# Capstone Project-Interim Report

AIML March Group 3A- CV

10-Feb-2020

### Executive Summary

This project represents a culmination of the Ten modules of the AI and ML Specialization offered by Great Lakes Executive Learning and University of Texas at Austin via Great Learning.

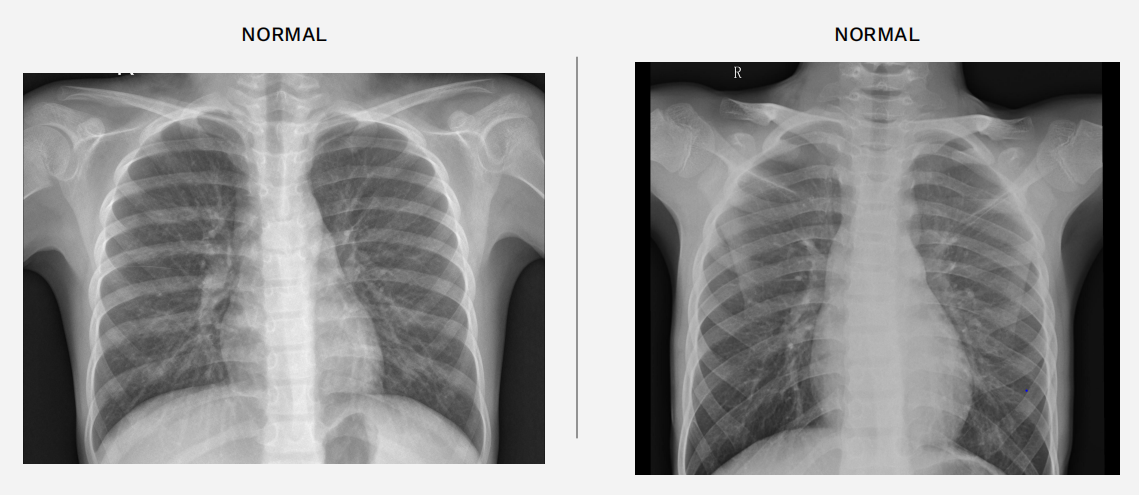
The Pneumonia Detection prediction model is built based on the basics of Computer Vision Technique techinques learned throughout the specialization.

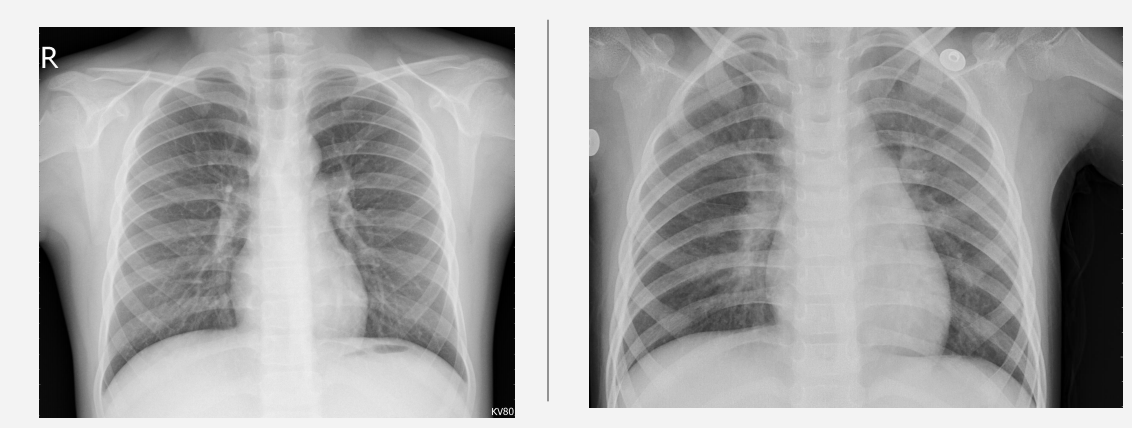
#### Sampling the Data

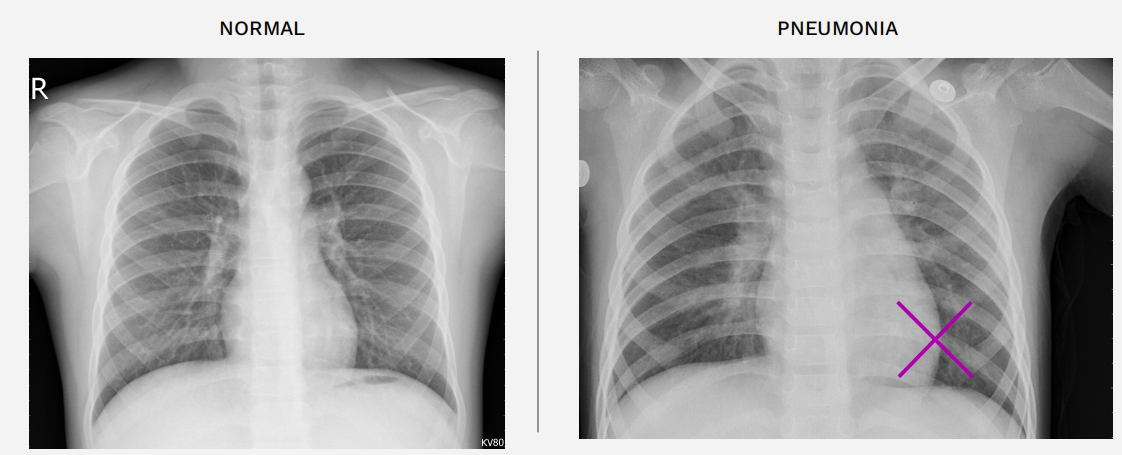
The corpora given comprises X-RAY of Lung images of very large datasets from Kaggle competition - of more than1000 and above DICOM images with file size of over 4 GB.

Kaggle dataset - quick look

* 5’863 X-Ray images
* pediatric patients 1-5 years old
* labeled by several specialists







In order to be accomodated within my system limitations (and in keeping with the approach recommended) a sample of the corpus was selected for study in order to build and train the prediction model.

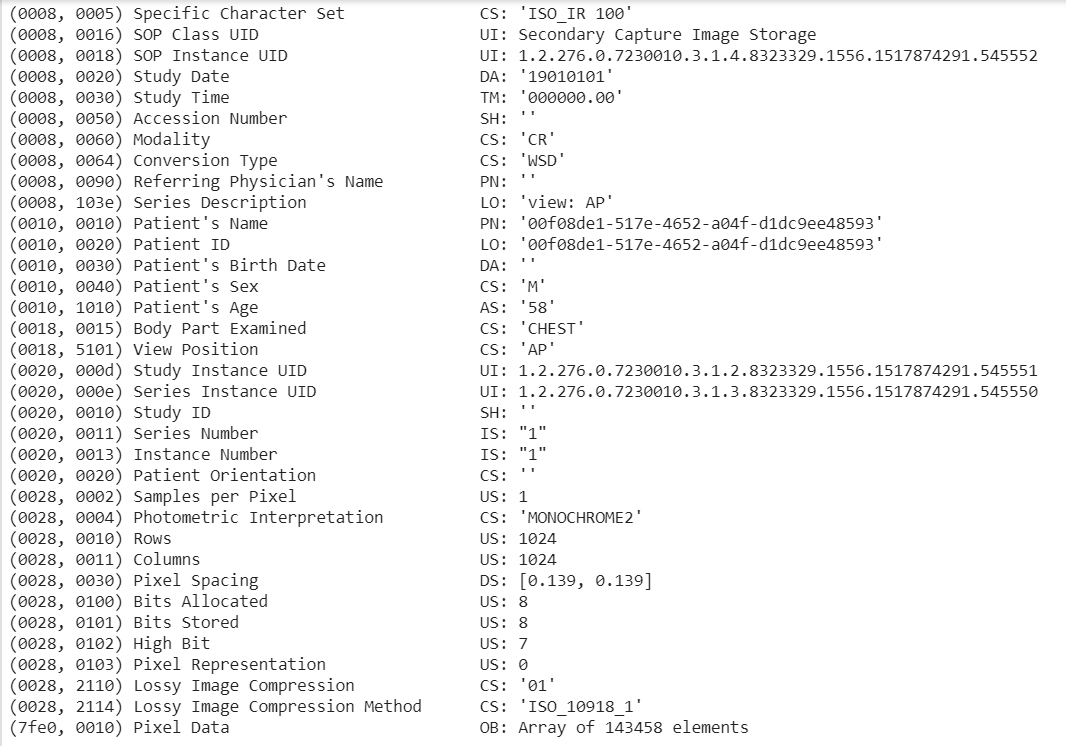
It is very important to understand the data in DICOM files before we work on Prediction Models.

import pydicom

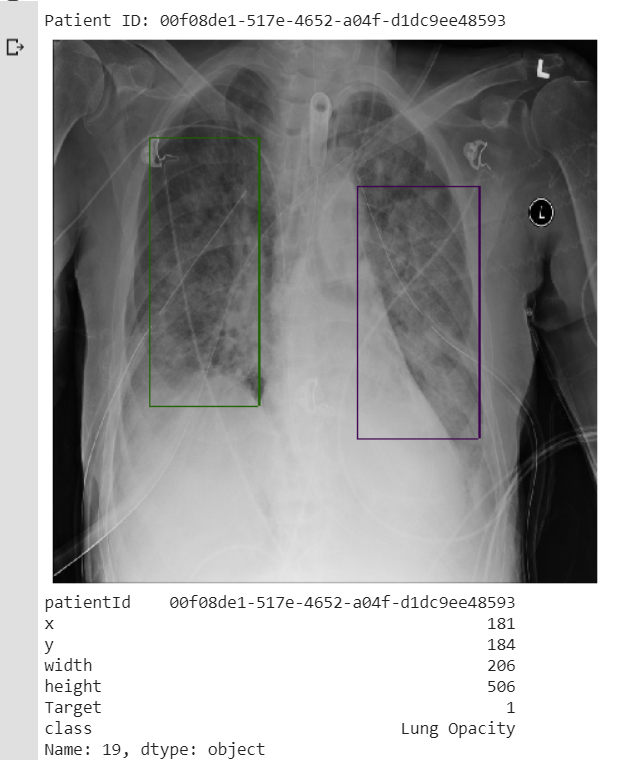
dcm\_file = '../content/RSNAdata/data/stage\_2\_train\_images/%s.dcm' % pat\_choose

dcm\_data = pydicom.read\_file(dcm\_file)

print(dcm\_data)

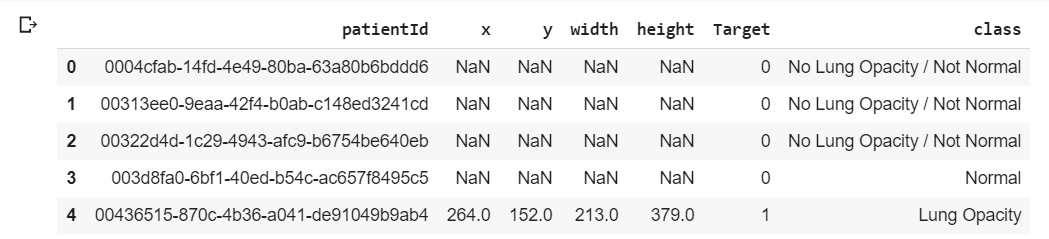


Understanding the data from the DICOM files is imperative to being able to ensure one's conceptualization of bounding boxes on the arrays from those files. We need to visualize those boxes in order to augment the knowledge regarding the visual aspects of pneumonia:

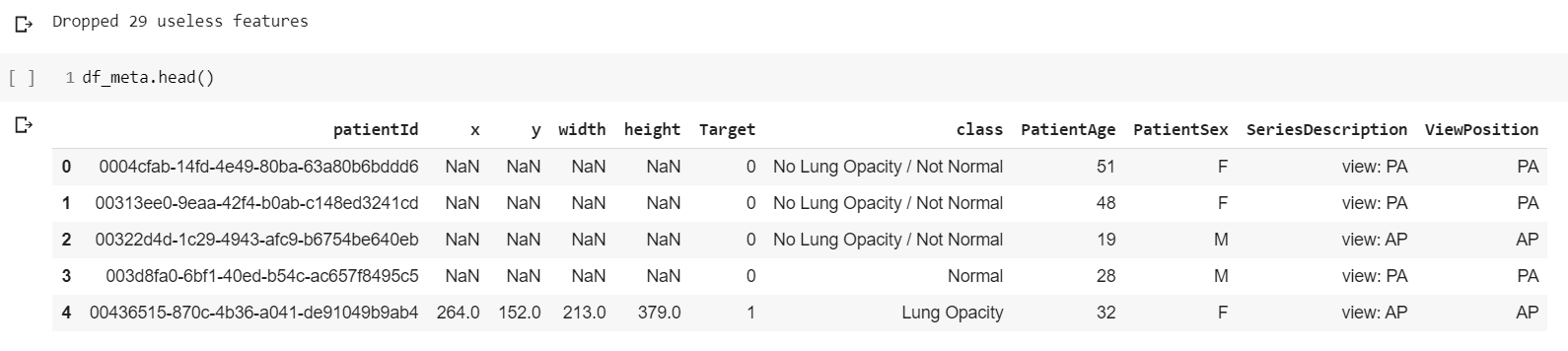


Now that we have visualized the bounding boxes and the XRAYs, we should take a look at the demographics of our features

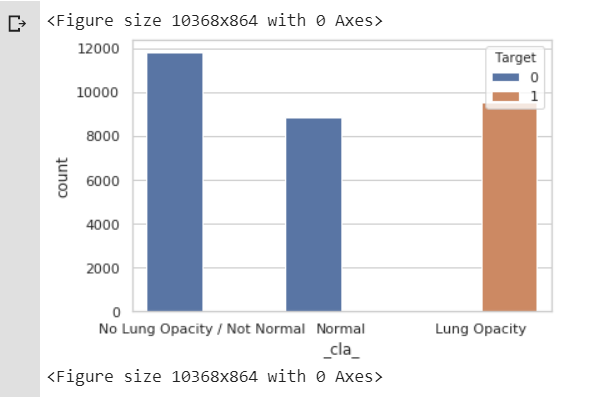
Next Perform EDA of the dataset Setup a dataframe - Gender, Viewing Position , Age etc.



After dropping 29 features



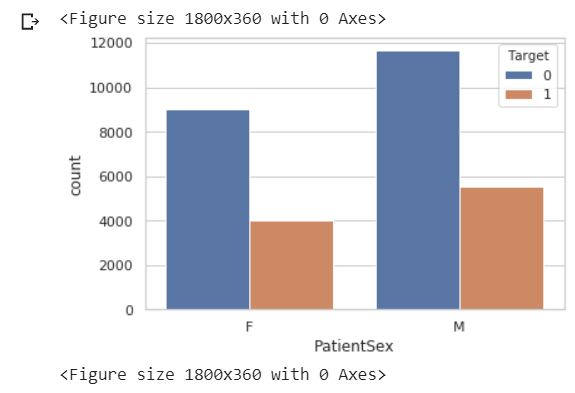
Frequency Chart of our detailed class ( Colored by Binary Class )



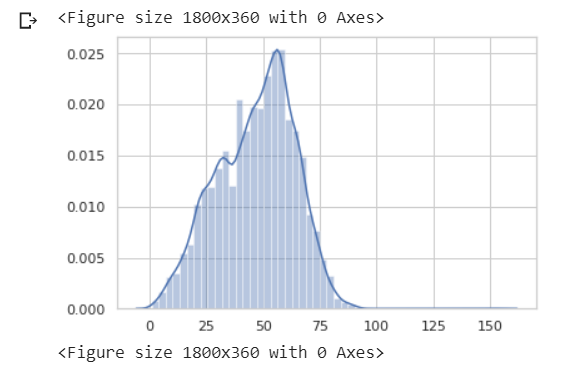
Frequency Chart of Binary Class ( Colored by Binary Class )



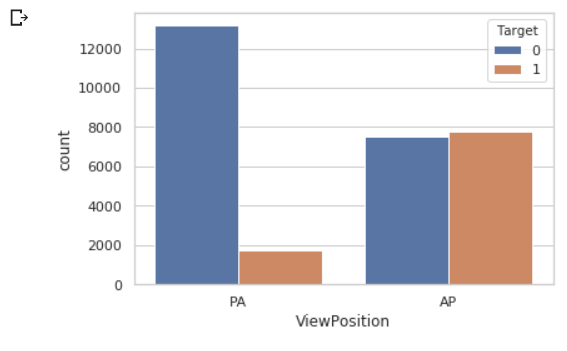
Frequency Chart of Sex ( Colored by Binary Class)



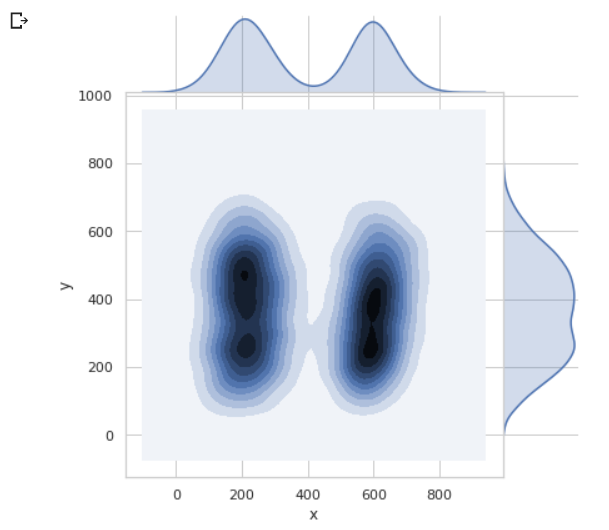
Distribution Plot of Patient Age



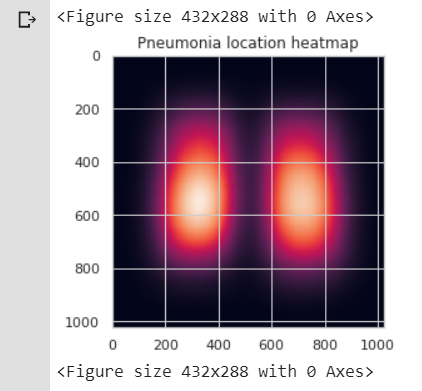
Clustered Column Chart based on viewing position



Heat map for x & y corners of each bounding box. But the below heatplot is imperfect.



So below will display Heat map of Pneumonia Presence in the sample image.



**Run Models within GCP :**

Generally, we were running our models in Jupyter notebook on Google Colab’s hosted run time.

Hosted run time runs on a new machine instance in Google cloud and we don’t need to set up any hardware.

We face situations like run time getting disconnected due to notebook inactivity and we had to re-run the code from the beginning.

For loading training and test data, first we have to upload to Google Drive and have to import it in our note book for accessing the data.

We can avoid the above two problems, by running Note book on local run time.

Local Runtime runs on your machine. You need to install Python, Jupyter, and set-up some necessary libraries required to run models. It is useful if you have a lot of data to process locally, or if you have your own powerful GPU to use.

Google Cloud Platform provides an option to create local run time through Virtual Machine Instance. Developers can customize the VM instance by choosing the computing power (CPU’s and GPU’s) which is required to run the models.

Below are the steps to create Virtual machine instance in Google Cloud platform.

In Chrome, go to URL: <https://cloud.google.com/>

Select “Get Started for free”. GCP allows free trial for 12 months and 300$ is credited for free.

In Step 1, select Country for example: India and click Accept Terms and Conditions check box and select continue.

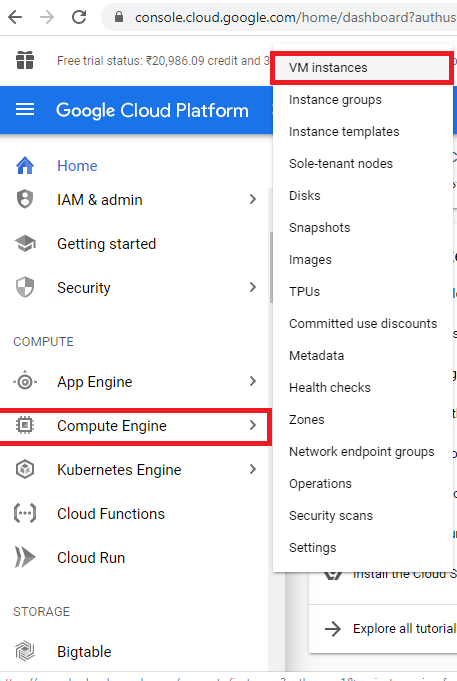
In Step 2, provide Name, Address and Credit Card details in payment. (Note: Provide valid credit card details only, as Rs.1/- will be deducted after OTP verification.)

Next, click on Start my free trial.

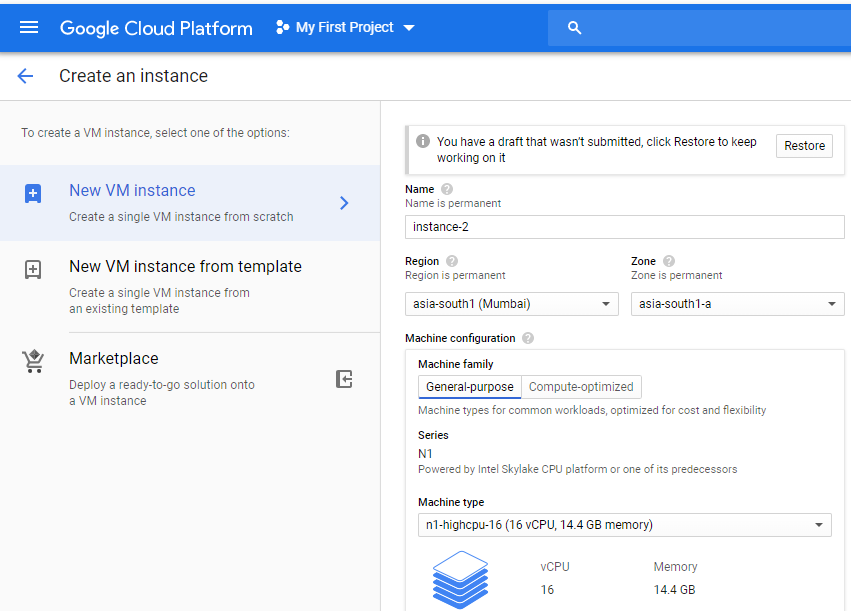
After Google Cloud Platform page is open, create a new project to access the all features of the GCP.

**Creating VM Instance:**

1. In the Navigation menu, go to Compute Engine > VM Instances as shown below.

****

1. In the VM instances page, click on “CREATE INSTANCE”. Below screen is displayed.



2.a Enter a name for the Instance.

2.b Change region to **asia-south1 Mumbai**, Select Zone as **asia-south1-a**

2.c Change the machine type to **n1-highCPU-8** or **n1-highCPU-16**

2.d GPU cannot be added under free trial.

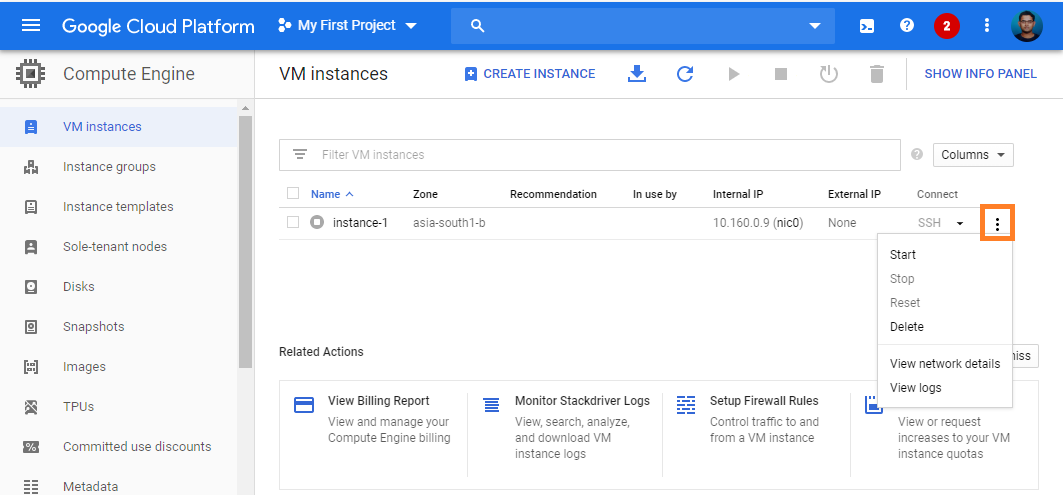
2.e Don’t change the boot disk, default Debian GNU/Linux is selected.

2.f Under fire wall, check the Allow http traffic and Allow https traffic checkboxes.

Click on Create.

1. Newly created VM instance is displayed as below.

Next, Click on Options menu to Start/Stop VM instance.



1. Once VM instance is started remotely in GCP, the Linux Operating System can be accessed from our local Window machine via SSH.

SSH provides a [secure channel](https://en.wikipedia.org/wiki/Secure_channel) over an unsecured network in a [client–server](https://en.wikipedia.org/wiki/Client%E2%80%93server_model) architecture, connecting an [SSH client](https://en.wikipedia.org/wiki/SSH_client) application with an [SSH server](https://en.wikipedia.org/wiki/SSH_server).

1. For connecting to Linux operating via SSH. Google Cloud has provided a **Command Line** Tool which is available in Google Cloud SDK. Download the SDK installer from the below link and install Google Cloud SDK on your local machine.

Link: <https://cloud.google.com/sdk/docs/downloads-interactive>

1. After installation, open Google Cloud command-line tool from Windows local machine as below:



1. For more info about Google Cloud Command-line tool, please refer to <https://cloud.google.com/sdk/gcloud>

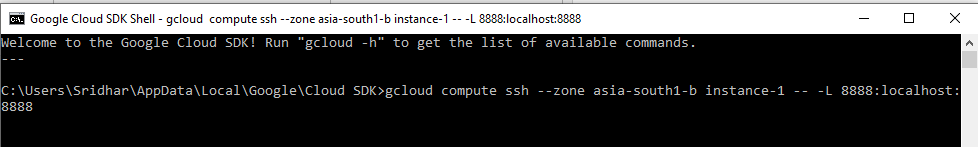
**What is the gcloud command-line tool?**

The gcloud command-line interface is a tool that provides the primary Command Line Interface to Google Cloud Platform. You can use this tool to perform many common platform tasks either from the command-line or in scripts and other automations.

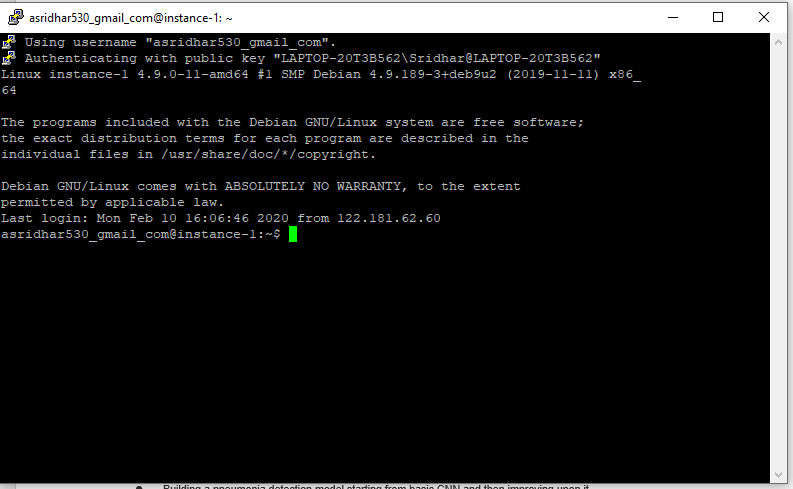
1. Connecting to Linux OS – Virtual Machine Instance.

In the command line interface, please run the below command.

**gcloud compute ssh --zone asia-south1-a instance-1 -- -L 8888:localhost:8888**

****

1. After running the above command, we can able to access the Linux OS running on Virtual Machine instance via SSH.



1. In this console, we can run the Unix commands to install Python, Jupyter and other libraries which are required to run our models.

10.a First we need to Install PIP command. The pip command is a tool for installing and managing Python packages.

Please refer to <https://linuxize.com/post/how-to-install-pip-on-debian-9/>

10.b Install Jupyter notebook. Command: **pip install notebook**

10.C Install Pandas, Numpy etc. using Command **pip install “package-name”**

1. After installing the necessary libraries, Install and enable the **jupyter\_http\_over\_ws**jupyter extension (one-time).

This Jupyter server extension allows running Jupyter notebooks that use a WebSocket to proxy HTTP traffic. Browsers do not allow cross-domain communication to localhost via HTTP, but do support cross-domain communication to localhost via WebSocket.

**pip install jupyter\_http\_over\_ws**

**export PATH=$PATH:~/.local/bin**

**jupyter serverextension enable --py jupyter\_http\_over\_ws**

1. Starting Jupyter Notebook server in Linux OS. Below command allows connection from

Google Colab.

1. First go to bin folder path:

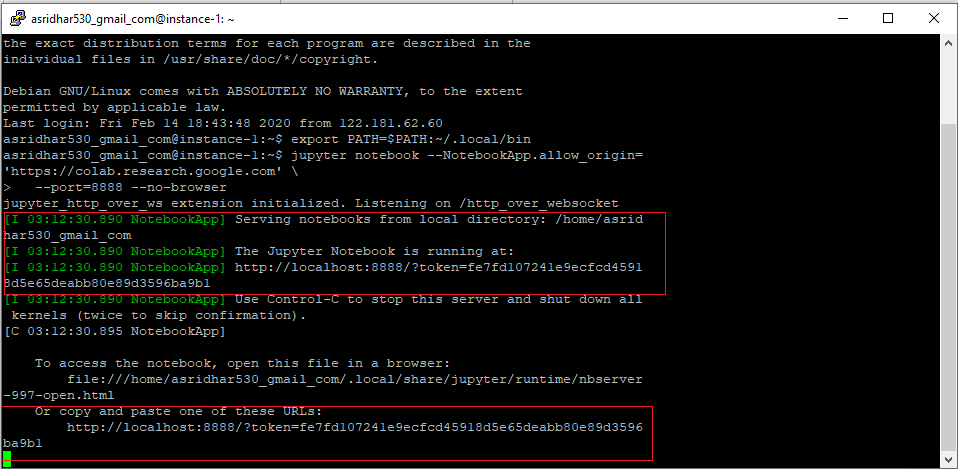
**export PATH=$PATH:~/.local/bin**

1. Run the command

**jupyter notebook --NotebookApp.allow\_origin='https://colab.research.google.com' \**

**--port=8888 --no-browser**

1. After running the above command, Jupyter Notebook will start running in VM instance as shown below:

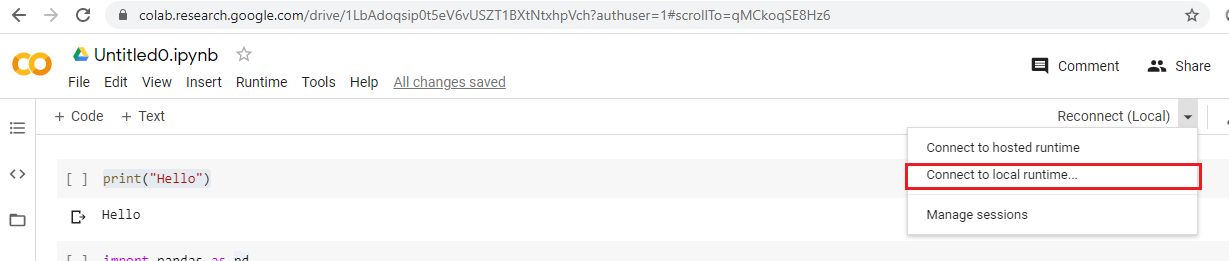


1. To access this Notebook in your Google Colab notebook, copy the URL displayed in the console.

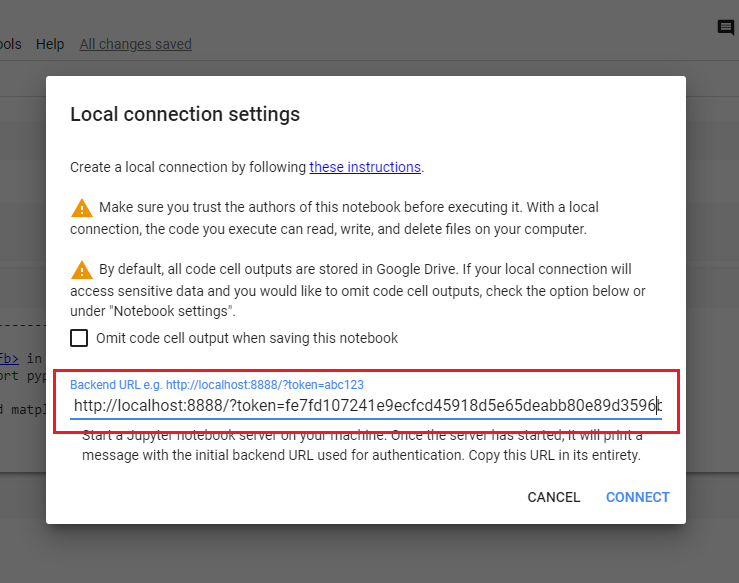
**Note: This URL will change every time when you start Notebook on the VM instance.**

**So, the above console should be running when you are running models in Google Colab.**

1. Open Google Colab in your Chrome browser, create a new Python note book. Click on **Connect to local runtime** as shown below.

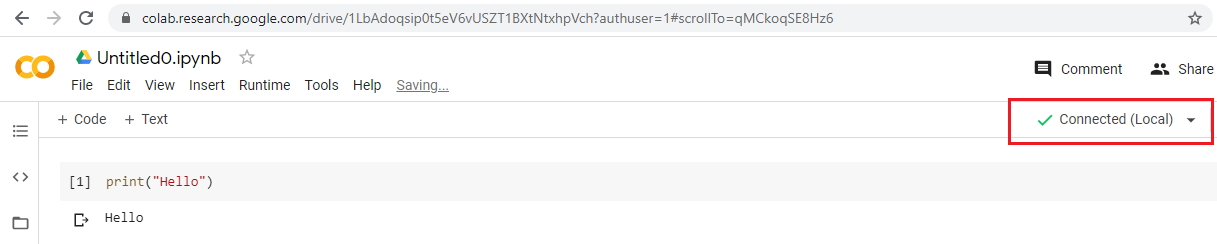


1. Copy the URL from console and paste it in the below marked as shown below and click on **Connect.**



1. After the connection is successful, the notebook should show as below:

Next, we can start running the code.



Deploy the Models

### The Approach Going Forward

Build prediction model based on following :

* Building a pneumonia detection model starting from basic CNN and then improving upon it.
* Train the model
* To deal with large training time, save the weights so that you can use them when training the model for the second time without starting from scratch.

● Test the model and report as per evaluation metrics

● Try different models ( SSD, YoloV3,

● Set different hyper parameters, by trying different optimizers, loss functions, epochs, learning rate, batch size, check pointing, early stopping etc..for these models to finetune them

● Report evaluation metrics for these models along with your observation on how changing different hyper parameters leads to change in the final evaluation metric.