

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 driver: overlay2"
May 30 15:43:01 localhost.localdomain dockerd[2826]: time="2021-05-30T15:43:01.427817947+05:30" level=warning msg="You are not running the Docker CLI on this host, and will not be able to use it. To resolve this, you have to run the Docker CLI on this host."
May 30 15:43:01 localhost.localdomain dockerd[2826]: time="2021-05-30T15:43:01.427844143+05:30" level=warning msg="You are not running the Docker CLI on this host, and will not be able to use it. To resolve this, you have to run the Docker CLI on this host."
May 30 15:43:01 localhost.localdomain dockerd[2826]: time="2021-05-30T15:43:01.428623977+05:30" level=info msg="Loading containers: start."
May 30 15:43:02 localhost.localdomain dockerd[2826]: time="2021-05-30T15:43:02.386174456+05:30" level=info msg="Default bridge (docker0) is already up. To create new bridges, use 'docker network create'. If you don't want it, you can remove it with 'docker network rm docker0'."
May 30 15:43:02 localhost.localdomain dockerd[2826]: time="2021-05-30T15:43:02.663811596+05:30" level=info msg="Loading containers: done."
May 30 15:43:02 localhost.localdomain dockerd[2826]: time="2021-05-30T15:43:02.729455609+05:30" level=info msg="Docker daemon is ready to accept requests from the client."
May 30 15:43:02 localhost.localdomain dockerd[2826]: time="2021-05-30T15:43:02.729791685+05:30" level=info msg="Daemon listening on socket /var/run/docker.sock"
May 30 15:43:02 localhost.localdomain dockerd[2826]: time="2021-05-30T15:43:02.746378556+05:30" level=info msg="API listen on /var/run/docker.sock"
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: sha256:5528e8b1b1719d34604c07e11dc1c0a20bedf46e83b5632cdeac91b8c04efc1
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@f6c8a79eacb3 /]#
```

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@f6c8a79eacb3 /]# yum install python3
failed to set locale, defaulting to C.UTF-8
centos Linux 8 - AppStream 262 kB/s | 6.3 MB 00:24
centos Linux 8 - BaseOS 74 kB/s | 2.3 MB 00:31
centos Linux 8 - 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+562+102826a appstream 19 k
installing dependencies:
platform-python-pip x86_64 9.0.3-18.el8 baseos 1.7 M
python3-pip x86_64 9.0.3-18.el8 appstream 20 k
python3-setuptools x86_64 39.2.0-6.el8 baseos 105 k
installing module streams:
python3 3.6
=====
Transaction Summary
-----
Install 4 Packages
Total download size: 1.9 M
Installed size: 7.6 M
Is this ok [y/N]: y
Downloading Packages:
1/4: python3-pip-9.0.3-18.el8.x86_64.rpm 22 kB/s | 20 kB 00:00
```

```

2/4): python3-3.8.4-3.module.el8_8-56.el8_8.2 0.2 MB/s | 19 kB 00:02
3/4): python3-setuptools-39.2.0-6.el8.noarch.r 40 MB/s | 162 kB 00:04
4/4): platform-python-pip-9.0.3-18.el8.noarch.r
-----
37 MB/s | 1.7 MB 00:46
Warning: /var/cache/dnf/appstream-0266d1c976b532/packages/python3-pip-9.0.3-18.el8.noarch.rpm: Header V3 RSA/SHA256 Signature, key ID 0483C654: NOKEY
Total
CentOS Linux 8 - AppStream
Reporting GPG key 0483C654:
Userid : "CentOS Linux 8 official signing key (<security@centos.org>)"
Fingerpr: 9008 74FA 83D7 C822 7F86 4862 0505 5503 0483 C650
From : /etc/pki/rpm-gpg/RPM-GPG-KEY-centosofficial
P. This ok [y/N]: y
My imported successfully
Running transaction check

```

```

Complete!
[root@fcb47eeccb3 ~]# pip3 install pandas
WARNING: Running pip install with root privileges is generally not a good idea. Try 'pip3 install --user' instead.
Collecting pandas
  Downloading https://files.pythonhosted.org/packages/c3/e2/09cacecafbab71c78393f0a6d84ca3105952f66d9bcad586e68379d4d/pandas-1.1.5-cp38-cp38m-manylinux_x86_64.whl (9.5MB)
    100% |#####| 9.5MB 888B/s
Collecting pytz>=2017.2 (from pandas)
  Downloading https://files.pythonhosted.org/packages/78/94/794179ca5d0892a8ef131c4f9233726240c04b0d040be77ca78b5f99dc96/pytz-2021.1-py2.py3-none-any.whl (516kB)
    100% |#####| 512kB 244kB/s
Collecting numpy>=1.15.4 (from pandas)
  Downloading https://files.pythonhosted.org/packages/45/b2/6c7545bb7a38754663848c70968944b0947328125d81bf13bea092c3ae3/numpy-1.19.5-cp38-cp38m-manylinux_x86_64.whl (13.4MB)
    100% |#####| 13.4MB 438kB/s
Collecting python-dateutil>=2.7.3 (from pandas)
  Downloading https://files.pythonhosted.org/packages/44/76/d66450c3d548ef87588824287ae897090de0b386ef2bce5d34d78d15ch/python-dateutil-2.8.1-py2.py3-none-any.whl (227kB)
    100% |#####| 225kB 300kB/s
Collecting numpy>=1.15 (from python-dateutil>=2.7.3->pandas)
  Downloading https://files.pythonhosted.org/packages/49/54/47c31a8e07372604091b84cf29c52f792b5a015067818bcf25c3d0f11/six-1.16.0-py2.py3-none-any.whl
Collecting collected packages: pytz, numpy, six, python-dateutil, pandas
Successfully installed numpy-1.19.5 pandas-1.1.5 python-dateutil-2.8.1 pytz-2021.1 six-1.16.0
[root@fcb47eeccb3 ~]#

```

```

[root@fcb47eeccb3 ~]# pip3 install scikit-learn
WARNING: Running pip install with root privileges is generally not a good idea. Try 'pip3 install --user' instead.
Collecting scikit-learn
  Downloading https://files.pythonhosted.org/packages/f5/4f/bc6796d59250d90479b01296cf4605be42b3b050963934144a8bc171a/scikit_learn-0.24.2-cp38-cp38m-manylinux_x86_64.whl (20.0MB)
    100% |#####| 20.0MB 17MB/s
Collecting scipy>=0.19.1 (from scikit-learn)
  Downloading https://files.pythonhosted.org/packages/c8/89/4b317228c5cd140f3cd58305c8068576ffcd925a938f63bb001010c2/scipy-1.5.4-cp38-cp38m-manylinux_x86_64.whl (25.9MB)
    100% |#####| 25.9MB 24MB/s
Collecting threadpoolctl>=2.0.0 (from scikit-learn)
  Downloading https://files.pythonhosted.org/packages/f7/12/ec3f2e203afa394a149811728357aa48efc59c20a2c1c287a08f33f133/threadpoolctl-2.1.0-py3-none-any.whl
Requirement already satisfied: numpy>=1.13.3 in /usr/local/lib64/python3.8/site-packages (from scikit-learn)
Collecting joblib>=0.11 (from scikit-learn)
  Downloading https://files.pythonhosted.org/packages/55/85/78c6402b67b0c9e6f3d4f467847e066525c355a4dc04f73b97a35e9067/joblib-1.0.1-py3-none-any.whl (100kB)
    100% |#####| 100kB 147kB/s
Installing collected packages: scipy, threadpoolctl, joblib, scikit-learn
Successfully installed joblib-1.0.1 scikit-learn-0.24.2 scipy-1.5.4 threadpoolctl-2.1.0
[root@fcb47eeccb3 ~]#

```

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/marks1r.csv sys1:/root/dataset
[root@localhost ~]#

```

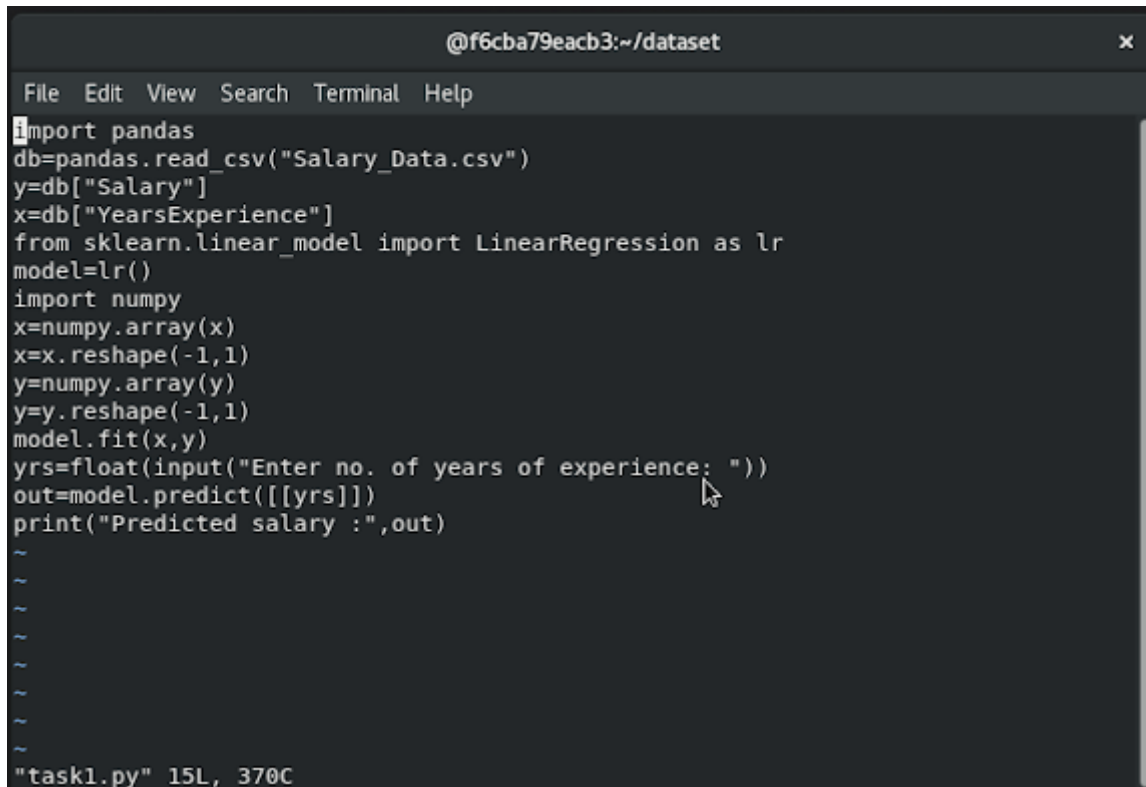
"dataset" is the directory which I have already created in my container using command "mkdir dataset".

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:



```
@f6cba79eacb3:~/dataset
File Edit View Search Terminal Help
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)
~
~
~
~
~
~
~
"task1.py" 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>