\*\*create python virtual environment and install pytorch cpu version in that and run a sample pytorch model in that

Python Virtual environment

\* Python Virtual Environment is an isolated space where you can work on your Python projects, separately from your system-installed Python.

\* You can set up your own libraries and dependencies without affecting the

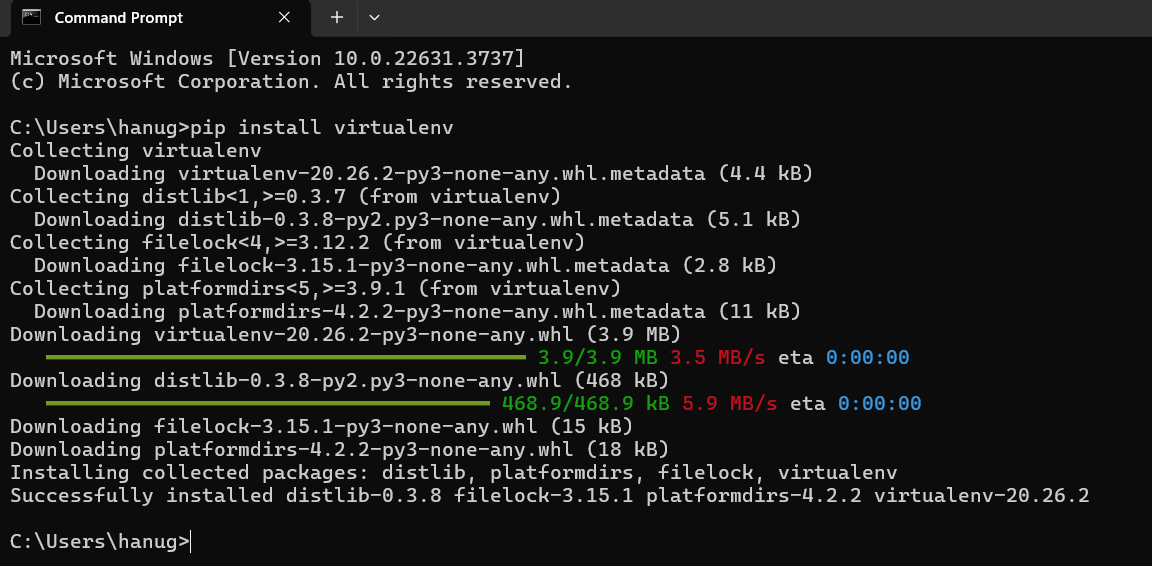
system Python.

\* we will use virtualenv to create a virtual environment in Python.

Creation of Python Virtual Environment

* Open cmd prompt then

1. Type command “ pip install virtualenv” for installing virtual environment



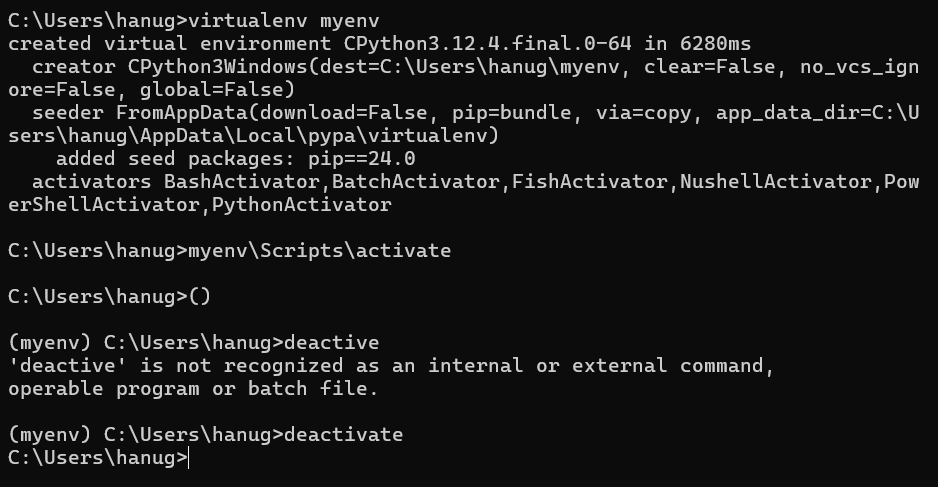
Now to create use “ virtualenv <name>

* Virtualenv myenv

To activate it cmd: <name>\Scripts\activate

myenv\Scripts\activate

for deactivate: use deactivate cmd or exit cmd



……………………………………………………………………………………………………………………………………………………………………………………..

Pytorch :

PyTorch is an open-source deep learning framework developed by Facebook's Al Research lab (FAIR). It is widely used for developing and training neural networks. Key features of PyTorch include:

\*Dynamic Computation Graphs: PyTorch uses dynamic computation graphs (define-by-run),making it easy to modify the network architecture during runtime. This is particularly useful for tasks involving variable input sizes, such as NLP.

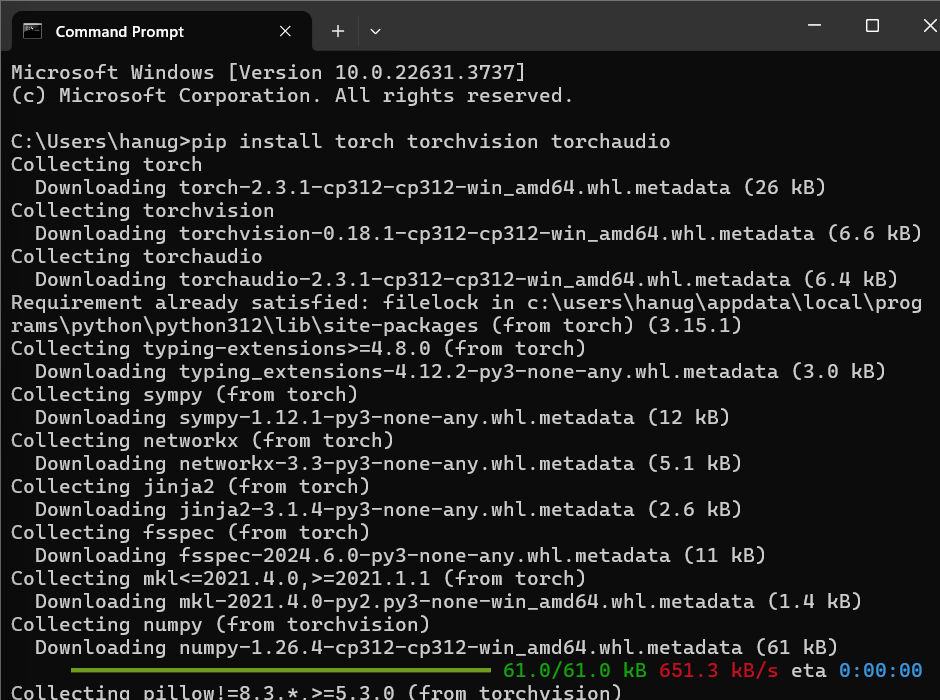
\*Easy to Use: PyTorch's interface is intuitive and similar to NumPy, making it easy for developers and researchers to learn and use.

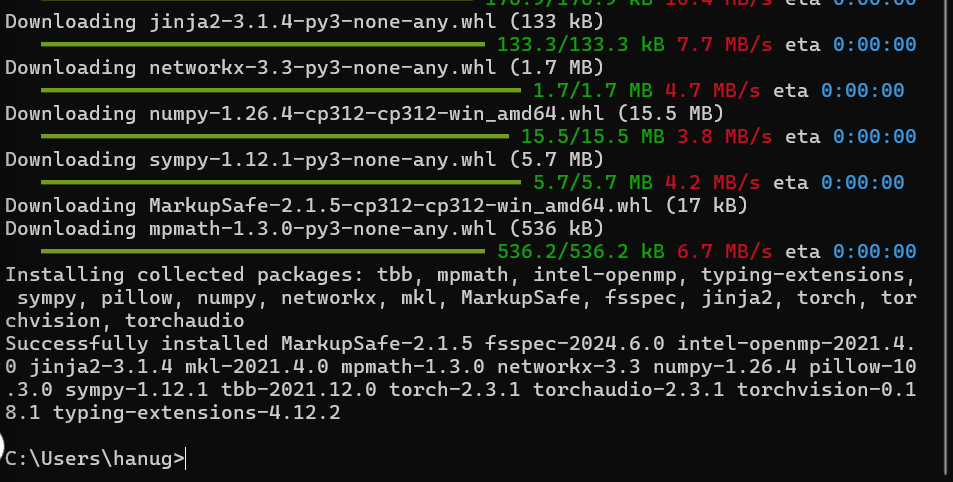
\*GPU Acceleration: PyTorch provides easy-to-use APIs to leverage GPUs for accelerating computations.

\*Integration with Python Ecosystem: PyTorch integrates seamlessly with other Python libraries, such as NumPy and SciPy.

Installing pytorch cpu version:

* Open cmd prompt type “ pip install torch torchvision torchaudio”





After the successful installation create a sample pytorch program;

Sample code:

import torch

import torch.nn as nn

import torch.optim as optim

# Define a simple neural network

class SimpleNN(nn.Module):

def \_\_init\_\_(self):

super(SimpleNN, self).\_\_init\_\_()

self.fc1 = nn.Linear(10, 5)

self.fc2 = nn.Linear(5, 1)

def forward(self, x):

x = torch.relu(self.fc1(x))

x = self.fc2(x)

return x

# Instantiate the model, loss function, and optimizer

model = SimpleNN()

criterion = nn.MSELoss()

optimizer = optim.SGD(model.parameters(), lr=0.01)

# Create dummy input and target tensors

input = torch.randn(10)

target = torch.randn(1)

# Forward pass

output = model(input)

loss = criterion(output, target)

# Backward pass and optimization

optimizer.zero\_grad()

loss.backward()

optimizer.step()

print("Output:", output)

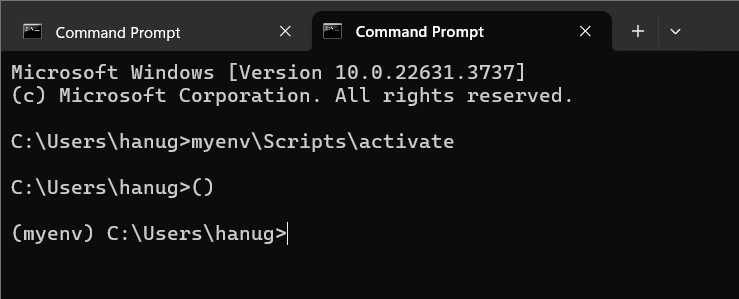
print("Loss:", loss.item())

saving as sample.py

now to run it first we need to activate the virtual enviroment

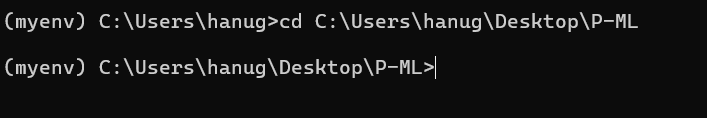
\*our virtual environment name is myenv

Cmd: myenv\Scripts\activate



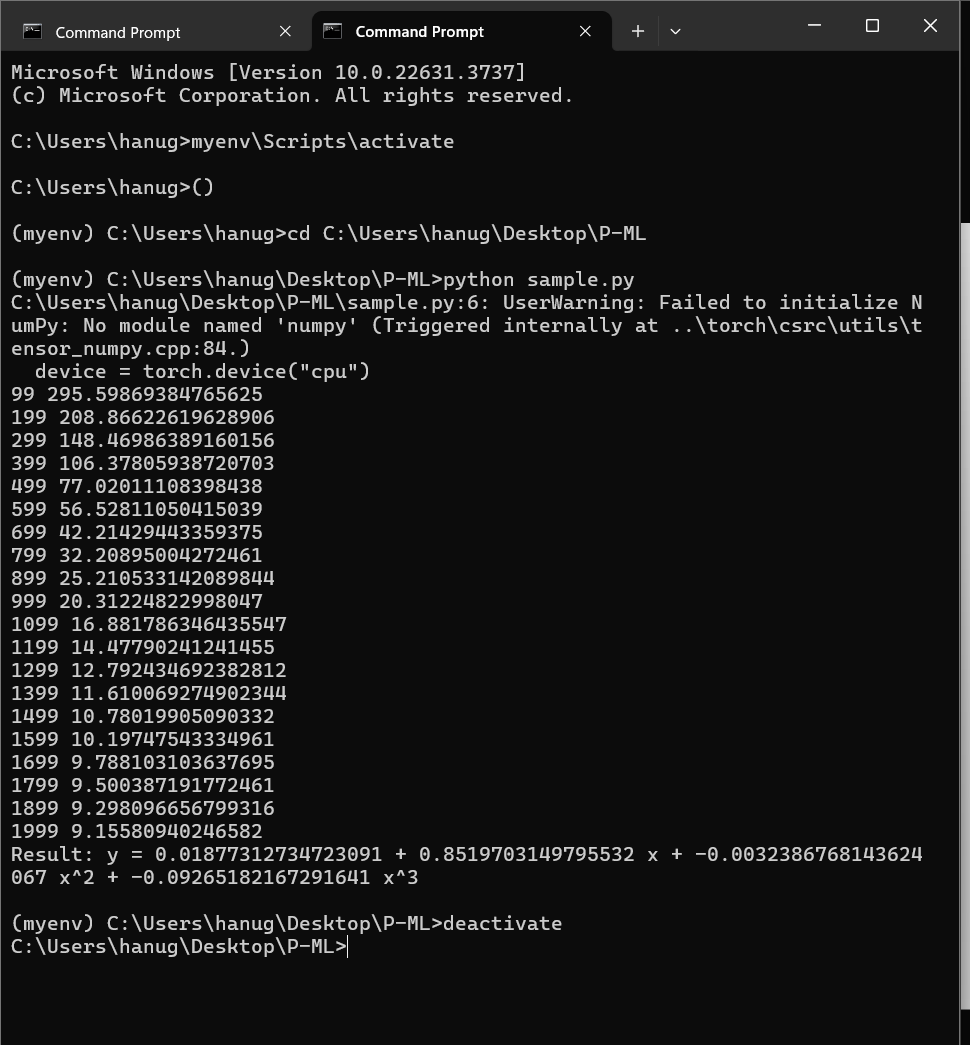
Now change the directory to program saved location

Cd <File path>



Now run the program by cmd: python filename.py

\*python sample.py



After successful execution u can deactivate the environment by cmd: deactivate.

……………………………………………………………………………………………………………………………………………………………………………………

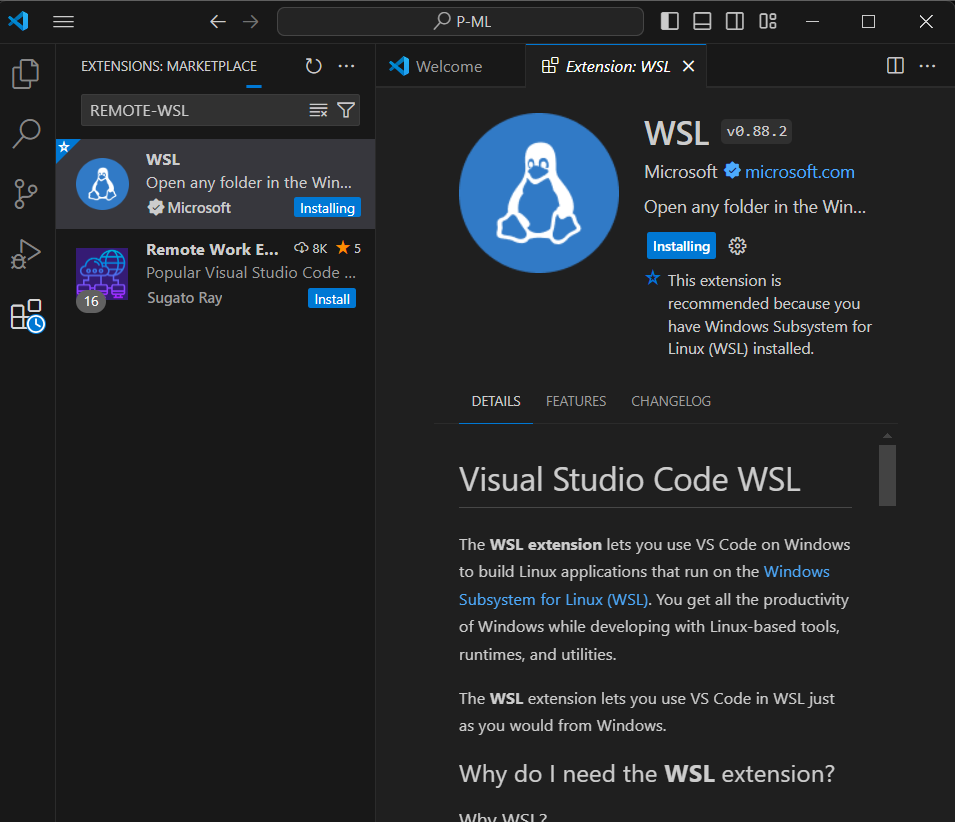
\*\*\*use visual studio code to access WSI terminal and modify and run the code

1.Install WSL (same procedure as in document 1)

2. install vscode 🡺https://code.visualstudio.com/download

3. open vs code and launch it

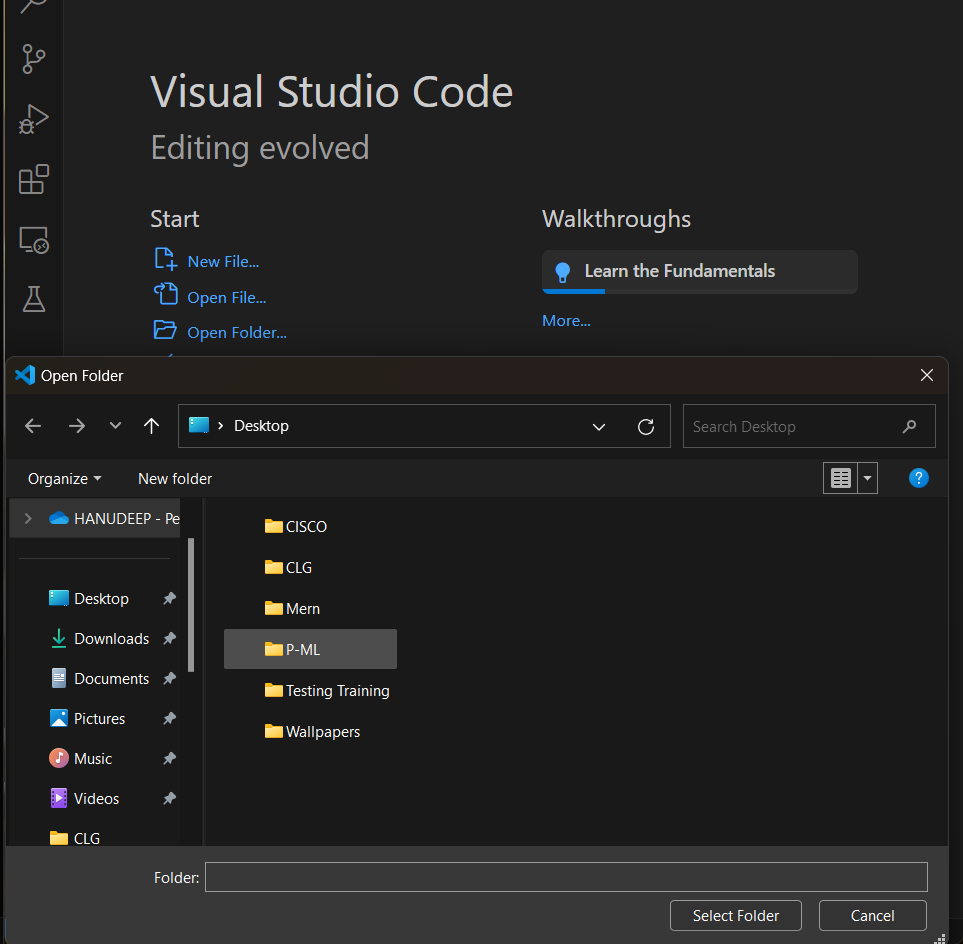
4.click on extensions and install remote wsl extension



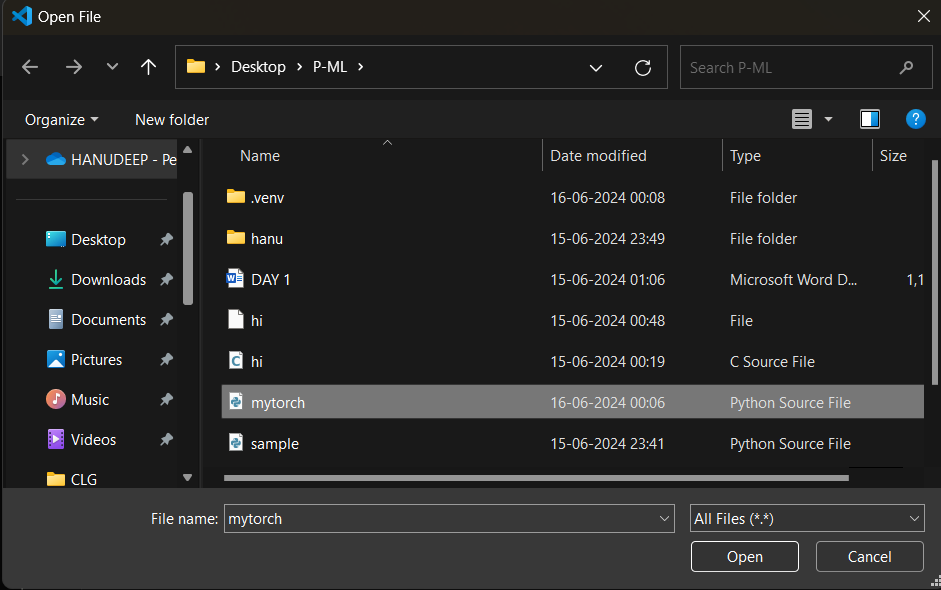
After downloading the file extensions .

Follow the process:

1.open the file location folder

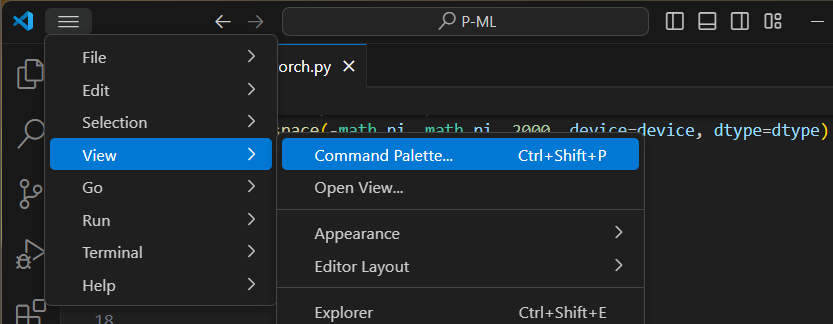


Then open the file…



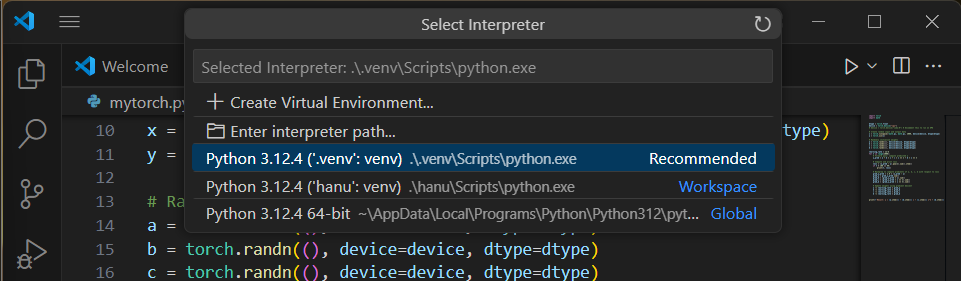
Then press ctrl +s to save ……

Now press ctrl+shift+p // go to view> select command pallate



In that choose python:Select interpreter

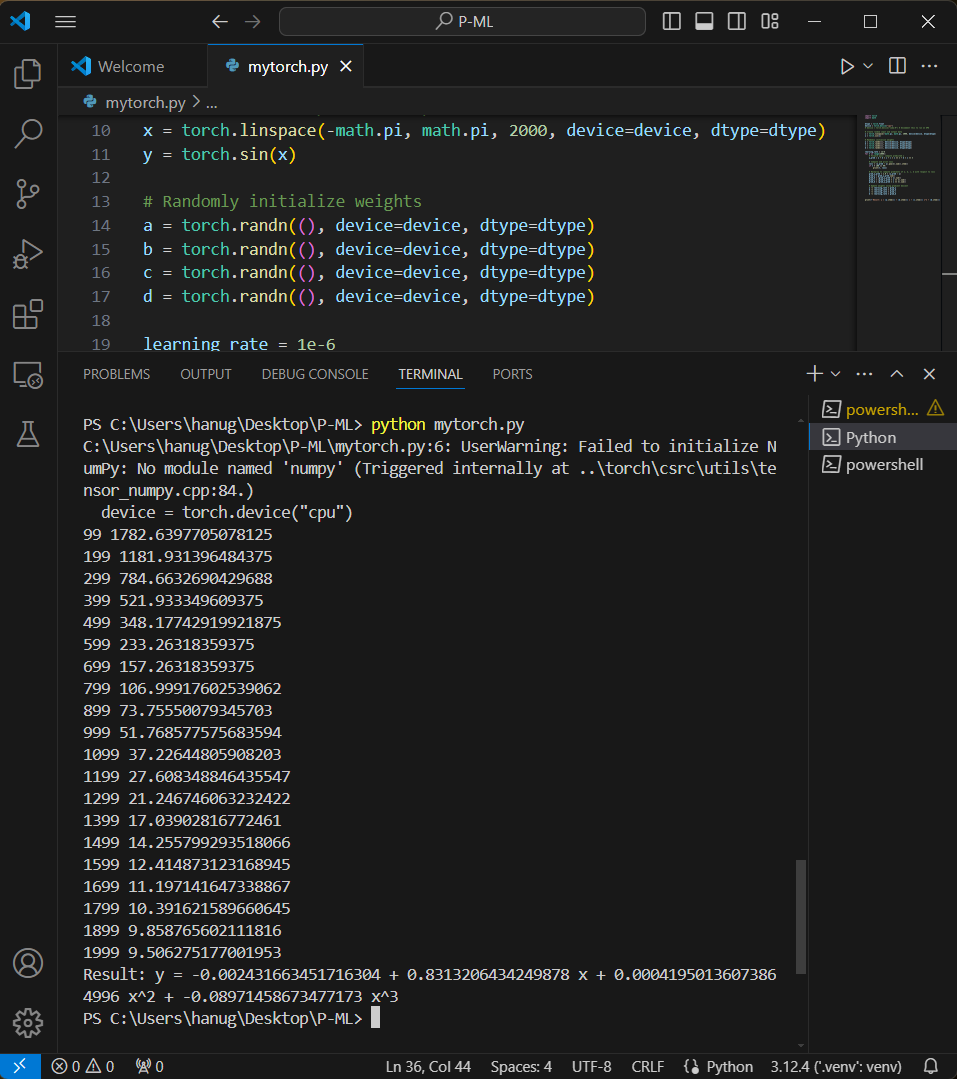
Then choose your virtual environment as it recommends



Now goto view>Terminal

To execite type python filename.py

\*python mytorch.py



Successfully completed executing pytorch program using vs code