



EC5ME
THE EDUCATION CONSORTIUM OF SMART
MANUFACTURING AND ELECTRONICS



國立清華大學
NATIONAL TSING HUA UNIVERSITY

SYNOPSYS®
Silicon to Software™

Workshop on Synopsys ARC CPU with TensorFlow Lite

Tutorial 3 – TensorFlow Project Environment Setup & Development Flow

主辦單位：國立清華大學電機系、新思科技

協辦單位：智慧製造電子應用聯盟

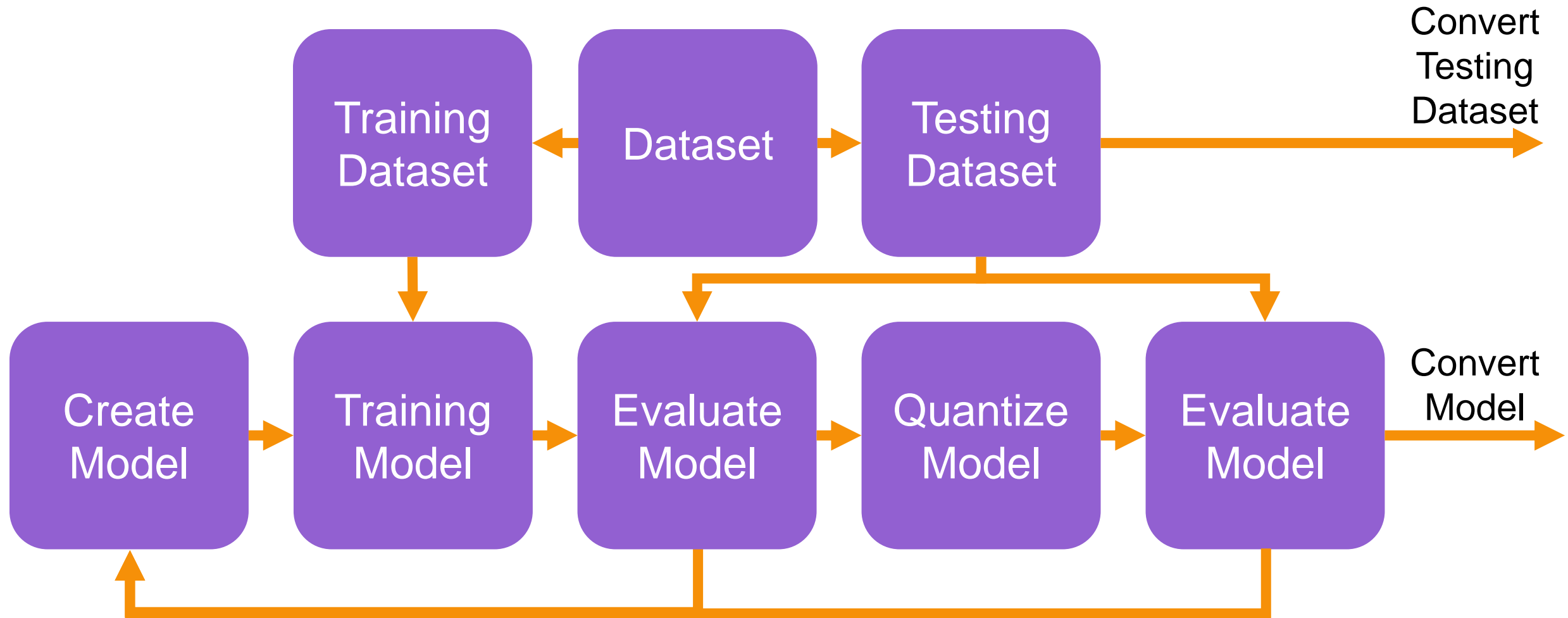
指導單位：教育部資訊及科技教育司

Project Development Flow



Stage	TensorFlow Model Development	Firmware Development	Running Application On WE-I
Tool	Anaconda Cygwin	Cygwin Metaware or ARC GNU VirtualBox (Ubuntu)	Tera Term USB Micro
Language	Python 3	C language C++ language	

TensorFlow Model Development





EC5ME
THE EDUCATION CONSORTIUM OF SMART
MANUFACTURING AND ELECTRONICS



國立清華大學
NATIONAL TSING HUA UNIVERSITY

SYNOPSYS®
Silicon to Software™

Anaconda3 Setup



Download

ANACONDA URL: <https://www.anaconda.com/products/individual>



Products ▾

Pricing

Solutions ▾

Resources ▾

Blog

Company ▾




Individual Edition

Your data science toolkit

With over 20 million users worldwide, the open-source Individual Edition (Distribution) is the easiest way to perform Python/R data science and machine learning on a single machine. Developed for solo practitioners, it is the toolkit that equips you to work with thousands of open-source packages and libraries.

Download


Anaconda Installers

Windows 

Python 3.8

64-Bit Graphical Installer (457 MB)


32-Bit Graphical Installer (403 MB)

MacOS 

Python 3.8

64-Bit Graphical Installer (435 MB)

64-Bit Command Line Installer (428 MB)

Linux 

Python 3.8

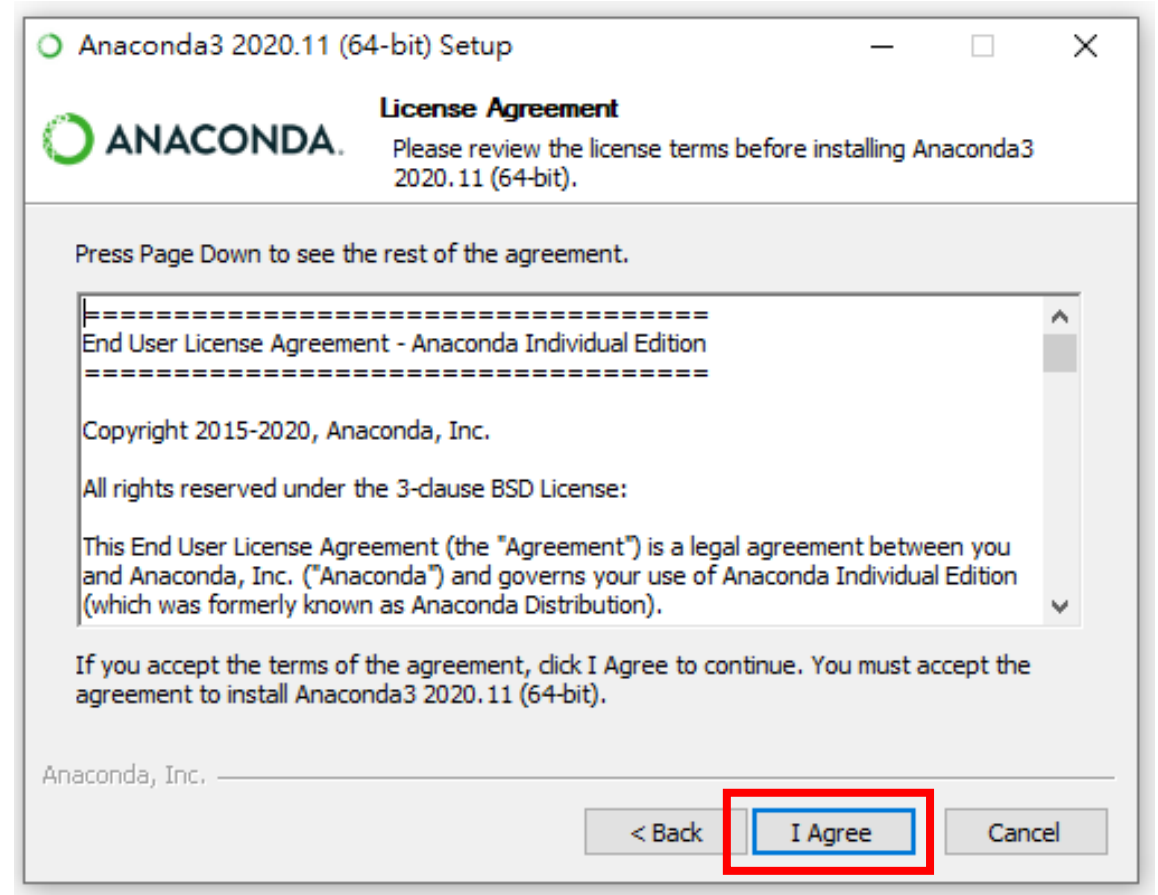
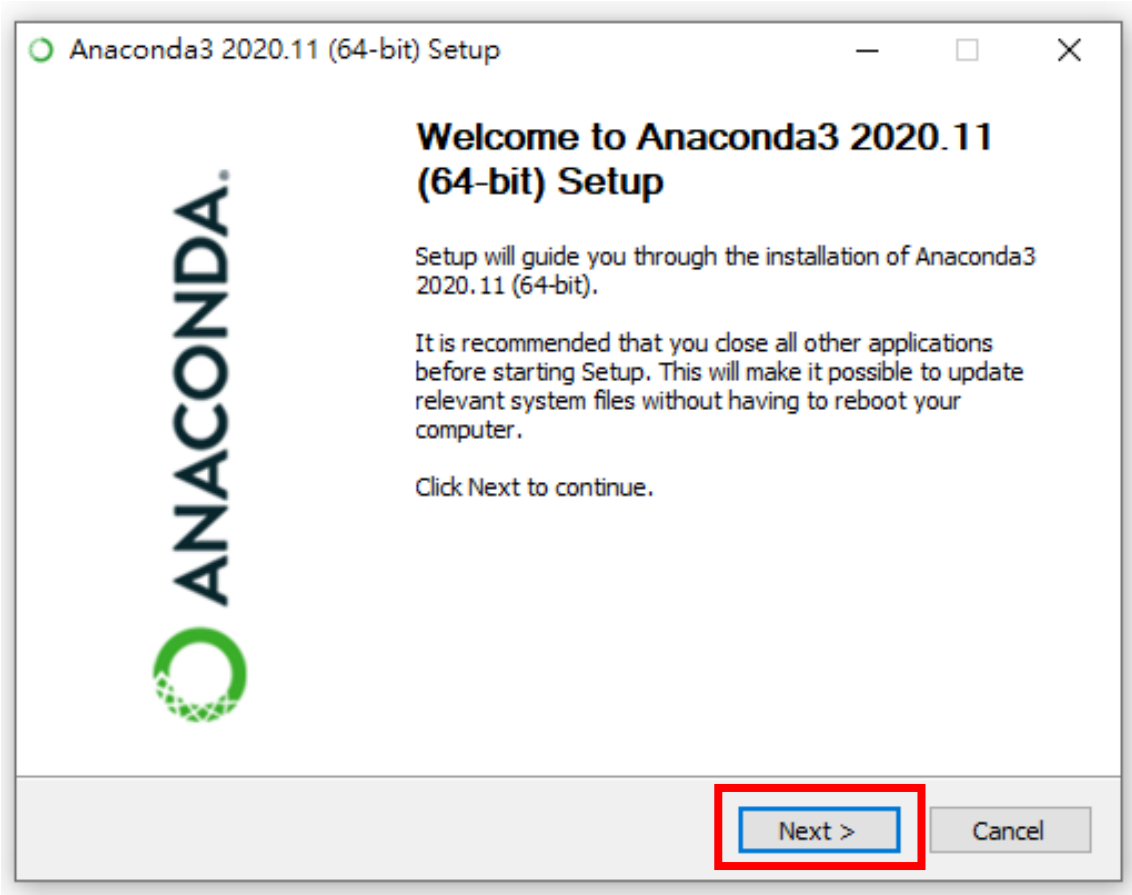
64-Bit (x86) Installer (529 MB)

64-Bit (Power8 and Power9) Installer (279 MB)

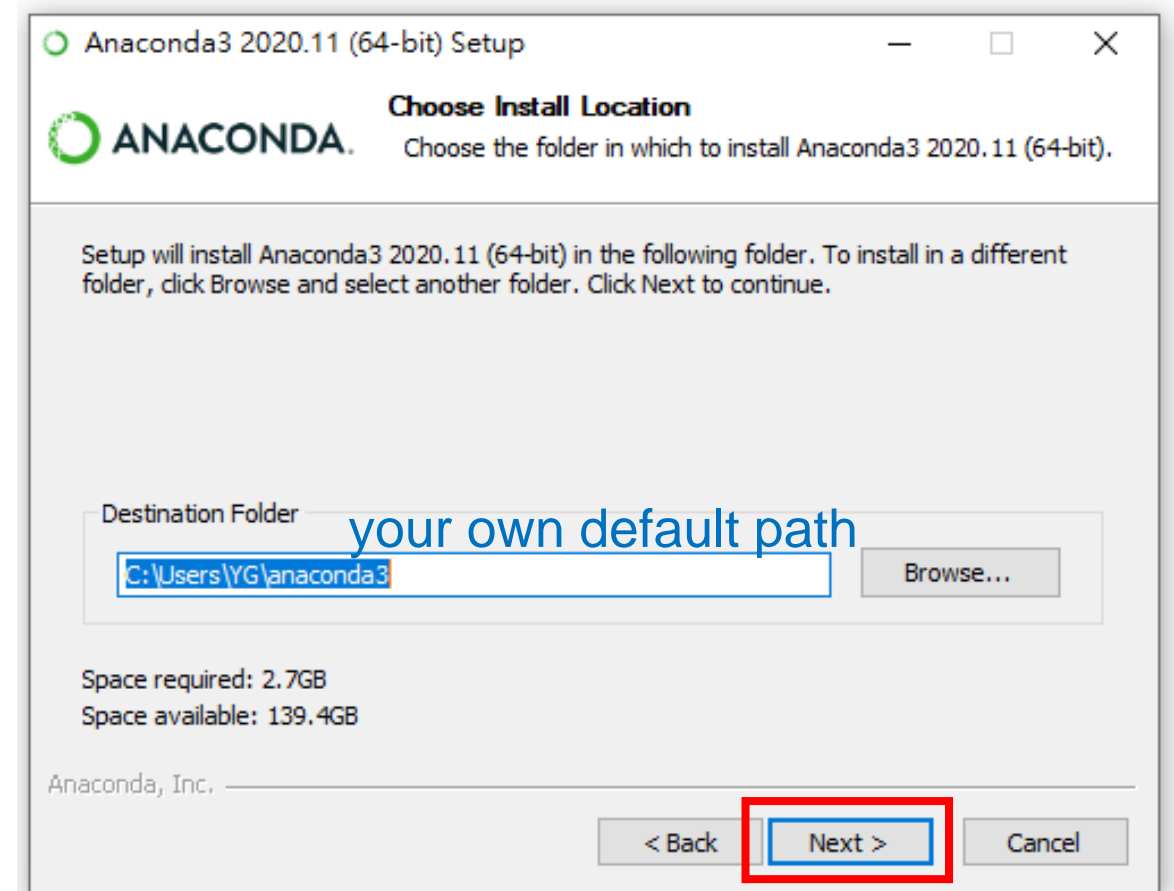
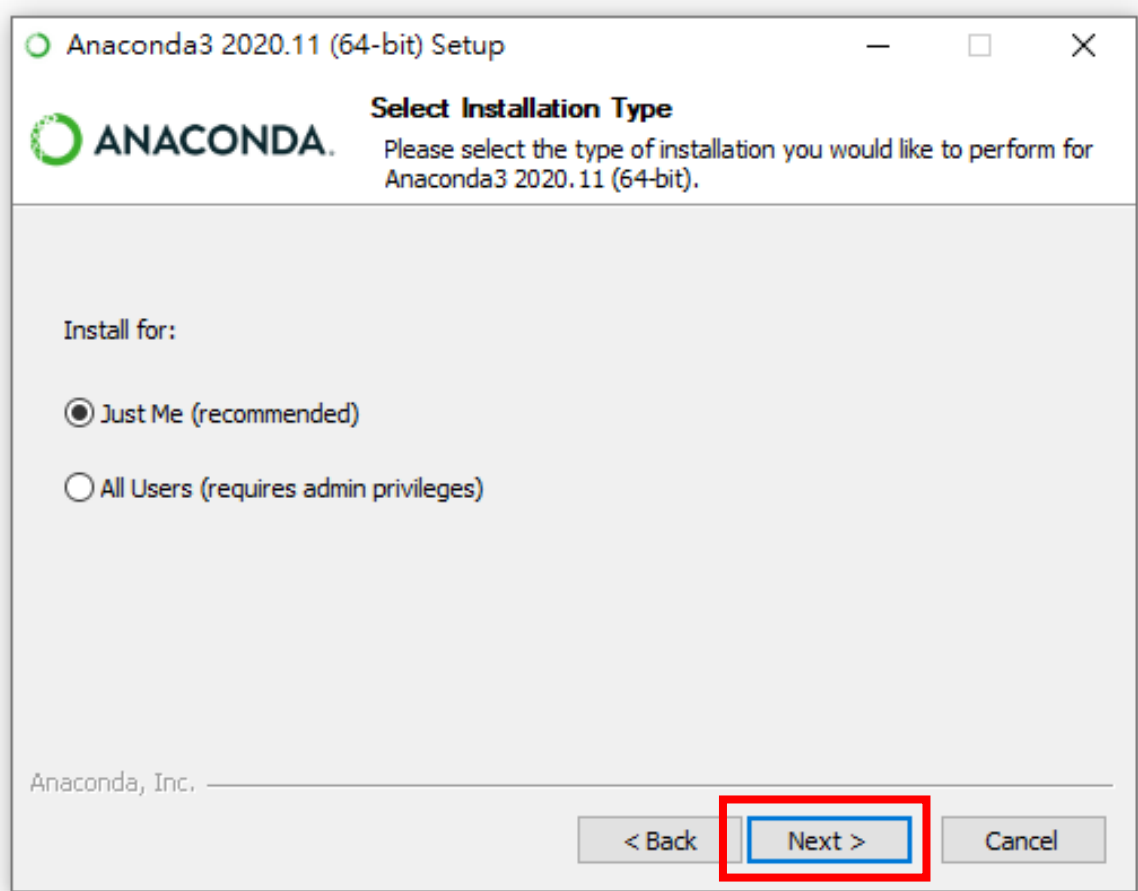
ADDITIONAL INSTALLERS

The [archive](#) has older versions of Anaconda Individual Edition installers. The Miniconda installer homepage can be found [here](#).

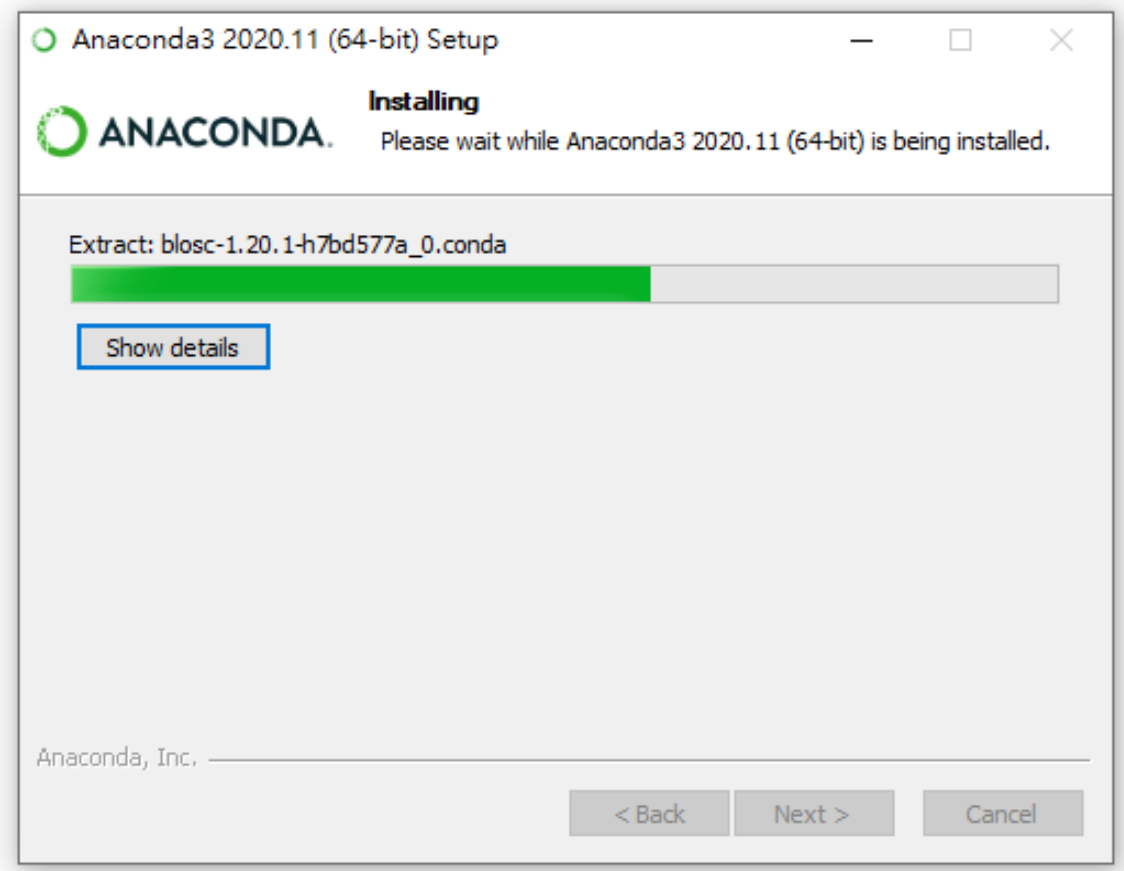
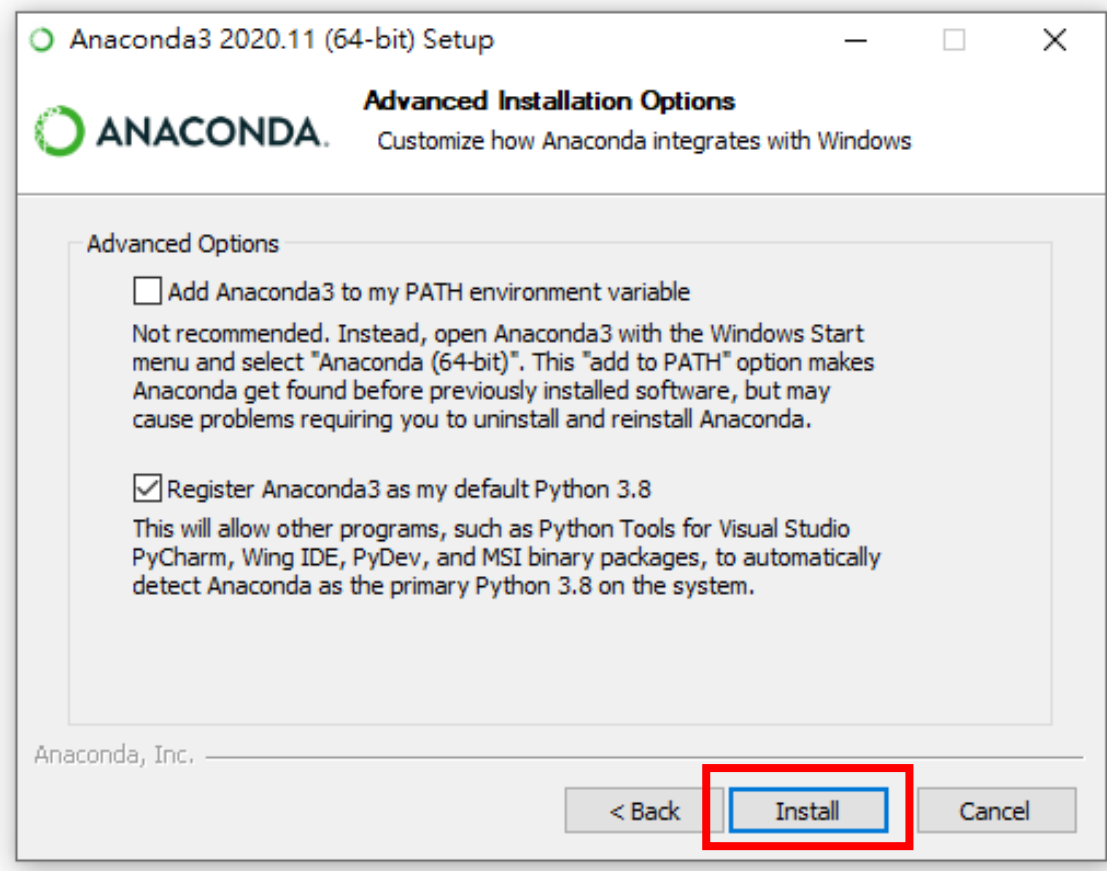
Anaconda3 Setup



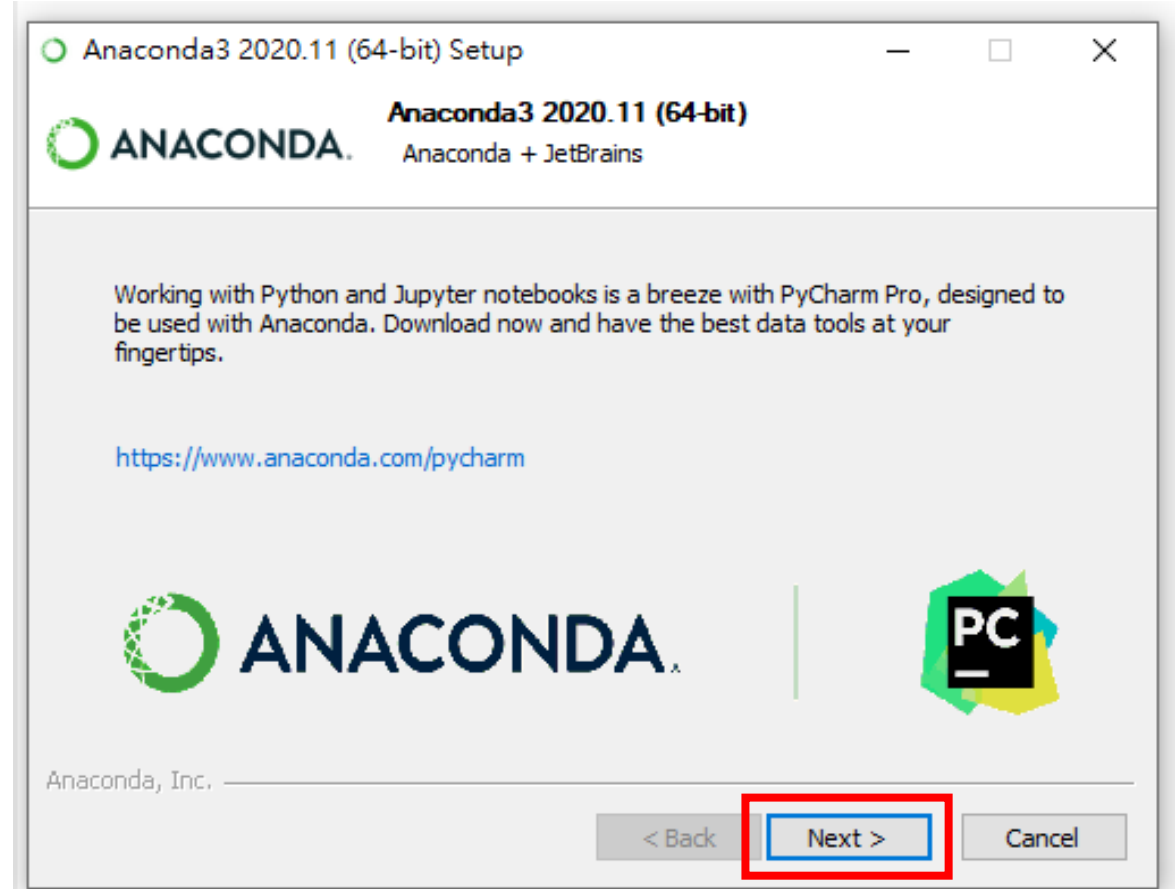
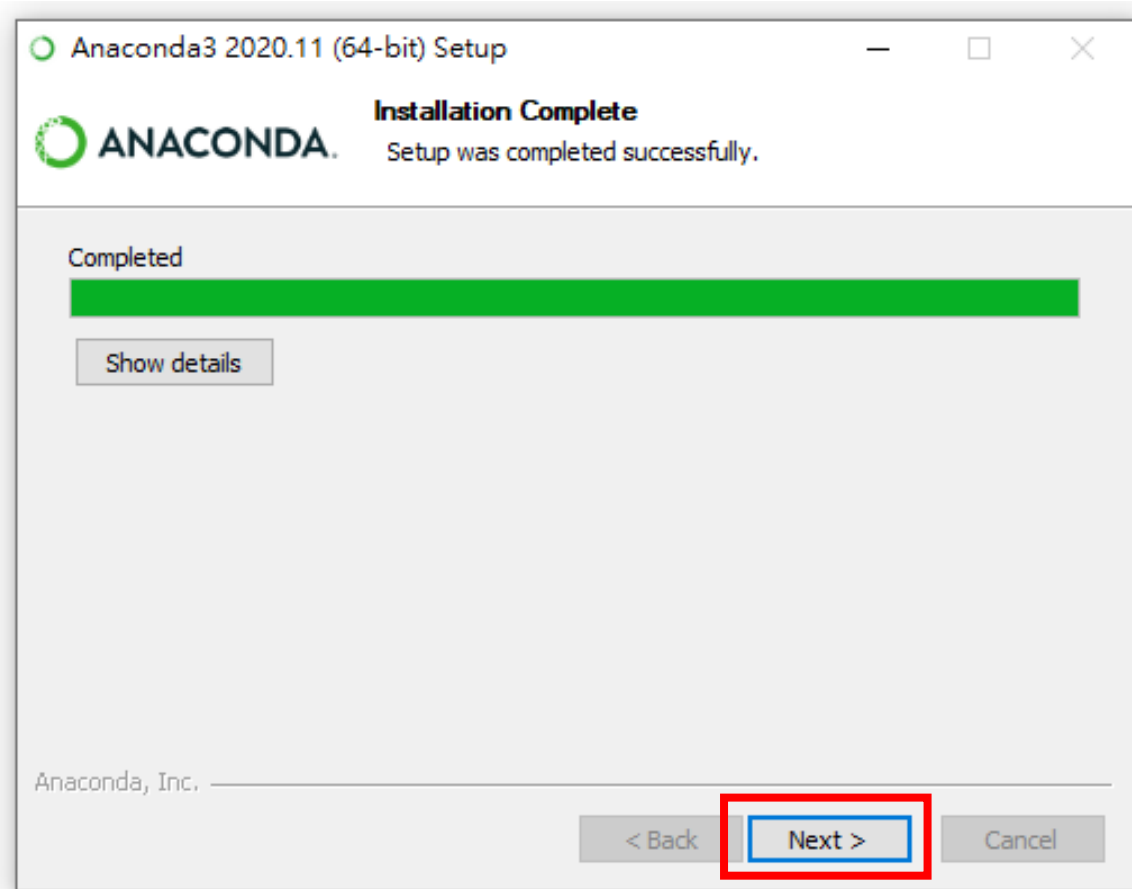
Anaconda3 Setup



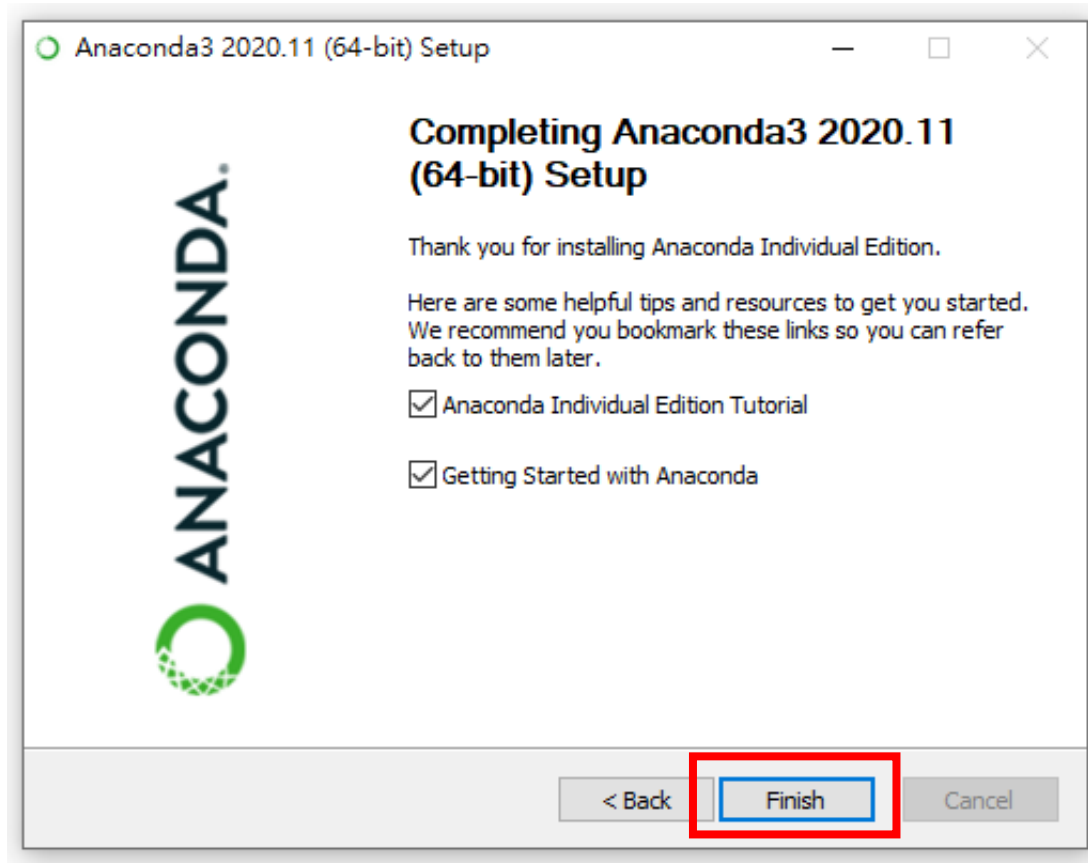
Anaconda3 Setup



Anaconda3 Setup



Anaconda3 Setup



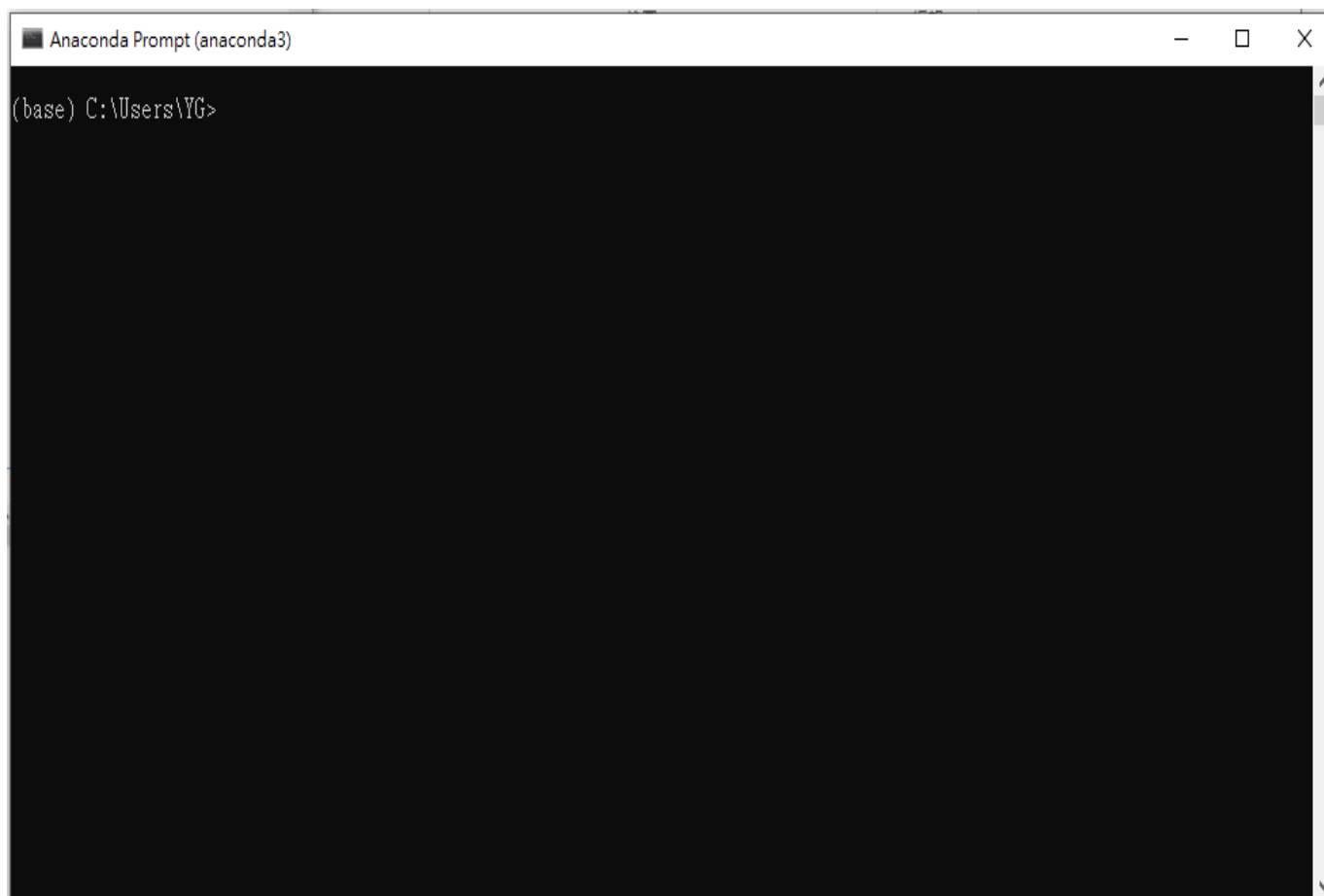
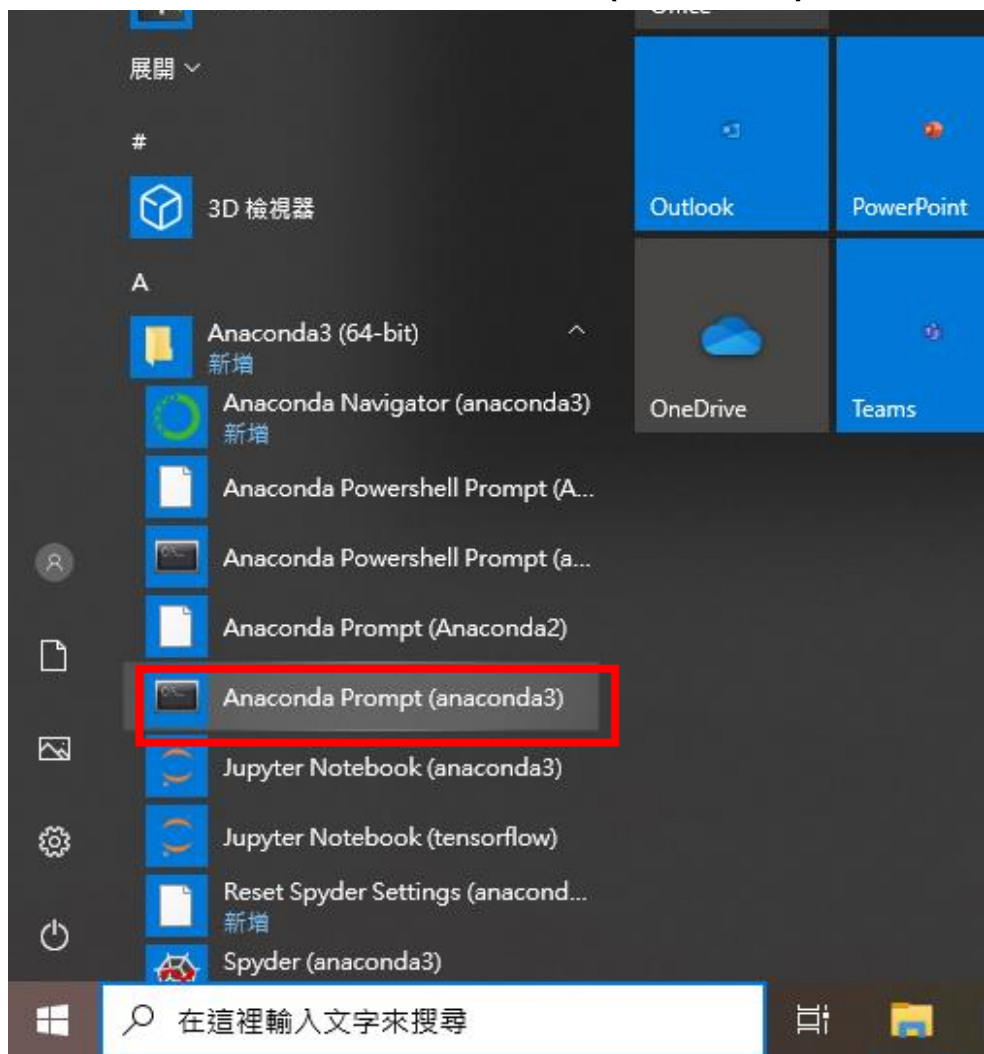


Tensorflow Environment Setup



Tensorflow Environment Setup

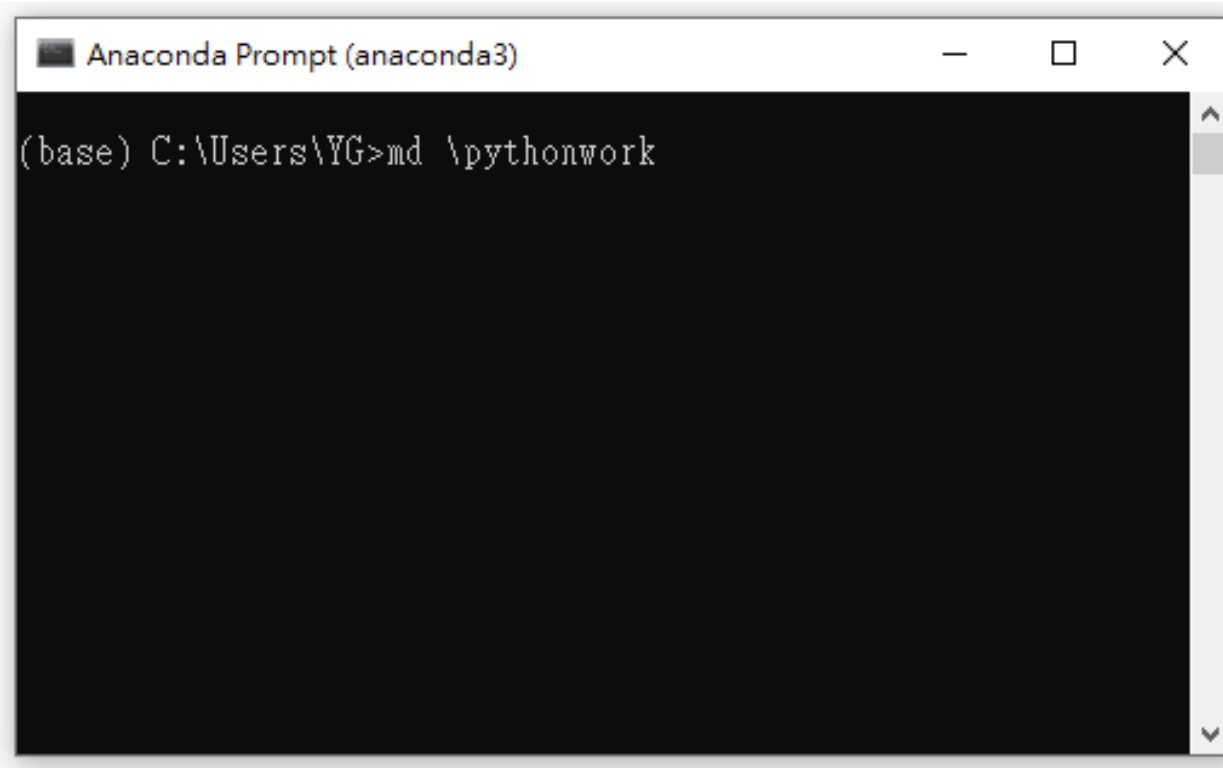
開始 > Anaconda3 (64-bit) > Anaconda Prompt (anaconda3)



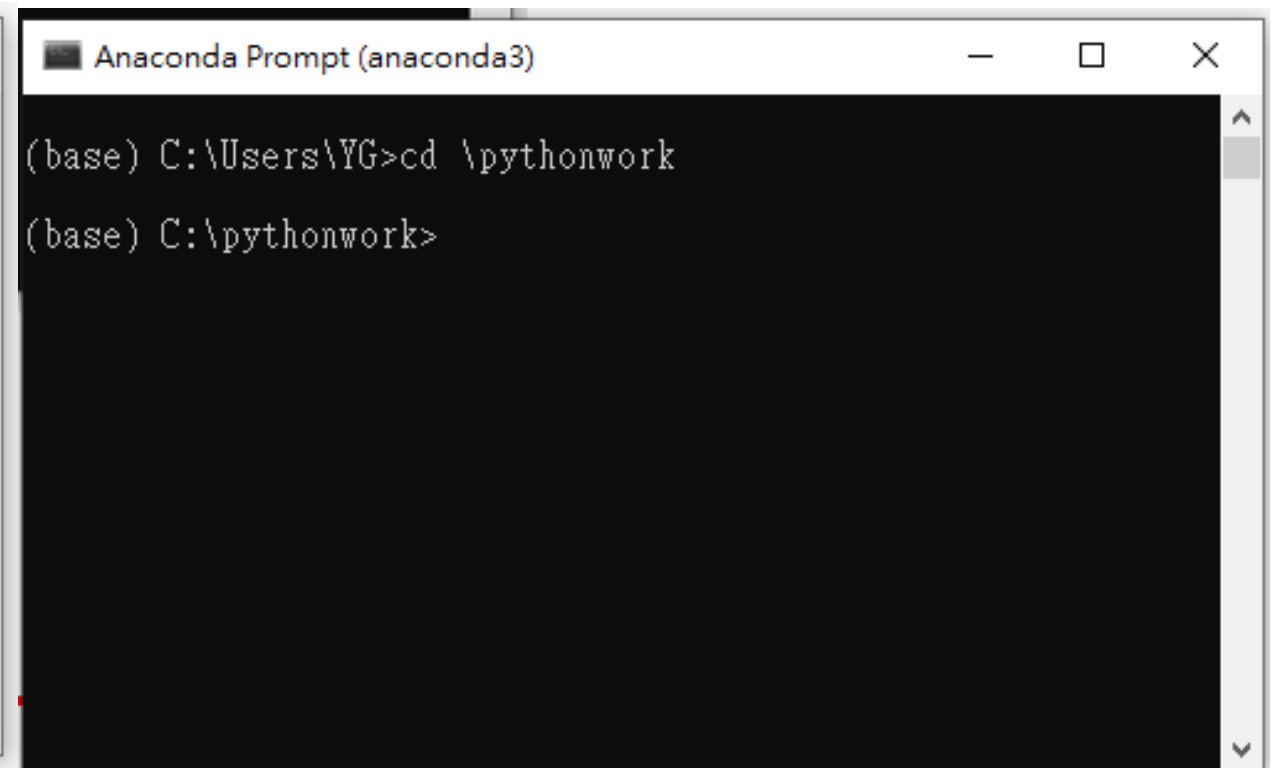
Tensorflow Environment Setup

\$ `md \pythonwork` (建立工作資料夾pythonwork)

\$ `cd \pythonwork`



```
Anaconda Prompt (anaconda3)
(base) C:\Users\YG>md \pythonwork
```



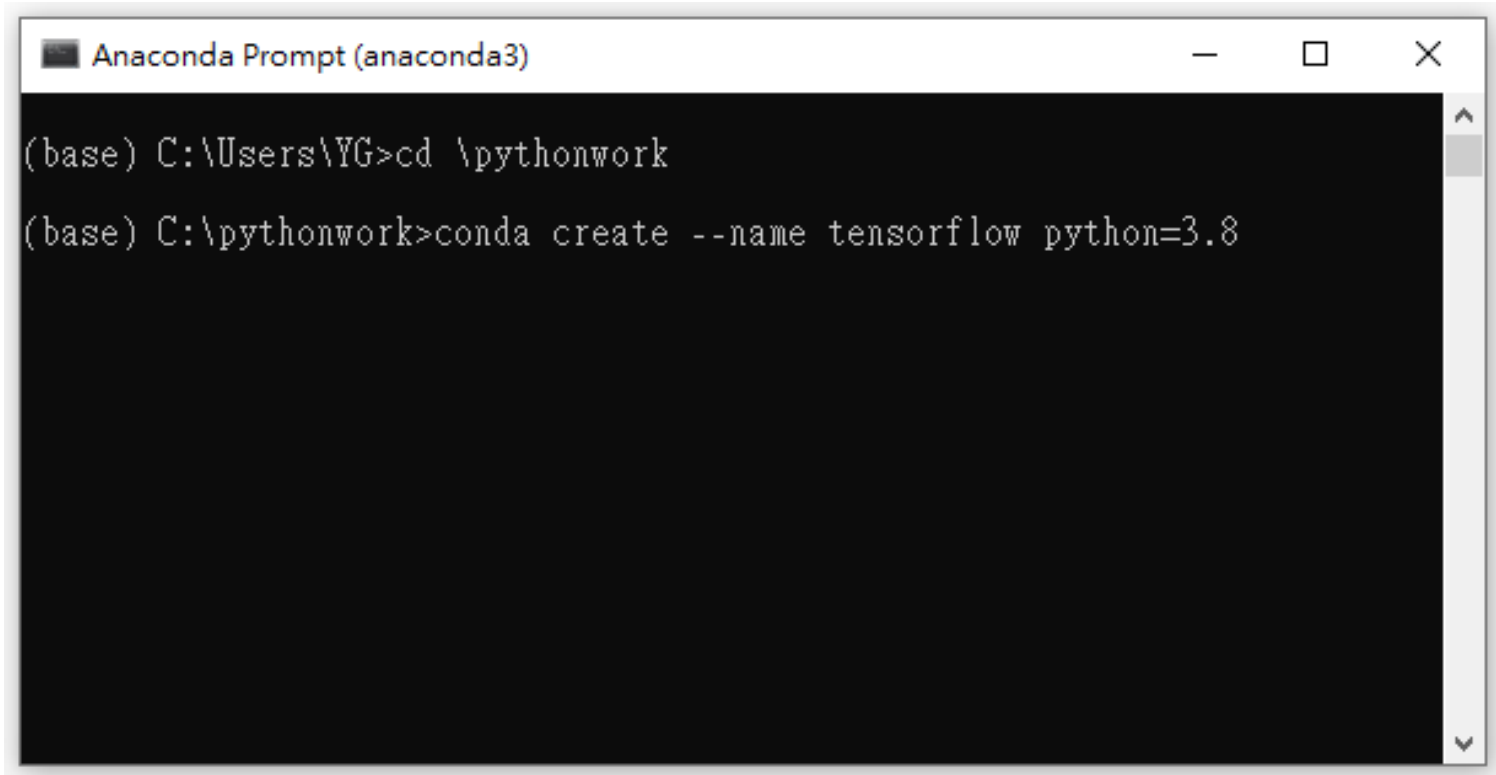
```
Anaconda Prompt (anaconda3)
(base) C:\Users\YG>cd \pythonwork
(base) C:\pythonwork>
```

Tensorflow Environment Setup

使用 conda 命令來建立一個命名為 tensorflow 的虛擬環境
並在裡面安裝 Python 3.8 版本

\$ conda create --name tensorflow python=3.8

畫面出現“Proceed([y]/n)?”
請按 **y** 繼續

A screenshot of the Anaconda Prompt window. The title bar reads "Anaconda Prompt (anaconda3)". The command prompt shows the user navigating to the \pythonwork directory and then running the command "conda create --name tensorflow python=3.8".

```
Anaconda Prompt (anaconda3)

(base) C:\Users\YG>cd \pythonwork
(base) C:\pythonwork>conda create --name tensorflow python=3.8
```

TensorFlow Environment Setup

1. 啟動剛建立的anaconda虛擬環境

\$ `conda activate tensorflow`

2. 安裝 Tensorflow

\$ `conda install tensorflow==2.3.0`

3. 安裝 Keras

\$ `conda install -c conda-forge keras`

4. 安裝 matplotlib

\$ `conda install matplotlib`

5. 安裝 numpy

\$ `conda install numpy`

6. 安裝 emnist

\$ `pip install emnist`

7. 安裝 Jupyter notebook

\$ `conda install jupyter notebook`

• 如果安裝過程遇到問題，可以上網查詢相關強制安裝指令。



EC5ME
THE EDUCATION CONSORTIUM OF SMART
MANUFACTURING AND ELECTRONICS



國立清華大學
NATIONAL TSING HUA UNIVERSITY

SYNOPSYS®
Silicon to Software™

TensorFlow Environment Test



TensorFlow Environment Test

1. 複製資料夾

“C:\workshop\himax_tflm\Synopsys_WEI\Example_Project\Lab5_tflm_conversion_tutorial”

到資料夾 “C:\Users\{username}\”
(Jupyter Notebook預設路徑)

2. 開始 > Aanaconda3 (64-bit) > Jupyter Notebook (tensorflow)



Quit

Logout

Files

Running

Clusters

Select items to perform actions on them.

Upload

New ▾



☐ 0



📁 /

Name ▾

Last Modified

File size

☐ 📁 3D Objects

4 days ago

☐ 📁 _TF2

2 months ago

TensorFlow Environment Test

1. 回到 Jupyter Notebook
2. 點選資料夾 Lab5_tflm_conversion_tutorial
3. 開啟 model_conversion.ipynb

Files

Running

Clusters

Select items to perform actions on them.

UploadNew↻

☐ 0 ▾

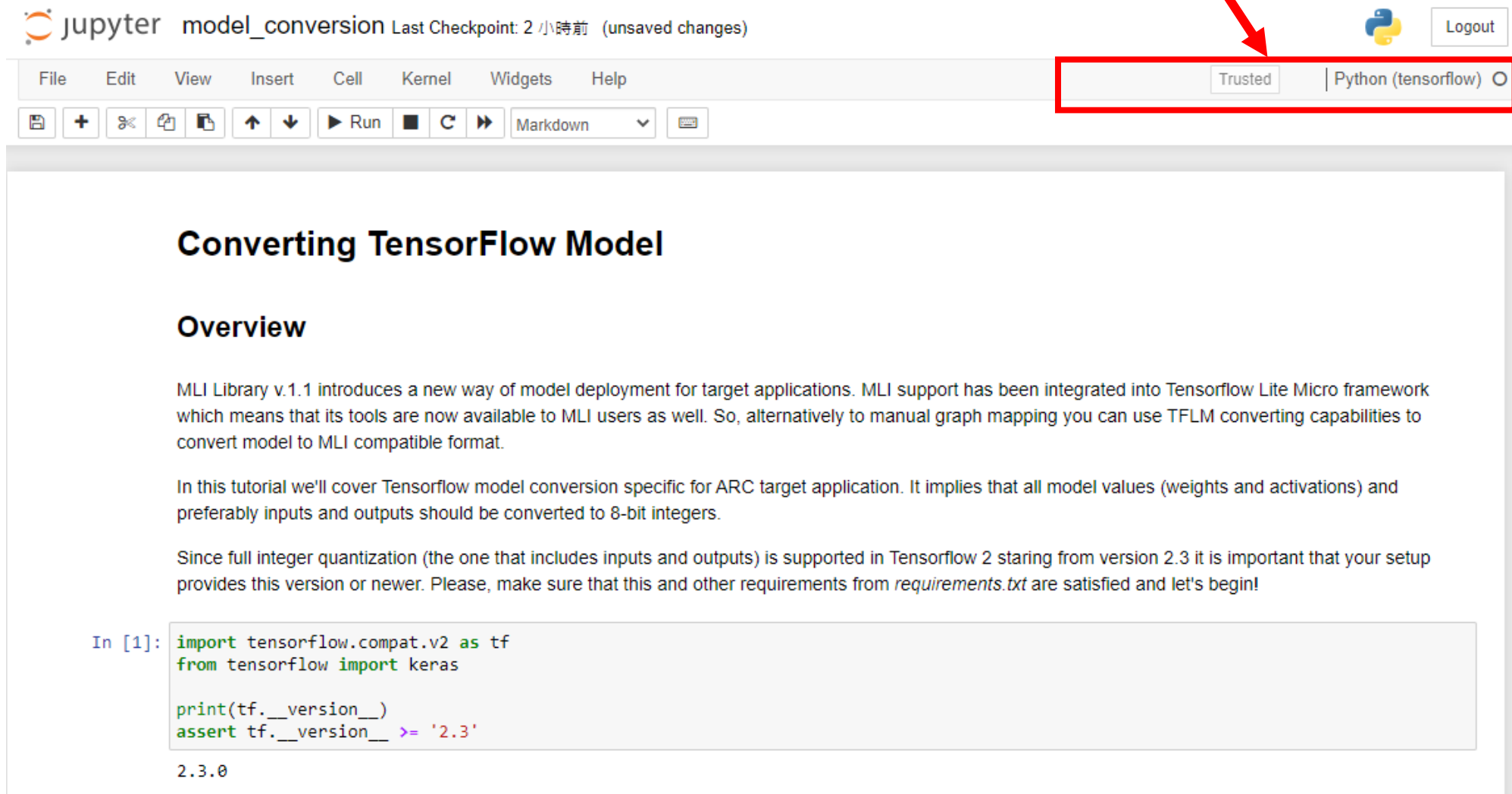
📁 / Lab5_tflm_conversion_tutorial

Name ▾Last ModifiedFile size

<input type="checkbox"/>	📁 ..	seconds ago	
<input type="checkbox"/>	📄 model_conversion.ipynb	11 days ago	15.3 kB
<input type="checkbox"/>	📄 mli_cnn_bn.h5	11 days ago	282 kB
<input type="checkbox"/>	📄 requirements.txt	11 days ago	82 B

TensorFlow Environment Test

4. 請先確認紅框內是否顯示Python (tensorflow) , 且()中的環境是否正確



The screenshot shows a Jupyter Notebook interface. At the top, the title bar reads "jupyter model_conversion Last Checkpoint: 2 小時前 (unsaved changes)". Below the title bar is a menu bar with options: File, Edit, View, Insert, Cell, Kernel, Widgets, Help. A toolbar below the menu bar contains icons for file operations, a "Run" button, and a "Markdown" dropdown. On the right side of the interface, there is a "Trusted" button and a dropdown menu showing "Python (tensorflow)". A red box highlights this area, and a red arrow points to the "Python (tensorflow)" text. The main content area of the notebook displays the title "Converting TensorFlow Model" and a section "Overview". The text in the "Overview" section discusses the MLI Library v.1.1 and TensorFlow model conversion. At the bottom, a code cell is shown with the following code:

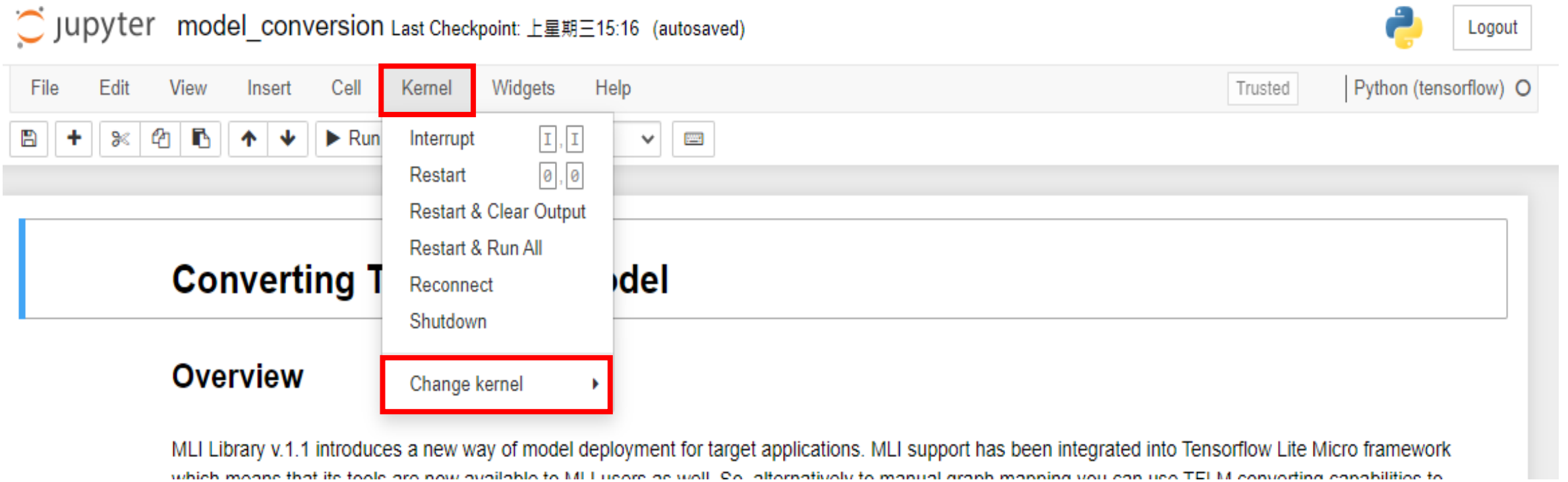
```
In [1]: import tensorflow.compat.v2 as tf
        from tensorflow import keras

        print(tf.__version__)
        assert tf.__version__ >= '2.3'
```

2.3.0

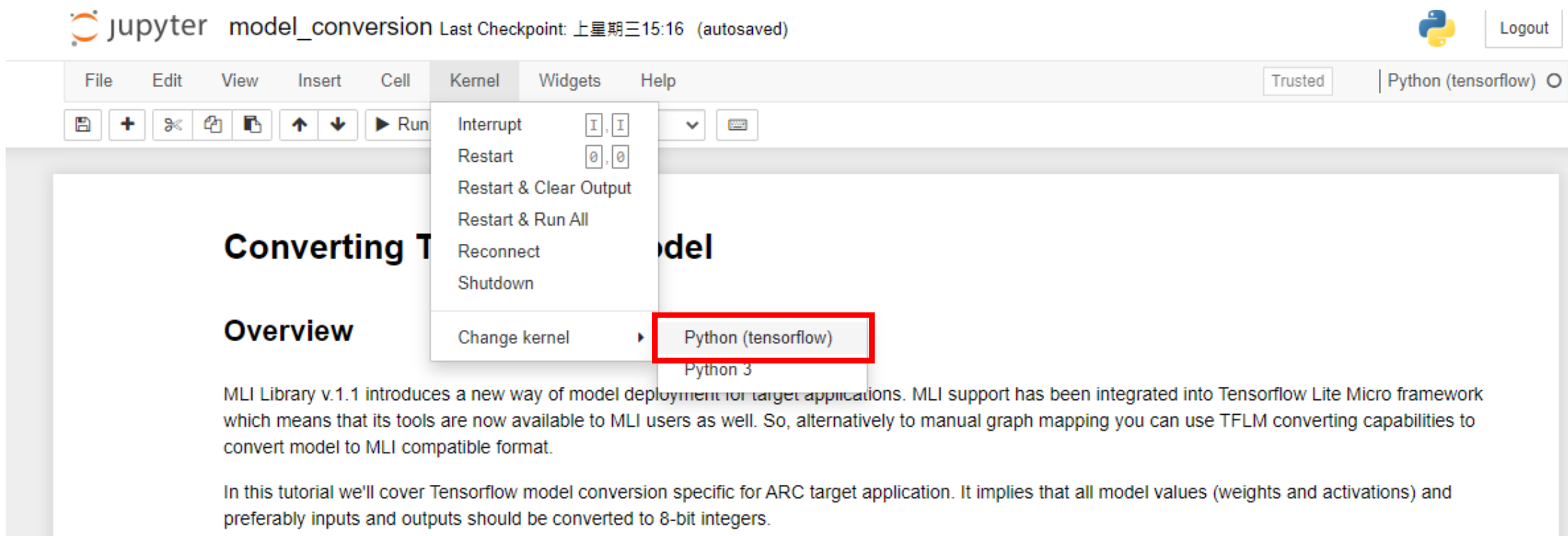
TensorFlow Environment Test

5. 若上頁有誤請試著做以下步驟修正:



The screenshot shows the JupyterLab interface for a notebook named 'model_conversion'. The top bar indicates the last checkpoint was saved on Wednesday at 15:16 (autosaved). The 'Kernel' menu is open, showing options: Interrupt, Restart, Restart & Clear Output, Restart & Run All, Reconnect, Shutdown, and 'Change kernel' (which is highlighted with a red box). The background shows the notebook content with the title 'Converting T...' and an 'Overview' section. The 'Overview' section text reads: 'MLI Library v.1.1 introduces a new way of model deployment for target applications. MLI support has been integrated into Tensorflow Lite Micro framework which means that its tools are now available to MLI users as well. So, alternatively to manual graph mapping you can use TEL M converting capabilities to'.

TensorFlow Environment Test



The screenshot displays the JupyterLab interface for a notebook named 'model_conversion'. The top bar shows the Jupyter logo, the notebook name, and the last checkpoint information: 'Last Checkpoint: 上星期三15:16 (autosaved)'. On the right, there is a Python logo and a 'Logout' button. The main menu bar includes 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. Below the menu bar, a toolbar contains icons for saving, creating new files, undo, redo, and running cells. The 'Kernel' menu is open, showing options: 'Interrupt', 'Restart', 'Restart & Clear Output', 'Restart & Run All', 'Reconnect', 'Shutdown', and 'Change kernel'. The 'Change kernel' option is expanded, and 'Python (tensorflow)' is highlighted with a red rectangle. The background content of the notebook shows the title 'Converting T...' and 'Overview' section, which discusses MLI Library v.1.1 and TensorFlow Lite Micro framework.

jupyter model_conversion Last Checkpoint: 上星期三15:16 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python (tensorflow)

Interrupt Restart Restart & Clear Output Restart & Run All Reconnect Shutdown Change kernel Python (tensorflow) Python 3

Converting T...

Overview

MLI Library v.1.1 introduces a new way of model deployment for target applications. MLI support has been integrated into Tensorflow Lite Micro framework which means that its tools are now available to MLI users as well. So, alternatively to manual graph mapping you can use TFLM converting capabilities to convert model to MLI compatible format.

In this tutorial we'll cover Tensorflow model conversion specific for ARC target application. It implies that all model values (weights and activations) and preferably inputs and outputs should be converted to 8-bit integers.

TensorFlow Environment Test

6. 若Kernel沒有Python(tensorflow)的選項:

安裝 ipykernel

```
$ pip install ipykernel
```

在(tensorflow)環境下輸入

```
$ python -m ipykernel install --name tensorflow
```

然後啟動jupyter notebook，點選kernel會發現有tensorflow

TensorFlow Environment Test

7. 若執行過程有錯誤，請確認下列module是否有安裝，或版本正確。

Numpy>=1.16.4

matplotlibjupyterlab>=1.1.0

tensorflow==2.3.0

keras>=2.2.4

emnist

TensorFlow Environment Test

8. `evaluate_model`會執行較久，請耐心等待

```
if prediction_values[index] == test_labels[index]:  
    accurate_count += 1  
accuracy = accurate_count * 1.0 / len(prediction_values)  
  
return accuracy * 100
```

Please, keep in mind that full test dataset evaluation on int8 model may take several minutes.

```
In [*]: ▶ print(str(evaluate_model(interpreter)) + "%")
```

Create a test set for target application

TensorFlow Environment Test

9. 執行完成後，回到Lab5_tflm_conversion_tutorial資料夾
會產生[generated/emnist_model_int8.tflite](#)與[test_samples.cc](#)

代表TensorFlow開發環境已經安裝完成

Lab5_tflm_conversion_tutorial

|

----mli_cnn_bn.h5

----model_conversion.ipynb

----requirements.txt

---- test_samples.cc

----generated

|

---- mnist_model_int8.tflite

TensorFlow Environment Test

10. 開啟Cygwin，並移動到Lab5_tflm_conversion_tutorial/generated

```
$ cd c:
```

```
$ cd Users/{username}/ (Jupyter Notebook Path)
```

```
$ cd Lab5_tflm_conversion_tutorial/generated/
```

11. Convert tflite to C model

```
$ cd Lab5_tflm_conversion_tutorial/generated/
```

```
$ xxd -i emnist_model_int8.tflite > model.h
```

12. You will see your TensorFlow model file model.h

13. Integrate model.h and test_samples.cc to your WE-I project
(Later tutorial)