

# NUS AI WORKSHOP DOC

**Intel oneAPI Hands-on workshop using Intel oneAPI AI  
Analytics toolkit and Intel OpenVINO toolkit**

**8th December 2020**

## Jupyter Notebook Basics

- a) This command at the top of a particular cell will create a bash script containing all the commands of that cell in the “filename.sh” file.

```
%%writefile filename.sh
```

- b) The Job(bash script file) is submitted to run on a CPU node using the following command

```
qsub filename.sh -l nodes=1:ppn=2 -d .
```

- c) To check the status of all the jobs `qstat` is used. It shows the running jobs.
- d) After the job is finished running, it creates one output and one error file with name `filename.sh.o.` & `filename.sh.e.` with some number as suffix.
- e) clone github repository using following command:

```
git clone https://github.com/vdevaram/oneapi_workshop.git
```

### Exercise 0

#### Environment set up for Opensource(Stock) Tensorflow and Openvino Benchmarking

- a) Open the “`ex0_environment_setup.ipynb`” and activate Python 3.7(oneAPI) kernel.
- b) Create a virtual environment “`opentfov`” for Opensource(Stock) Tensorflow and OpenVINO Benchmarks.

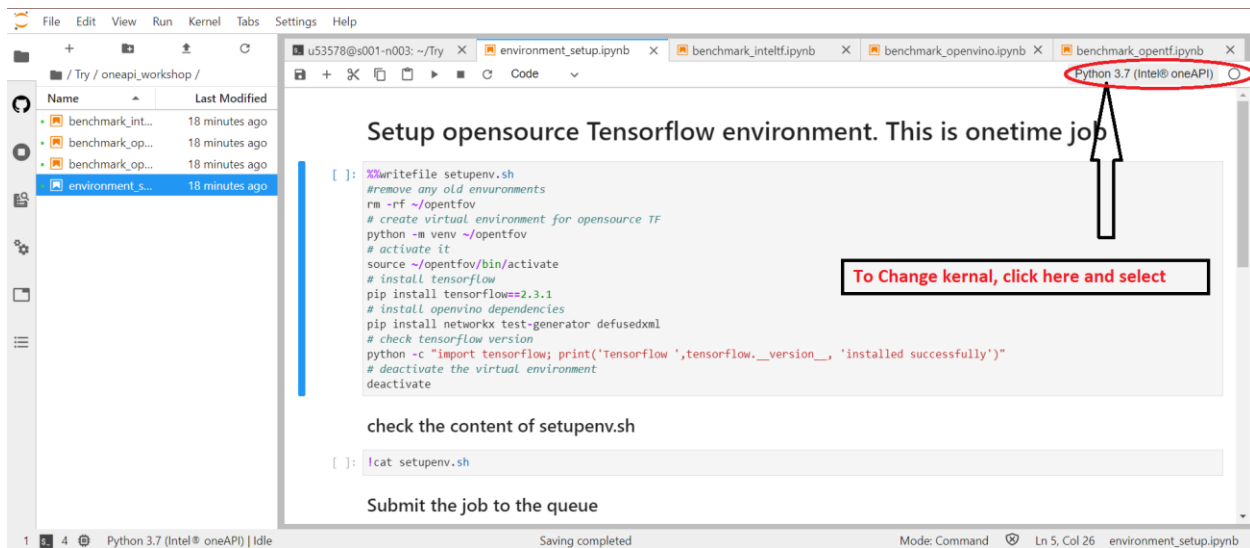


Figure: How To change the kernel

## Exercise 1

### Running the Resnet-50 Training & Inference Benchmarks with Opensource(Stock) Tensorflow

- Open the “[ex1\\_benchmark\\_opentf.ipynb](#)” and activate Python 3.7(oneAPI) kernel.
- Download Tensorflow Benchmarks from Github Repository.
- Activate the virtual environment “opentfov”.
- Submit the job containing the commands to run training and inference benchmarks.
- After the job finishes running, note down the training and inference throughput values(images/sec).

## Exercise 2

### Running the Resnet-50 Training & Inference Benchmarks with Intel Optimized Tensorflow

- a) Open the “[ex2\\_benchmark\\_inteltf.ipynb](#)” and activate Tensorflow(AI kit) kernel.
- b) For Intel-Tensorflow, No need to create an environment and install any dependency as it's part of the pre-installed AI toolkit.
- c) Submit the job containing the commands to run training and inference benchmarks.
- d) After the job finishes running, note down the training and inference throughput values(images/sec).

## Exercise 3

### Running the Resnet-50 Inference Benchmarks with OpenVINO

- a) Open the “[ex3\\_benchmark\\_openvino-rn50.ipynb](#)” and activate Python 3.7(oneAPI) kernel.
- b) Download the pre-trained Resnet-50 model and Convert it to Intermediate Representation(IR having .xml and .bin files) using Model Optimizer(MO) and Run the Inference Benchmarks.
- c) Submit the job containing the commands to download, convert the downloaded model, and run inference benchmarks for different parameters.
- d) After the job finishes running, note down the inference throughput values(FPS) for different parameters.

## Exercise 4

### Running the Bert(FP32 & Int8 precision) Inference Benchmarks

- a) Open the “[ex4\\_benchmark\\_openvino-bert.ipynb](#)” and activate Python 3.7(oneAPI) kernel.
- b) Download the pre-trained Resnet-50 model and Convert it to Intermediate Representation(IR having .xml and .bin files) using Model Optimizer(MO) and Run the Inference Benchmarks.
- c) Submit the job containing the commands to download, convert the downloaded model, and run inference benchmarks for different parameters.
- d) After the job finishes running, note down the inference throughput values(FPS) for different parameters.
- e) Download IR(.xml and .bin) directly for BERT(FP32 & Int8) model.
- f) Download vocab file for words interpretation from link:
- g) Do the benchmarking. Note the throughput & Latency differences of FP32 and Int8 model.

## Exercise 5

### Question-Answering Demo with Bert model

- a) Open the “[ex5\\_benchmark\\_openvino-bert\\_demo.ipynb](#)” and activate Python 3.7(oneAPI) kernel.
- b) A “[questions.txt](#)” file needs to be created for questions.
- c) Write your question questions.txt cell.
- d) To run a Question-Answering demo, you can use python\_demos available with OpenVINO using help:
- e) A successful run prompts user to enter questions and after entering, it will search whole text in the given link and give the answer with confidence metrics. Here we are providing questions using “[question.txt](#)” file.

## **Documentation Links:**

Intel DevCloud for oneAPI – Documentation

[https://devcloud.intel.com/oneapi/get\\_started](https://devcloud.intel.com/oneapi/get_started)

Intel oneAPI AI Analytics toolkit – Documentation

<https://software.intel.com/content/www/us/en/develop/tools/oneapi/ai-analytics-toolkit.html>

OpenVINO Documentation

<https://docs.openvinotoolkit.org/latest/index.html>

## **Feedback**

**If you have any feedback on this document or ideas to improve it, please write an email to any of us at:**

[vinod.devarampati@intel.com](mailto:vinod.devarampati@intel.com)

[lakshminarasimhan.ranganathan@intel.com](mailto:lakshminarasimhan.ranganathan@intel.com)

[aditya.sirvaiya@intel.com](mailto:aditya.sirvaiya@intel.com)