



AI ANALYTICS WORKSHOP DOC



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

17th June 2021

Jupyter Notebook Basics(to read)

- 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.

Clone github repository using following command:

```
git clone https://github.com/asirvaiy/AI-Workshop.git
```

Move all the files from AI-Workshop to home:

```
mv ~/AI-Workshop/* ~/
```

Exercise 0

Environment set up for Intel-Tensorflow and Stock-Tensorflow

- a) Open the “[condaenvsetup.ipynb](#)” and activate Python 3.7(oneAPI) kernel.
- b) Run through the cells one by one, “qsub cell” submits the job to do all the setup. Now running “qstat” will show 2 jobs running.
- c) Once qstat shows only one job(for Jupiter) setupenv.sh job has finished running and Setup should be completed.
- d) Following are the steps involved in setup.

Stock TensorFlow setup steps

- a. Create conda env:
`a. conda create -n stock-tensorflow python matplotlib ipykernel psutil pandas gitpython`
- b. Activate the created conda env:
`source activate stock-tensorflow`
- c. Install stock Tensorflow with a specific version:
`pip install tensorflow==2.3.0`
- d. Install extra needed package:
`pip install cxxfilt`
- e. Deactivate conda env:
`conda deactivate`
- f. Register the kernel to Jupyter NB:
`~/conda/envs/stock-tensorflow/bin/python -m ipykernel install --user --name=stock-tensorflow`
- g. NOTE: Please change the python path if you have a different folder path for anaconda3. After profiling, users can remove the kernel from Jupyter NB with `\$jupyter kernelspec uninstall stock-tensorflow`

Intel TensorFlow setup steps

NOTE: Intel-optimized Tensorflow is on DevCloud. However, users don't have access to install extra packages. Therefore, we need to clone Intel Tensorflow into the user's home directory for installing extra packages.

- a. Source oneAPI environment variables:

```
source /opt/intel/inteloneapi/setvars.sh
```

- b. Create conda env:

```
conda create --name intel-tensorflow --clone tensorflow
```

- c. Activate the created conda env:

```
source activate intel-tensorflow
```

- d. Install extra needed package:

```
pip install cxxfilt matplotlib ipykernel psutil pandas gitpython
```

- e. Deactivate conda env:

```
conda deactivate
```

- f. Register the kernel to Jupyter NB:

```
~/.conda/envs/intel-tensorflow/bin/python -m ipykernel install --  
user --name=intel-tensorflow
```

- g. NOTE: Please change the python path if you have a different folder path for anaconda3. After profiling, users can remove the kernel from Jupyter NB with `$jupyter kernelspec uninstall intel-tensorflow`

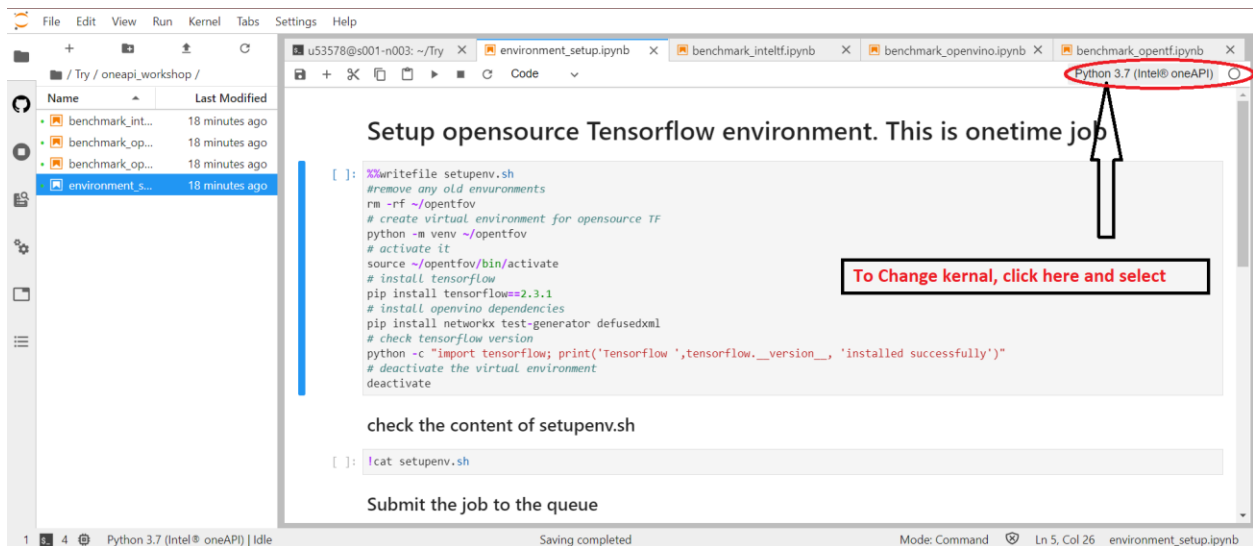


Figure: How To change the kernel

Exercise 1

Running the Census oneDAL notebooks

- Open the “[census_without_onedal.ipynb](#)” and activate Python 3.7(oneAPI) kernel. Run through all the cells. This notebook is without using scikit-learn from oneDAL.
- Open the “[census_with_onedal.ipynb](#)” and activate Python 3.7(oneAPI) kernel. Run through all the cells. This notebook is with using scikit-learn from oneDAL.
- In “[census_with_onedal.ipynb](#)” notebook, there is time comparison to show the speedup.

Exercise 2

Intel-TF improvements over Stock-TF

- a. Open the terminal and Clone Intel Model Zoo:
`git clone https://github.com/IntelAI/models.git`
- b. Browse to the perf_analysis folder:
`cd models/docs/notebooks/perf_analysis`
- c. Open the notebook benchmark_perf_comparison.ipynb
- d. Note: For "stock v.s. Intel Tensorflow" analysis type, please change your Jupyter notebook kernel to either "stock-tensorflow" or "intel-tensorflow" (highlighted in the diagram below)
- e. Run through every cell of the notebook one by one
- f. NOTE: For "stock vs. Intel Tensorflow" analysis type, in order to compare between stock and Intel-optimized TF results in section "Analyze TF Timeline results among Stock and Intel Tensorflow", users need to run all cells before the comparison section with both stock-tensorflow and intel-tensorflow kernels.
- g. Open the 2nd notebook benchmark_perf_timeline_analysis.ipynb.
- h. Run through every cell of the notebook one by one to get the analysis result.
- i. NOTE: There is no requirement for the Jupyter kernel when users run the 2nd notebook to analysis performance in detail.

Documentation Links:

Intel DevCloud for oneAPI – Documentation

https://devcloud.intel.com/oneapi/get_started

Intel oneAPI AI Analytics toolkit – Documentation and Samples

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

<https://github.com/oneapi-src/oneAPI-samples/tree/master/AI-and-Analytics>

Tensorflow Performance Analysis

https://github.com/oneapi-src/oneAPI-samples/tree/master/AI-and-Analytics/Features-and-Functionality/IntelTensorFlow_PerformanceAnalysis

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

lakshminarasimhan.ranganathan@intel.com

Kavita.aroor@intel.com

jing.xu@intel.com

aditya.sirvaiya@intel.com