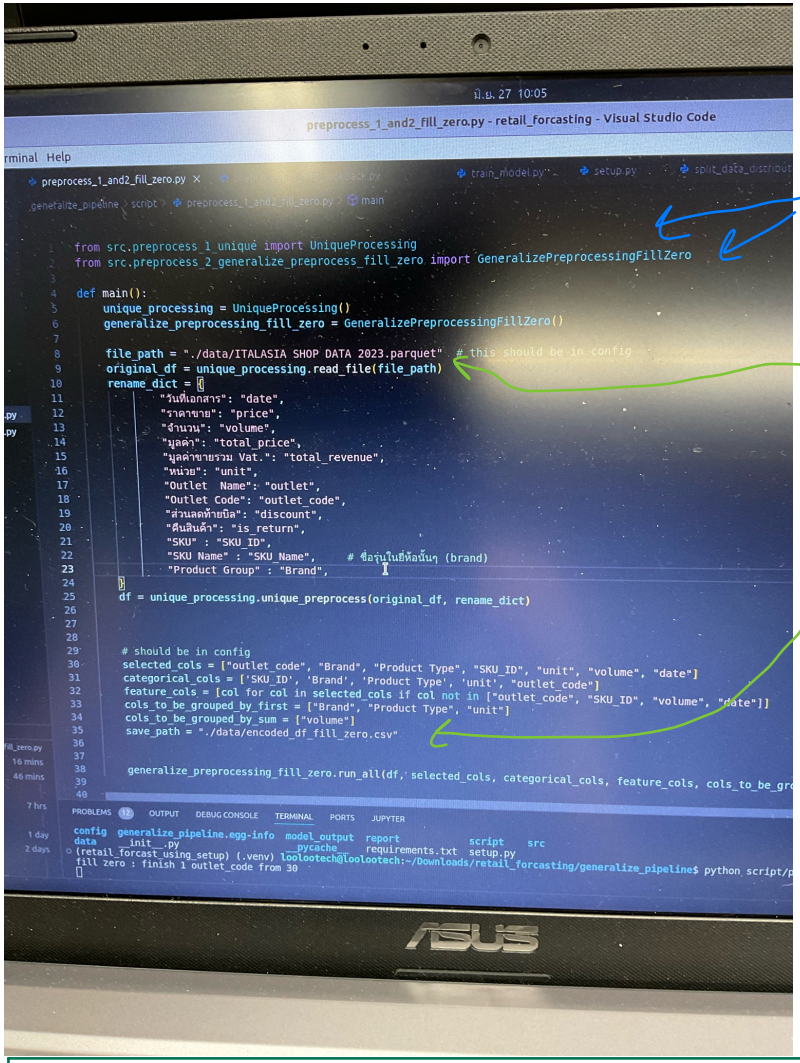
1. Create setup.py file at root of the project (in generalize_pipeline) (1 step inside compared to before)

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




```
che__    requirements.txt  setup.py
ech@loolootech:~/Downloads/retail_forcasting/generalize_pipeline$ python script/preprocess_1_and2_fill_zero.py
```

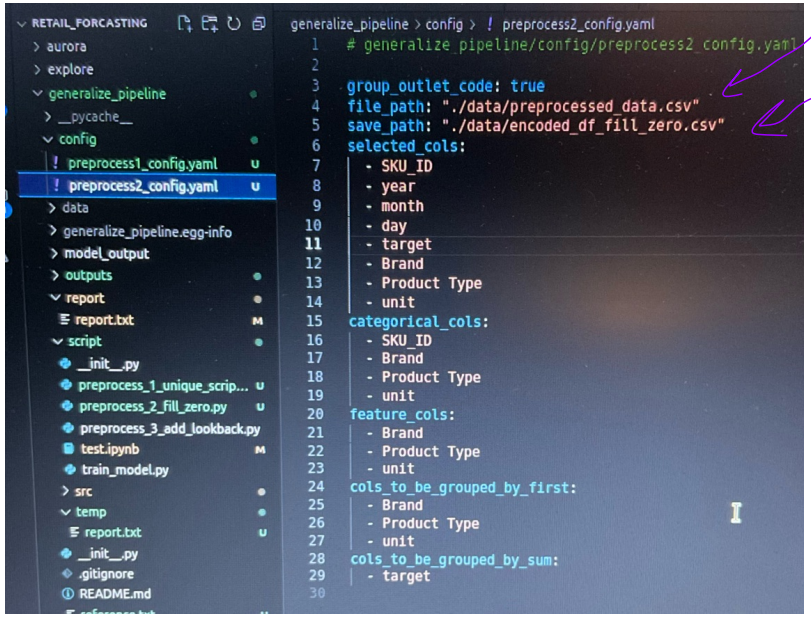
run at

Config file

Method1

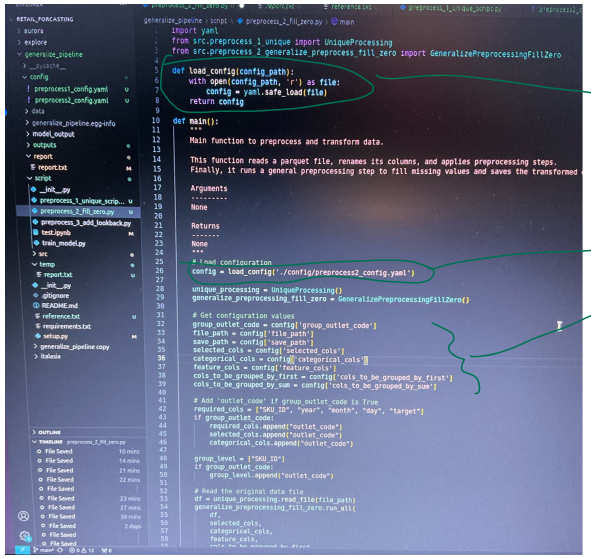
1. Write yaml file and load

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

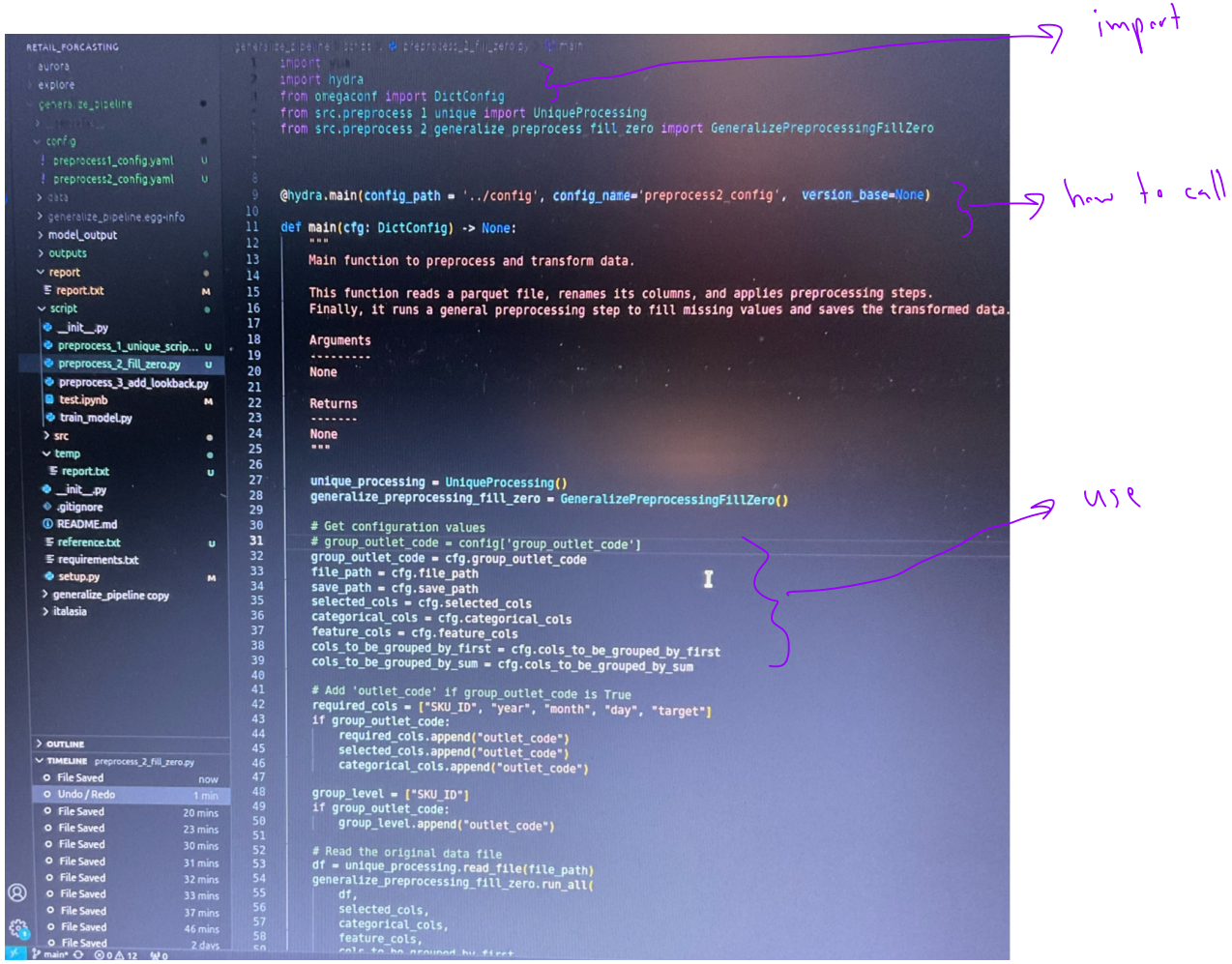


how to load

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-omegaconf-a-comprehensive-guide-5cf1d38e01f7



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