Hands-on Lab Description

CS-ML-00001 – Setup Machine Learning Running Environment on Linux

Category:

```
CS-ML: Machine Learning
```

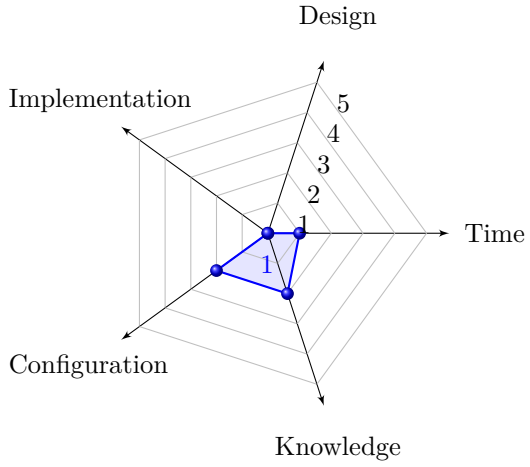
Objectives:

```
Learn how to set up Machine Learning (ML) data analysis framework such as Anaconda
Learn how to set up ML data analysis programming environment such as Python
```

Estimated Lab Duration:

```
Expert: 20 minutes
Novice: 100 minutes
```

Difficulty Diagram:



Difficulty Table.

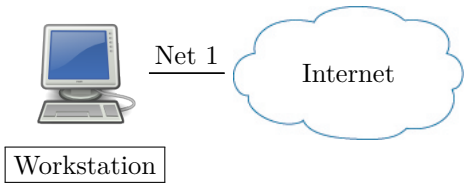
Measurements	Values (0-5)
Time	1
Design	0
Implementation	0
Configuration	2
Knowledge	2
Score (Average)	1

Required OS:

```
Linux: Ubuntu 18.04 LTS
```

Lab Running Environment:

```
VirtualBox https://www.virtualbox.org/ (Reference Labs: CS-SYS-00101)
```




```
1 Server: Linux (Ubuntu 18.04 LTS)
2 Network Setup: connected to the Internet
```

Lab Preparations:

```
Initial setup: basic Ubuntu 18.04 LTS is required for this lab
Basic Linux knowledge and operations. Reference Lab: CS-SYS-00001.
```

In this lab, we describe how to establish a Machine Learning (ML) environment on Linux-based Virtual Machines such as Ubuntu. The description includes the setup of Python, Machine and Deep Learning Package and create Virtual Environments. Additionally, we also describe how to setup GPU acceleration software if the system provides the GPU support. This lab describes the essential steps in creating a ML environment and there are many additional options for you to include, however this setup is simple and robust. We assume that basic Ubuntu OS has been installed. Basic steps include:

1. Create a virtual environment and install Python
2. Install Machine Learning packages

In addition, we will present how to install GPU acceleration software (CUDA and cuDNN)

Task 1 Install Python

First, you need to check current python version installed on your system:

```
$    python --version  % check if the system has installed version 2.7
$    python3 --version % check which version of python 3.x installed
```

```
$    sudo apt-install python3 python3-pip python3-env
```

By default the Linux Ubuntu distribution install python 3.6. In order to use a later version of python such as version 3.7, you can do follows:

1. Install python by typing:

   ```
   $       sudo apt update -y
   $       sudo apt install python3.7
   ```

2. Add python 3.6 and python 3.7 to update-alternatives:

   ```
   $       sudo update-alternatives --install /usr/bin/python3 python3
             /usr/bin/python3.6 1
   $       sudo update-alternatives --install /usr/bin/python3 python3
             /usr/bin/python3.7 2
   ```

3. Update python 3 to point to python 3.7.

   ```
   $       sudo update-alternatives --config python3
   ```

   You will see the following figure (Figure CS-ML-00001.1) and choose 0 or 2 in the example to select *python3.7* as the default python version corresponding to command *python3*:

4. Another approach is to use symbolic link to set up default running python version. You will need to update python3 to point to python3.7, and as a result */usr/bin/python3* is just a *symlink*. You can delete it and make a new *symlink* to *python3.7*:

   ```
        sudo rm /usr/bin/python3
        sudo ln -s python3.7 /usr/bin/python3
   ```

5. Now, you can check your *python3* version:

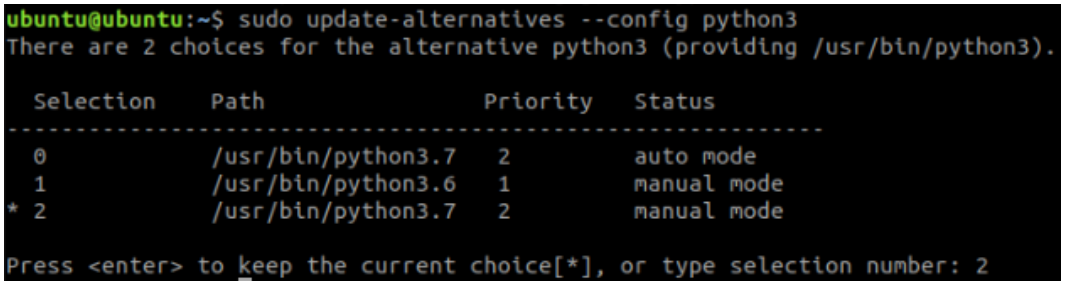


Figure CS-ML-00001.1
Assign proper version of python using *update-alternatives* command.

```
$       python3 --version
```

Task 2 Install Anaconda

Anaconda is an open-source package manager, environment manager, and distribution of the Python and R programming languages. Here, you are guided to install Anaconda on an Ubuntu 18.04 VM.

1. Retrieve the Latest Version of Anaconda: rrom a web browser, go to the Anaconda Distribution page, available via the following link: `https://www.anaconda.com/distribution/`. Find the latest Linux version and copy the link to the installer bash script. Then you can perform the following in your working folder:

   ```
   $       curl -O https://repo.anaconda.com/archive/Anaconda3-2019.10-Linux-x86_64.sh
   ```

   You can also run *wget* to download the same bash script.

2. Verify the Data Integrity of the Installer: ensure the integrity of the installer with cryptographic hash verification through SHA-256 checksum:

   ```
   $       sha256sum Anaconda3-2019.10-Linux-x86_64.sh
   ```

   It should show you responds and tell you if the checking past or not.

3. Run the Anaconda Script:

   ```
   $       bash Anaconda3-2019.10-Linux-x86_64.sh
   ```

   You may also change the script to executable by using *chmod* command and then run the script directly. You will receive the following output to review the license agreement by pressing ENTER until you reach the end. You should choose “yes” and enter.

   ```
   Welcome to Anaconda3 2019.10

   In order to continue the installation process, please review the license
   agreement.
   Please, press ENTER to continue
   >>>
   ...
   Do you approve the license terms? [yes|no]
   ```

Once you agree to the license, you will be prompted to choose the location of the installation. You can press ENTER to accept the default location, or specify a different location. for example, you may install Anaconda3 into this location: */home/ubuntu/anaconda3*, where ubuntu is the user name in the system.

4. Once installation is complete, you'll receive the following output, and it is recommended that you type yes to use the conda command.:

```
...
installation finished.
Do you wish the installer to prepend the Anaconda3 install location
to PATH in your /home/sammy/.bashrc ? [yes|no]
[no] >>>
```

5. Activate Installation: you can now activate the installation with the following command:

```
$       source ~/.bashrc
```

6. Now you will see *(base)* in front of your command line prompt. Now, you can test the Installation by using the *conda* command to test the installation and activation:

```
$       conda list
```

You'll receive output of all the packages you have available through the Anaconda installation.

7. To control whether or not each shell session has the base environment activated or not, you can run:

```
$       conda config --set auto_activate_base True.
```

8. Set Up Your Own Anaconda Environments: you can create Anaconda environments with the *conda create* command. For example, a Python 3 environment named *my_env* can be created with the following command:

```
$       conda create --name my_env python=3
```

9. Activate the new environment like so:

```
$       conda activate my_env
```

Then you will see the *(base)* is changed to *(my_env)* before your command line prompt, and you are now ready to begin work on a project. Note that after installed Anaconda3-2019.10, and setup the running environment, Anaconda will use the latest version python3.8 in the environment.

10. To update anaconda, you can run the following command to update anaconda files:

```
$       conda update conda -n root % When you run in user enviornment, you need to
          get the root mode for update and use the option -n root
$       conda update anaconda -n root
```

After this step, you should have most of popular anaconda packages in place. Then you can run various Anaconda graphic-based applications by running:

```
$       anaconda-navigator
```

You can also run graphic python app Spyder:

```
$       spyder
```

Note that if you cannot run those commands, you should check your running path, and you can add anaconda running path by using the following command:

```
$       export PATH=/home/yourUserName/anaconda3/bin:$PATH
```

Task 3 Install Machine Learning Modules

You need to install a few popular ML modules to run you data science related projects. Anaconda has the most popular ML data analysis and processing modules in place, such as *Numpy* and *Pandas*. You can go to `https://docs.anaconda.com/anaconda/packages/pkg-docs/` to check packages that included in Anaconda. However, it does not include *Kreas*.

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation. Use Keras if you need a deep learning library that:

- Allows for easy and fast prototyping (through user friendliness, modularity, and extensibility).
- Supports both convolutional networks and recurrent networks, as well as combinations of the two.
- Runs seamlessly on CPU and GPU.

Task 4 Machine Learning GUI App

Anaconda provides a GUI called *anaconda-navigator* for managing several GUI-based ML applications. In order to install anaconda-navigator, after running anaconda Bash installation script, you can run the following commands:

```
$    conda update conda
$    conda update anaconda
```

Then, you can run:

```
$    source .bashrc % load the bash shell that has set up the anaconda running
       environment
$    anaconda-navigator
```

After anaconda-navigator launched, you can see several icons such as JupyterLab, Jupyter Notebook, Spyder, Orange 3, RStudio, VS Code, etc. Then you can click on Spyder to launch the GUI app for python programming and running environment. You can also run Spyder directly from the command line:

```
$    spyder
```

Now, you can open any python file from the GUI app. Once the file has been loaded, please make sure you set the console working directory to the path this file is shown in Figure CS-ML-00001.2. This way you do not have to give the entire path, just the relative path of the dataset file.

Now, you can run your python code by pressing F5 or click on the green triangle button and see the running progress on the right-lower side of console of the spyder app.

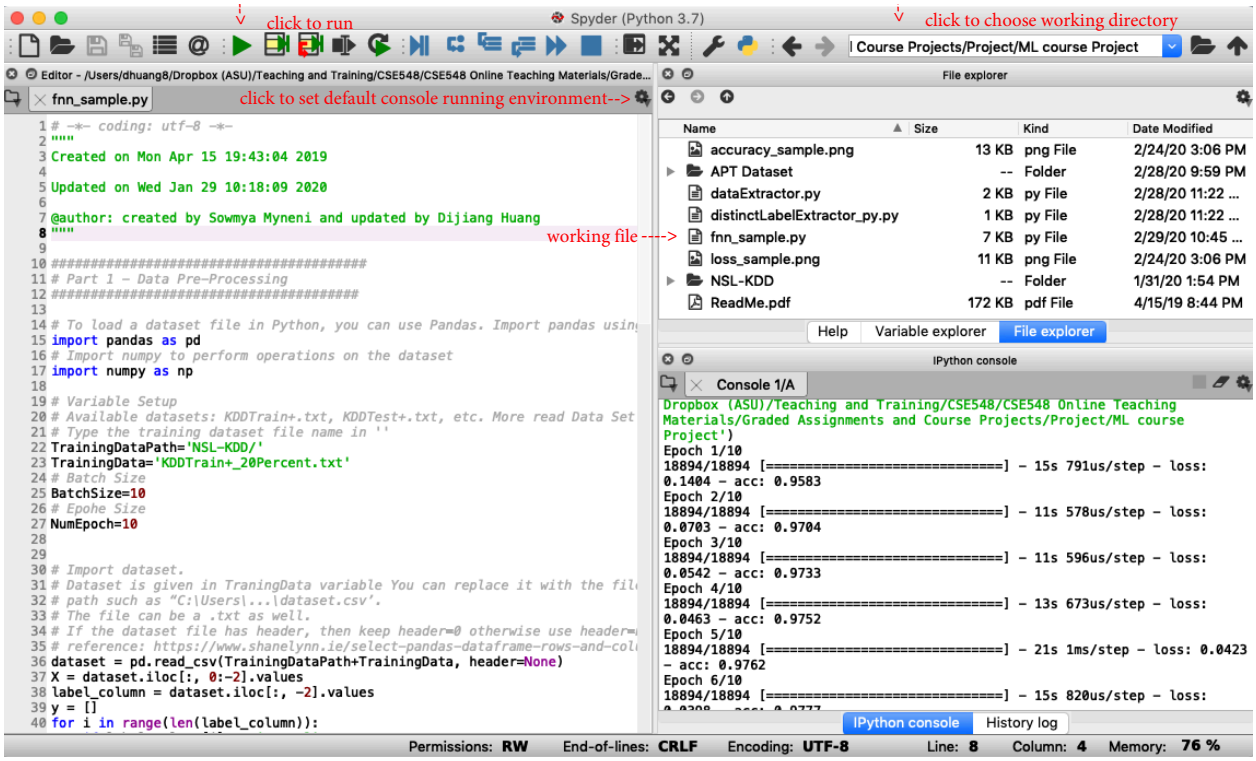


Figure CS-ML-00001.2
Spyder App.

Related Information and Resource

```
Python: https://www.python.org/
Anaconda: https://www.anaconda.com/
Spyder: https://www.spyder-ide.org/
```