

## TASK#1

# Creating the ML model in Docker Container



- May 30, 2021



This blog shows in a gradual manner, how to train and run a machine learning model over a docker container in a simple and efficient way.

## What exactly is a Docker Container?

A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools,

system libraries and settings.

Container images become containers at runtime and in the case of Docker containers - images become containers when they run on Docker Engine. Available for both Linux and Windows-based applications, containerized software will always run the same, regardless of the infrastructure. Containers isolate software from its environment and ensure that it works uniformly despite differences for instance between development and staging.

For this task we need to follow these steps:-

1.To install docker in our base OS we need to use "yum" and we need to be logged into our root account.

```
[root@localhost anib]# yum install docker-ce --nobest -y
```

Now we need to launch our Docker by using command "systemctl start docker".

We can check the status of program by using "systemctl status docker"

```
[root@localhost anib]# systemctl start docker
[root@localhost anib]# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)
   Active: active (running) since Sun 2021-05-30 15:43:02 IST; 9s ago
     Docs: https://docs.docker.com
Main PID: 2826 (dockerd)
   Tasks: 22
  Memory: 120.7M
 CGroup: /system.slice/docker.service
         ├─2826 /usr/bin/dockerd -H fd://
         ├─2844 containerd --config /var/run/docker/containerd/containerd.toml --log-level info
...
May 30 15:43:01 localhost.localdomain dockerd[2826]: time="2021-05-30T15:43:01.427079565+05:30" level=info msg="Graph >
May 30 15:43:01 localhost.localdomain dockerd[2826]: time="2021-05-30T15:43:01.427817947+05:30" level=warning msg="You>
May 30 15:43:01 localhost.localdomain dockerd[2826]: time="2021-05-30T15:43:01.427844143+05:30" level=warning msg="You>
May 30 15:43:01 localhost.localdomain dockerd[2826]: time="2021-05-30T15:43:01.428623977+05:30" level=info msg="Loadin>
May 30 15:43:02 localhost.localdomain dockerd[2826]: time="2021-05-30T15:43:02.386174456+05:30" level=info msg="Defaul>
May 30 15:43:02 localhost.localdomain dockerd[2826]: time="2021-05-30T15:43:02.663811596+05:30" level=info msg="Loadin>
May 30 15:43:02 localhost.localdomain dockerd[2826]: time="2021-05-30T15:43:02.729455609+05:30" level=info msg="Docker>
May 30 15:43:02 localhost.localdomain dockerd[2826]: time="2021-05-30T15:43:02.729791685+05:30" level=info msg="Daemon>
May 30 15:43:02 localhost.localdomain dockerd[2826]: time="2021-05-30T15:43:02.746378556+05:30" level=info msg="API li>
May 30 15:43:02 localhost.localdomain systemd[1]: Started Docker Application Container Engine.
Lines 1-21/21 [END]
```

2. Now we need to pull image of CentOS from docker hub over internet.

```
[root@localhost anib]# docker pull centos
Using default tag: latest
latest: Pulling from library/centos
Digest: sha256:5528e8b1b1719d34604c87e11dc1c0a20bedf46e83b5632cdeac91b8c04efc1
Status: Image is up to date for centos:latest
docker.io/library/centos:latest
[root@localhost anib]#
```

Since I have already installed CentOS so it shows that the image is up to date.

You can also use other OS images to perform this task.

3. We need to launch our docker container now with name "sys1" or you can give any other name as well.

```
File Edit View Search Terminal Help
[root@localhost anib]# docker run -it --name sys1 centos
[root@f6cba79eacb3 /]#
```

Here:-

- it: "interactive terminal"
- name: providing a name to the container
- it: "interactive terminal"
- name: providing a name to the container

4. Now we need to install python and some packages inside our docker container.

Commands:

- yum install python3
- pip3 install pandas
- pip3 install scikit-learn

```
root@f6cba79eacb3:/# yum install python3
Failed to set locale, defaulting to C.UTF-8
centos Linux 4 - AppStream          262 kB/s | 6.3 MB  00:24
centos Linux 4 - BaseOS             74 kB/s | 2.3 MB  00:31
centos Linux 4 - Extras              6.2 kB/s | 9.4 kB  00:01
Dependencies resolved.
=====
Package           Arch      Version       Repo      Size
=====
Installing:
python3           x86_64    3.6.8-2.module_el8.3.0+962+49628269 appstream 19 k
Installing dependencies:
platform-python3-gip          search 9.6.3-18.el8          bases   1.7 M
python3-pip            search 9.0.3-18.el8          appstream 20 k
python3-setuptools         search 39.2.0-6.el8          bases   183 k
python3               3.6
=====
Transaction Summary
=====
Install 4 Packages

Total download size: 1.9 M
Installed size: 7.4 M
Is this ok [y/N]: y
Unloading Packages:
 4/4: python3-pip-9.0.3-18.el8.noarch.rpm      22 kB/s | 20 kB  00:00
```

```
[root@localhost ~]# python3.6-3.6.0-2.module_ell8.3.0+503+e102 9.2 kB/s | 29 kB 00:02
[4/4]: python3-setuptools-38.2.0-6.el8.noarch.r 40 kB/s | 163 kB 00:04
[root@localhost ~]# platform-python-pip-9.0.0-18.el8.noarch.rpm
total 37 kB/s | 1.7 kB 00:46
[root@localhost ~]# rpm -Uvh /var/cache/rpm/appstream-82e86d1c97eb0530/packages/python3-pip-9.0.0-18.el8.noarch.rpm: Header V3 RSA/SHA256 Signature, key ID 0483c65d: NOKEY
Importing GPG key 0483c65d:
Userid : CentOS Official Signing Key <security@centos.org>
Fingerprint: 9008 74fa 8307 cb22 7f86 4882 6085 3383 8481 c650
Is this ok [y/N]: y
Key imported successfully
Running transaction check
```

```
[root@localhost ~]# pip3 install pandas
Collecting pandas
  Using cached https://files.pythonhosted.org/packages/c3/e2/09ecacea7ba8071c789293f99ad84ca3189952f664bca9559ed83870d4d/pandas-1.1.5-cp36-cp36-manylinux1_x86_64.whl (9.3MB)
    100% |██████████| 9.3MB 888B/s
Collecting pytz==2019.2 (from pandas)
  Using cached https://files.pythonhosted.org/packages/78/94/704179ca5d00892a98d113cd4933372924c040d0040e37ca7a76857f99ca6/pytz-2019.2-py2.py3-none-any.whl (514kB)
    100% |██████████| 512kB 2644B/s
Collecting numpy==1.19.5 (from pandas)
  Using cached https://files.pythonhosted.org/packages/45/b2/6c7545bb7a7a0375463848c7696884ub947328135681b13bea692c3aa3/numpy-1.19.5-cp36-cp36-manylinux1_x86_64.whl (13.4MB)
    100% |██████████| 13.4MB 438B/s
Collecting python-dateutil==2.7.3 (from pandas)
  Using cached https://files.pythonhosted.org/packages/09/04/w7c31aebd875f2a88891b884cf2dc52df9205a0150678180cf25c93da711/six-1.16.0-py2.py3-none-any.whl (227kB)
    100% |██████████| 225kB 3000B/s
Collecting six==1.5.2 (from python-dateutil==2.7.3->pandas)
  Using cached https://files.pythonhosted.org/packages/64/79/d59450c3d40ef87588824287ae0907098a60306af2bce54034479415cb/python_dateutil-2.7.3-py3-none-any.whl
Installing collected packages: pytz, numpy, six, python-dateutil, pandas
Successfully installed numpy-1.19.5 pandas-1.1.5 python-dateutil-2.7.3 pytz-2019.2 six-1.16.0
[root@localhost ~]#
```

```
[root@localhost ~]# pip3 install scikit-learn
Collecting scikit-learn
  Using cached https://files.pythonhosted.org/packages/f5/ef/bcd79e6d9259d95a479a812996c8405b4e42b3be5a99639543464db1271a/scikit_learn-0.24.2-cp36-cp36-manylinux1_x86_64.whl (20.09B)
    100% |██████████| 20.09B 17kB/s
Collecting scipy==0.19.1 (from scikit-learn)
  Using cached https://files.pythonhosted.org/packages/c8/09/03171228d5ced149f5ced50305c89e8576fc025a98058fe5b4462b91bc2/scipy-0.19.1-cp36-cp36-manylinux1_x86_64.whl (25.09B)
Collecting threadpoolctl==2.1.0 (from scikit-learn)
  Using cached https://files.pythonhosted.org/packages/f7/12/e3ff2e200fa394a14991729057axa4affc50c30a2c1c297a88d33f133/threadpoolctl-2.1.0-py3-none-any.whl
Requirement already satisfied: numpy<1.19.3 in /usr/local/lib64/python3.6/site-packages (from scikit-learn)
Collecting joblib==0.11 (from scikit-learn)
  Using cached https://files.pythonhosted.org/packages/55/85/79c6440b7e70d9ef73da4f4a7847e986535c355a4cd4f71b97a3549697/joblib-0.11.0-py3-none-any.whl (30.93B)
Installing collected packages: scipy, threadpoolctl, joblib, scikit-learn
Successfully installed joblib-0.11.0 scikit-learn-0.24.2 scipy-0.19.1 threadpoolctl-2.1.0
[root@localhost ~]#
```

5. Here we need to copy the dataset from our system(windows) to the redhat8 baseOS from which we will copy the dataset to our CentOS.

I am using Winscp application for copying the data.

In the Host name enter the ip of your baseOS(redhat8), in user name enter "root" and in password enter your root password. Then login and drag and drop your files from your system to your baseOS in virtual box.

Now to copy the dataset to your container:-

```
[root@localhost ~]# docker cp /root/Salary_Data.csv sys1:/root/dataset
[root@localhost ~]# docker cp /root/marksLR.csv sys1:/root/dataset
[root@localhost ~]#
```

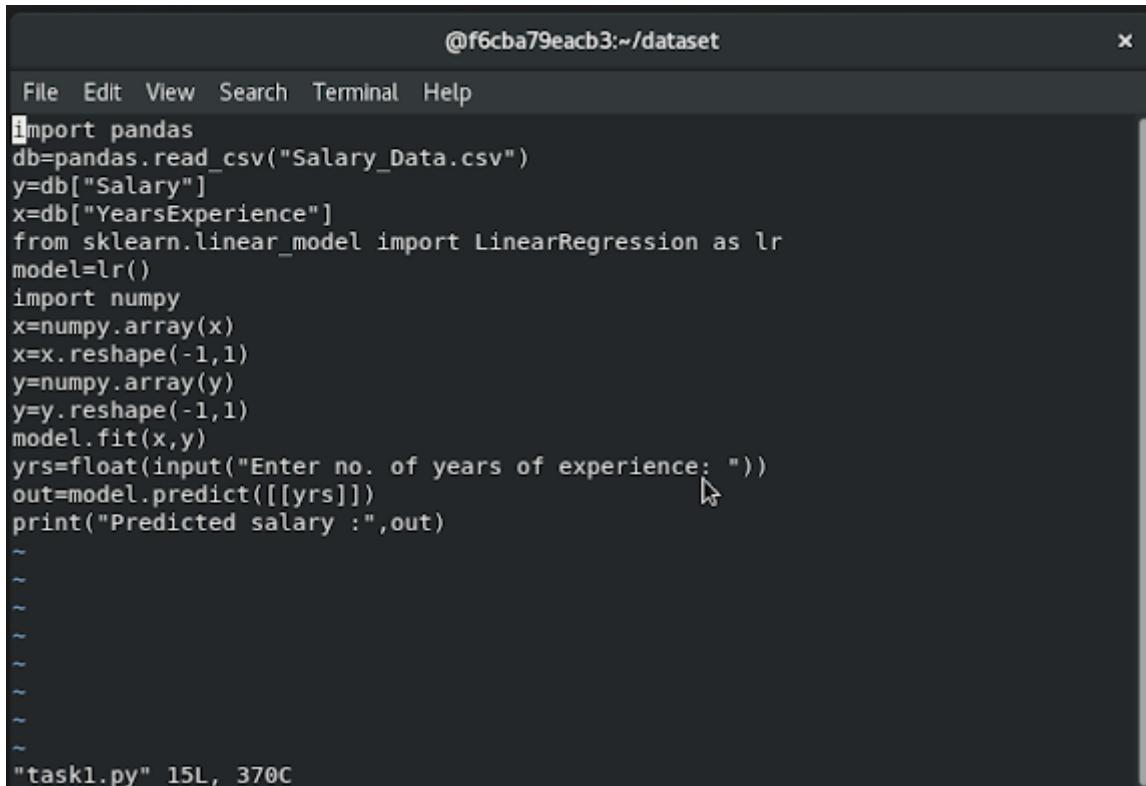
"dataset" is the directory which I have already created in my container using command "mkdir dataset".

6. Now all that remains is to write the python code.

Using command "vi task1.py" open a file in editor inside your container and enter "i"

for insertion of code.

Code:



The screenshot shows a terminal window titled '@f6cba79eacb3:~/dataset'. The window contains a menu bar with File, Edit, View, Search, Terminal, and Help. Below the menu is a block of Python code. The code imports pandas, reads a CSV file named 'Salary\_Data.csv', and performs linear regression using the 'LinearRegression' class from 'sklearn.linear\_model'. It prompts the user for years of experience and prints the predicted salary. The code is as follows:

```
import pandas
db=pandas.read_csv("Salary_Data.csv")
y=db["Salary"]
x=db["YearsExperience"]
from sklearn.linear_model import LinearRegression as lr
model=lr()
import numpy
x=numpy.array(x)
x=x.reshape(-1,1)
y=numpy.array(y)
y=y.reshape(-1,1)
model.fit(x,y)
yrs=float(input("Enter no. of years of experience: "))
out=model.predict([[yrs]])
print("Predicted salary :",out)
```

At the bottom of the terminal window, it shows the file name "task1.py" and its statistics: 15L, 370C.

7. Now we need to run our code and enter some input in order to test our code.

```
[root@f6cba79eacb3 dataset]# vi task1.py
[root@f6cba79eacb3 dataset]# python3 task1.py
Enter no. of years of experience: 4.5
Predicted salary : [[68317.03064522]]
```

We can observe that our code is working!

LinkedIn:- <https://www.linkedin.com/in/anirudh-b-2807b61b1>