# PGP AI ML – Capstone Final Submission

# Pneumonia Detection Challenge

Group A CV Batch 1

# Automating Pneumonia screening in chest radiographs and providing affected area details through powerful AI techniques can assist physicians to make better clinical decisions

- **PROBLEM STATEMENT:** Pneumonia accounts for over 15% of all deaths of children under 5 years old internationally. Accurately diagnosing pneumonia requires review of a chest radiograph (CXR) by highly trained specialists and confirmation through clinical history, vital signs and laboratory exams. **Pneumonia usually manifests as an area or areas of increased opacity on CXR**. However, the **diagnosis of pneumonia on CXR is complicated** because of a number of other conditions in the lungs such as fluid overload (pulmonary edema), bleeding, volume loss (atelectasis or collapse), lung cancer, or post-radiation or surgical changes. A number of factors such as positioning of the patient and depth of inspiration can alter the appearance of the CXR, complicating interpretation further. Tissues with sparse material, such as lungs which are full of air, do not absorb the X-rays and appear black in the image. Dense tissues such as bones absorb X-rays and appear white in the image
- **OBJECTIVE:** We need to ***build a Pneumonia detection model*** to detect a visual signal for pneumonia in medical images. Specifically, the algorithm needs to automatically locate lung opacities on chest radiographs, providing affected area details through bounding box
- **DATA DESCRIPTION**: **Data provided is Dicom original images**: - Medical images are stored in a special format called DICOM files (*.dcm). They contain a combination of header metadata as well as underlying raw image arrays for pixel data. While we are theoretically detecting “lung opacities”, there are lung opacities that are not pneumonia related. In the data, some of these are labeled “Not Normal No Lung Opacity”. This extra third class indicates that while pneumonia was determined not to be present, there was nonetheless some type of abnormality on the image and oftentimes this finding may mimic the appearance of true pneumonia. Lung Opacity class refers to Pneumonia cases. There are around 26000 2D single channel CT images in the pneumonia dataset that provided in DICOM format
- **FINDINGS:** xyz…

# Process steps

Salient features of data pre-processing steps, the algorithms used

# Project Approach

**1**

**Import and warehouse data**

Install, Import all required libraries such as Pydicom to read Dicom images, Albumentations for augmentations, Torch, torch vision and Detectron2

**2**

**Data Curation and Augmentation**

- Merge Train labels and class info to get bounding box dimensions Perform augmentation using albumentations
- Convert images to jpg

**3**

**Analysis, Visualization, preliminary insights**

- Perform detailed statistical analysis on the data

**4**

**Data processing and feature engineering**

- We noted that images can contain multiple bounding boxes
- Perform train-test split

**5**

**Model training, testing and tuning**

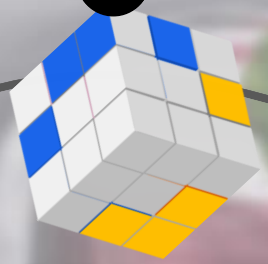
- Train and test variety of models architectures
- Compare models using classification matrix, mAP
- Re-run model post augmentations and compare metrics pre and post

**6**

**UI development**

- Select and pickle the final model
- Perform Model deployment - rest API
- Created a UI that allows user to input image and the output displays whether Pneumonia is present or not

**7**



***Conclusion and Key Insights***

# MODEL BUILDING: Salient features of each model used

INSERT DET/
for each mo

## Faster R-CNN

- We apply transfer learning of the Faster R-CNN model i.e. pre-trained model on the COCO dataset
- We split the data into 90:10, trained the model
- This code is developed using RoIAlign instead of RoIPooling
- We used  Visualizer utility to draw the predictions on the image

## U-NET

xyz

## VGG19

xyz

## ResNet50

xyz

## Inception Net v3

xyz

# Model Evaluation

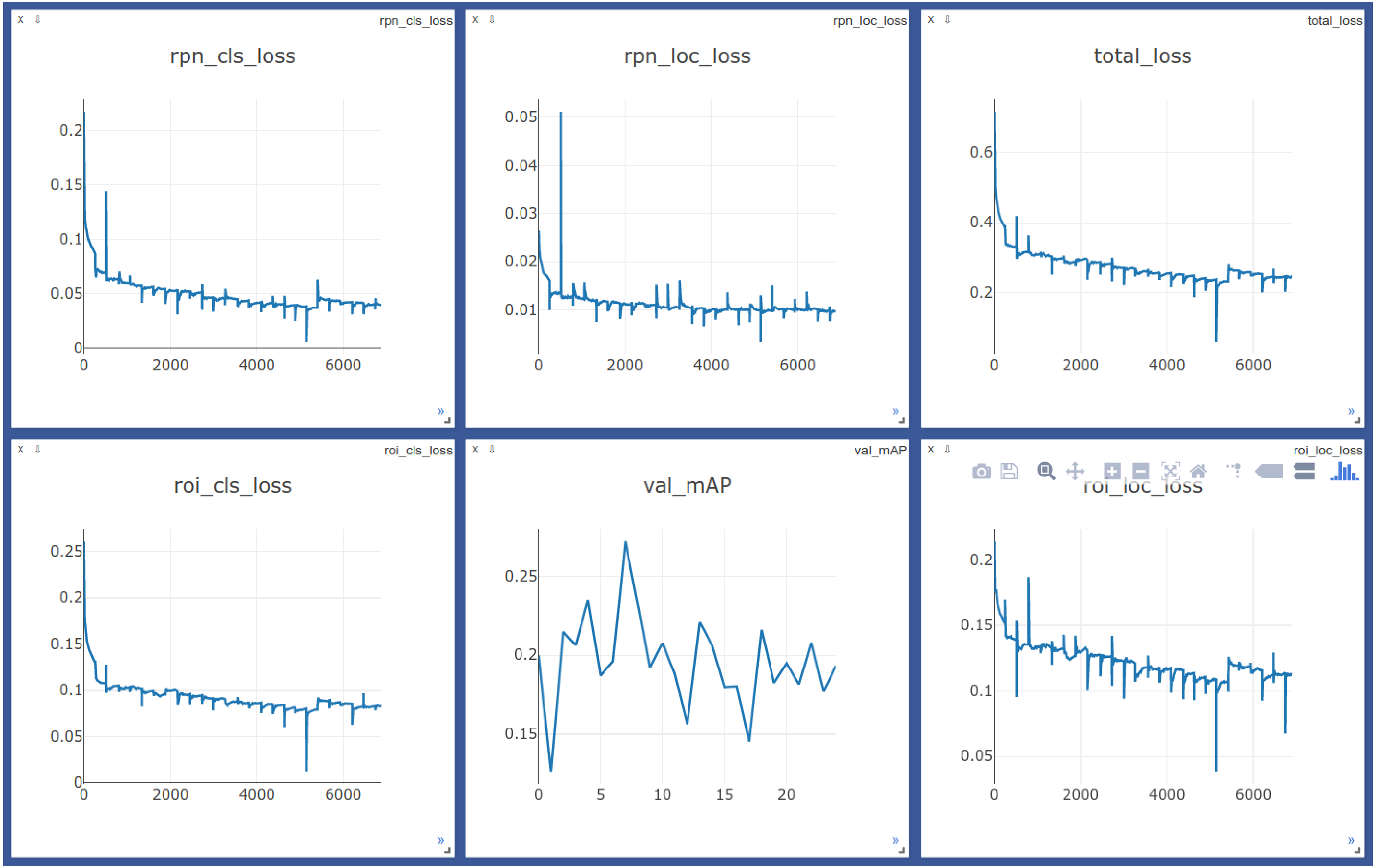
***What parameters were prominent and how did you evaluate the success of your models?***

## Model evaluation

- We use classification matrix to check for precision values of each model

- Since its an object detection problem, we finalize faster R-CNN among all the architectures

- This project has also been evaluated on the mean average precision at different intersection over union (IoU) thresholds for a set of predicted bounding boxes and ground truth bounding boxes. The metric sweeps over a range of IoU thresholds, at each point calculating an average precision value. A true positive is counted when a single predicted object matches a ground truth object with an IoU above the threshold. A false positive indicates a predicted object had no associated ground truth object. A false negative indicates a ground truth object had no associated predicted object




SAMPLE can we make & display graphs li this?

# How does your final solution compare to the benchmark you laid out at the outset? Did you improve on the benchmark? Why or why not?

REVISIT

- We expected high mean average precision of ### at the outset but did not quite achieve the same. We measured it for various IoU thresholds

# Summary of findings

| Architectures | Accuracy | Precision | Recall | AUC-ROC |
|---|---|---|---|---|
| UNet | 23% | 23% | 100% | |
| Faster R-CNN | 63.4% | 37% | 91% | |
| VGG19 | 81% | 55% | 93% | 91.7% |
| VGG16 | 80% | 50% | 81% | 90.7% |
| ResNet50 | 77% | 55% | 80% | 83.2% |
| InceptionNet v3 | 86% | 55% | 81% | 86.8% |

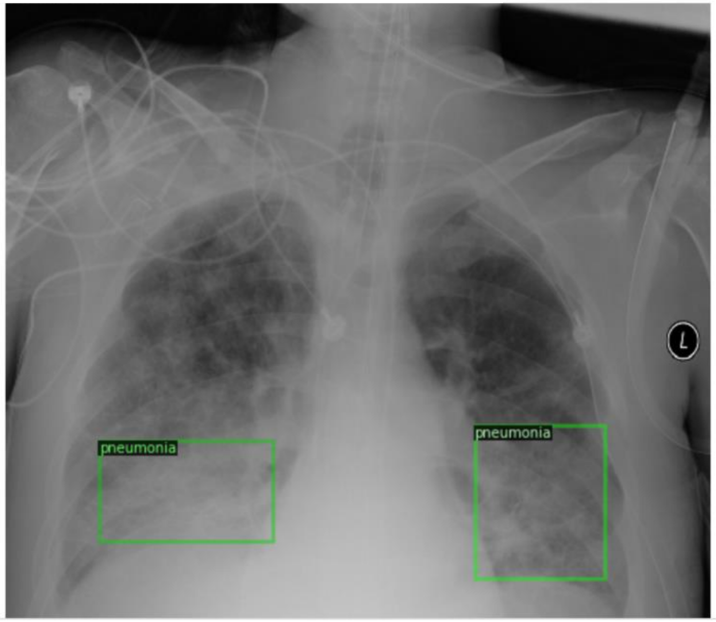
Although from the consolidated results, **VGG19** seems to be the best model with highest recall, we will continue with Faster RCNN since this is an object detection problem

Since the problem statement requires us to read images, we use CNN and various state of the art deep learning architectures listed here to identify lung opacity with higher recall and accuracy

We have run these architectures on full data set except UNet which was processed on subset of data

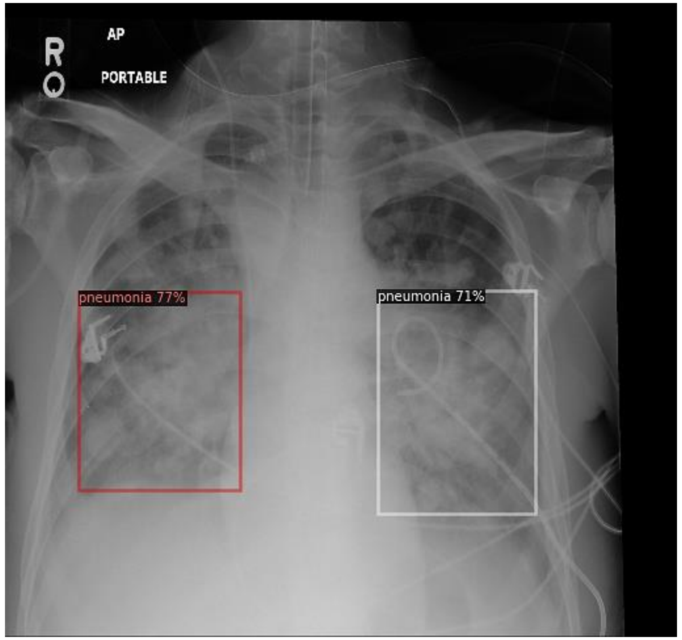
## The Faster R-CNN model is trained to predict the bounding box of the pneumonia area with a confidence score

GROUND TRUTH




PREDICTION (Against ground truth)

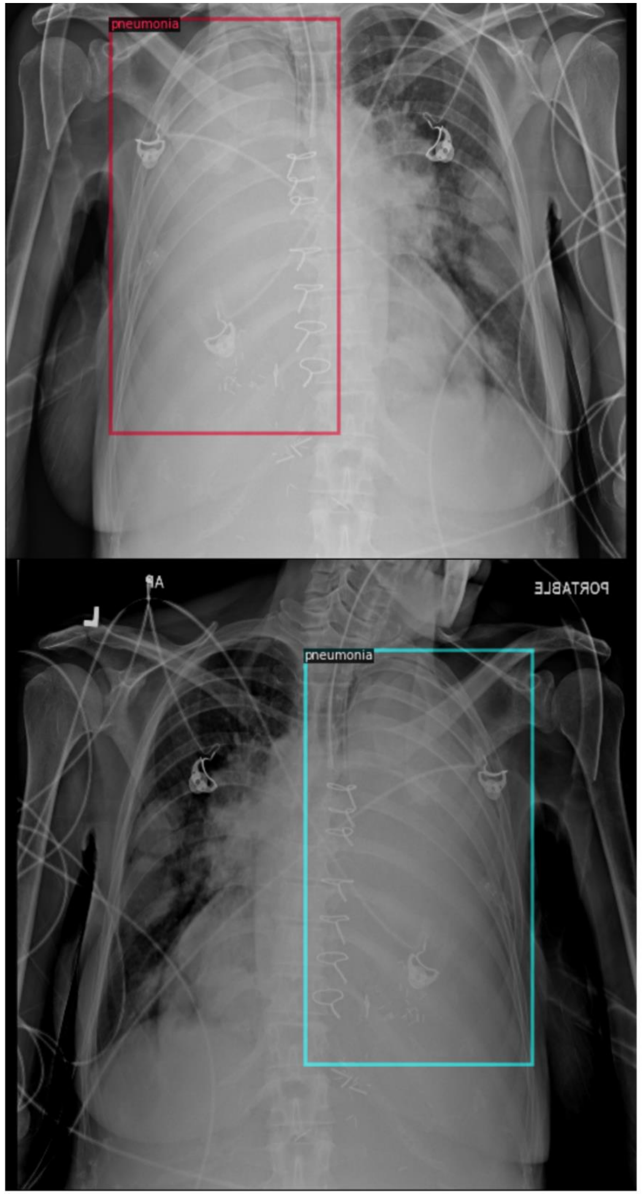

PREDICTION FOR ANOTHER PATIENT




Above are few samples of “Faster R-CNN” model predicting bounding boxes of pneumonia patients with confidence intervals

# Data Augmentation on Faster R-CNN

**Detectron2 is built using Pytorch, and provided a very easy API. Detectron2 originates from Mask R-CNN benchmark. Detectron2 helps in Panoptic segmentation ( combination of semantic and instance based). It can also be used as a wrapper on top of other projects and can be exported to easily accessible formats. Detectron2  is flexible and fast training on single or multiple GPU servers.**




Faster R-CNN model was developed using Detectron2 on Pytorch. All dicom images are converted to JPG as Detectron2 can't directly read dicom images. We prepare the dataset into Json format and then feed into the Detectron2 model for training

Standard augmentation function for images with bounding boxes was not feasible and hence we developed a pipeline using Albumentations library from PyPi. Augmentation was performed only on 10% of dataset sampled randomly.

The X-ray picture shows that augmentation is working with **horizontal flip of image** with bounding box

Below is the outcome of our model with augmentation. **However, since recall of Faster R-CNN before augmentation is better, we will proceed with it.**

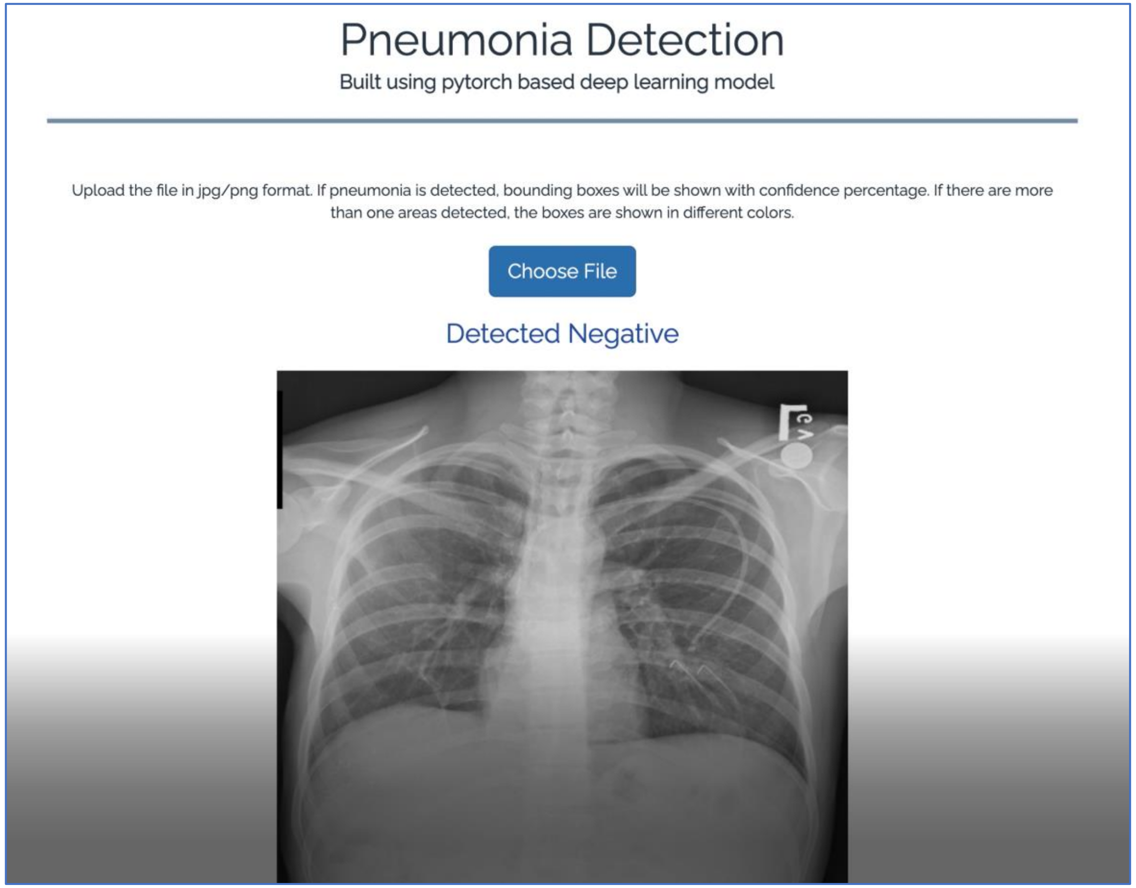
```
print(classification_report(df_aug['ground_truth'], df_aug['predicted']))
accuracy = accuracy_score(df_aug['ground_truth'], df_aug['predicted'])
print('Model accuracy is: ', accuracy)
```

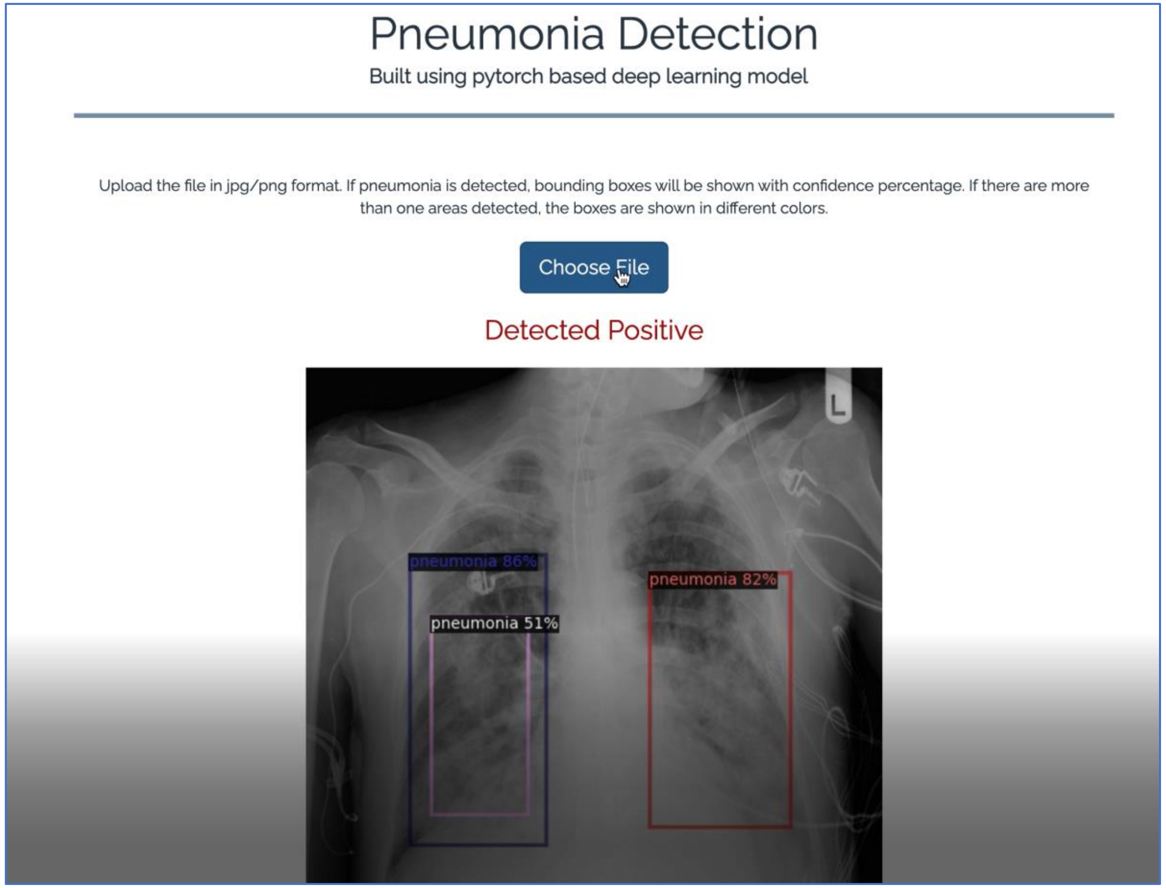
```
              precision    recall  f1-score   support

           0       0.97      0.44      0.60      2076
           1       0.39      0.96      0.55       769

    accuracy                           0.58      2845
   macro avg       0.68      0.70      0.58      2845
weighted avg       0.81      0.58      0.59      2845

Model accuracy is:  0.5792618629173989
```

# Model Deployment - xyz

**INSERT DETAILS deployment**

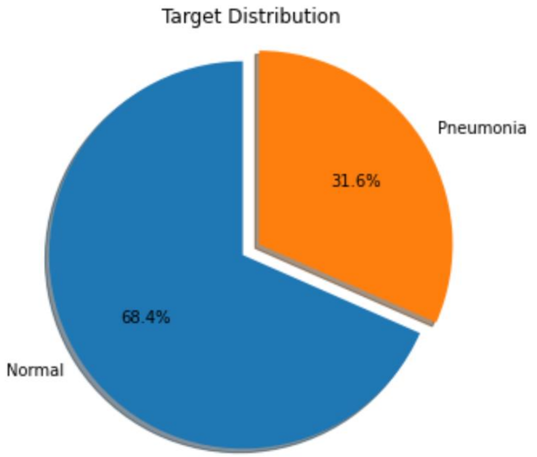
## USER INTERFACE

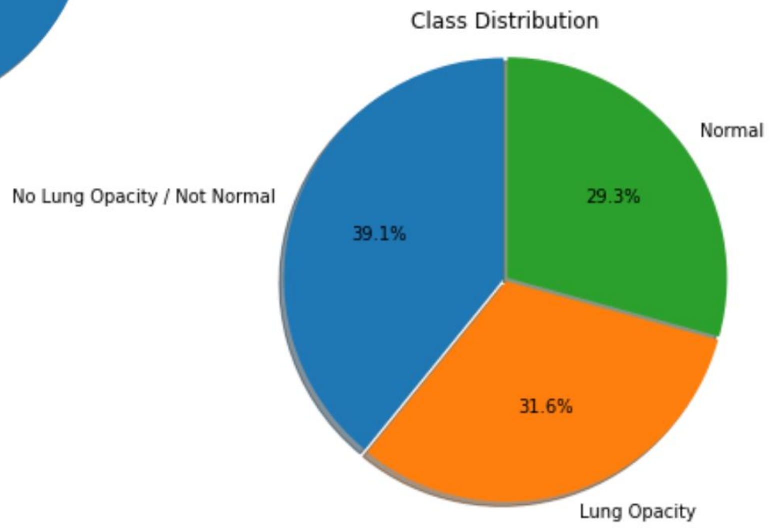


# Exploratory Data Analysis

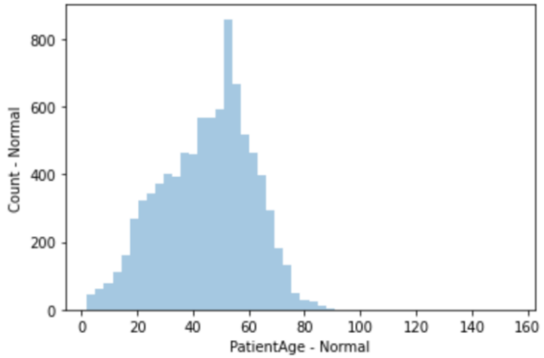
- Used pydicom library to read Dicom images
- 26684 unique patientIDs
- Few patients have multiple bounding boxes
    - 3266 patients have 2 bounding boxes defined
    - 119 patients have 3 bounding boxes defined
    - 13 patients have 4 bounding boxes defined
- We will run a binary classification to predict patients with pneumonia
- Target – 1 indicates patients with pneumonia and target – 0 indicates normal patients or patients having lung opacity but not pneumonia
- 8,851 (29.3%) patients are healthy/Normal
- 9,555 (31.6%) patients have Lung Opacity
- 11,821 (39.1%) patients have No Lung Opacity but may have other lung abnormality
- Clearly, dataset is bit imbalanced with Target class being only 31.6% of the whole dataset

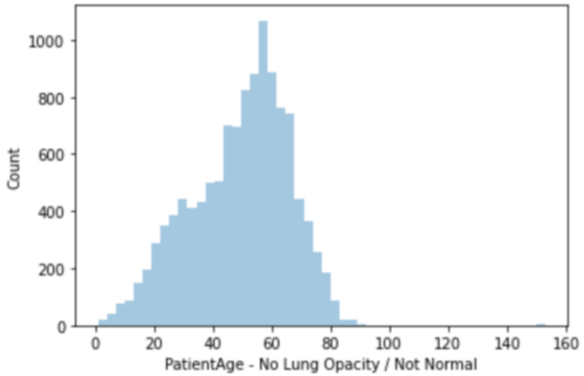


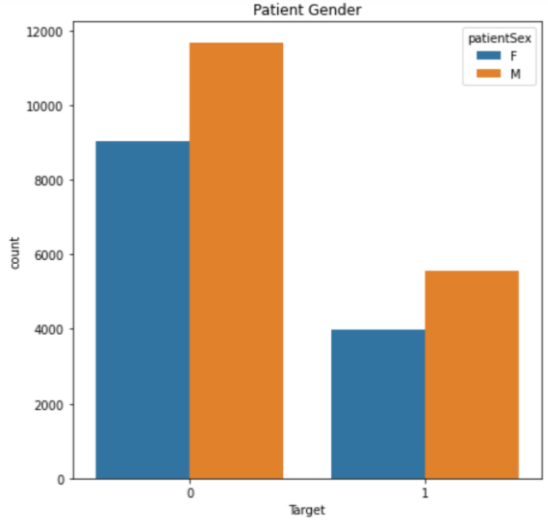



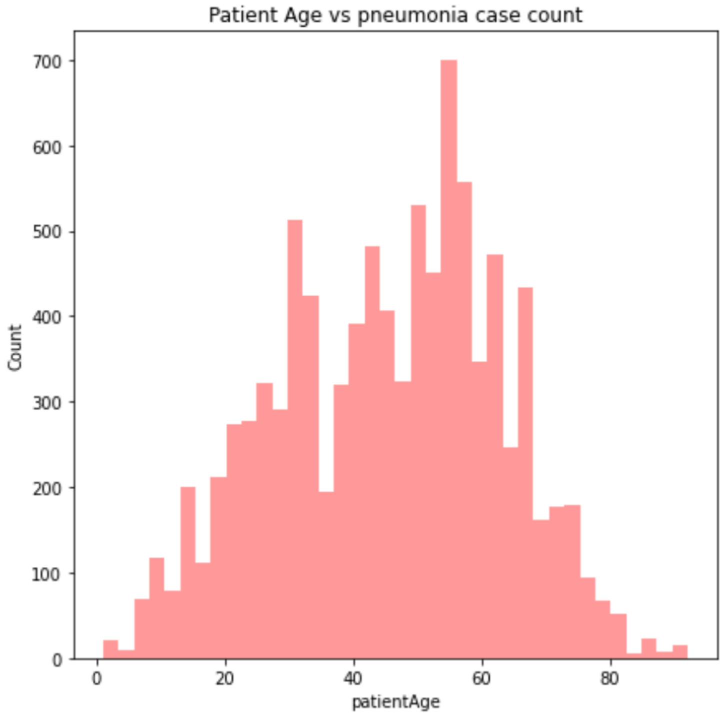

| | class | Target | Patient Count |
|---|---|---|---|
| **0** | Lung Opacity | 1 | 9555 |
| **1** | No Lung Opacity / Not Normal | 0 | 11821 |
| **2** | Normal | 0 | 8851 |

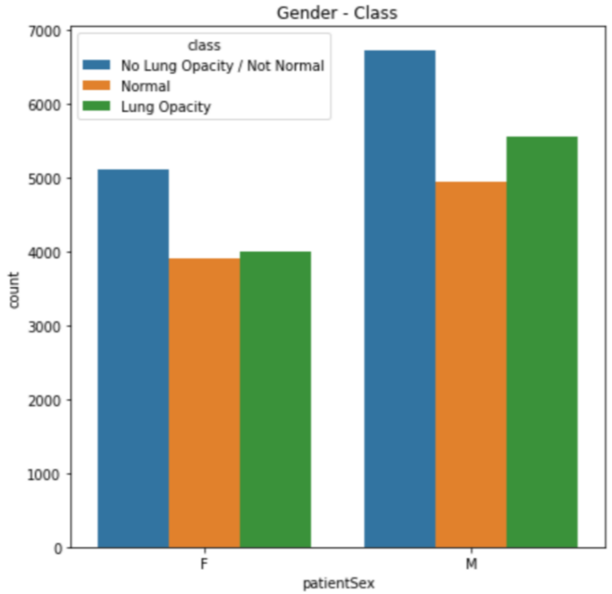
- Patient distribution shows most pneumonia patients or patients are found between age 40-60
- More records for male is observed in the data
- Proportion of pneumonia and normal patients re equal for female patients
- However, for males, proportion of pneumonia patients are more than normal

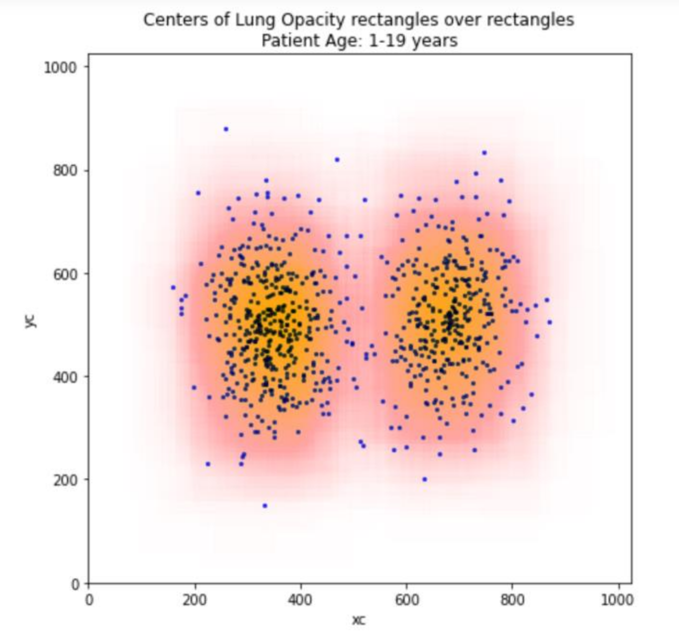


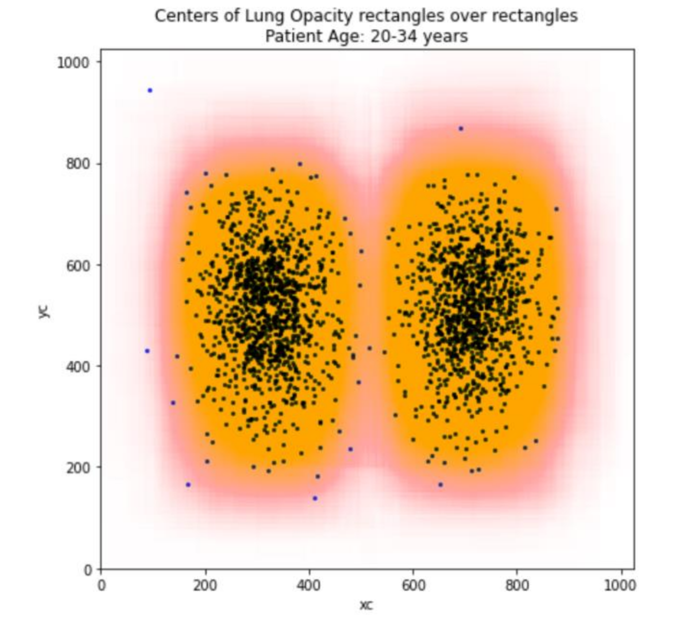



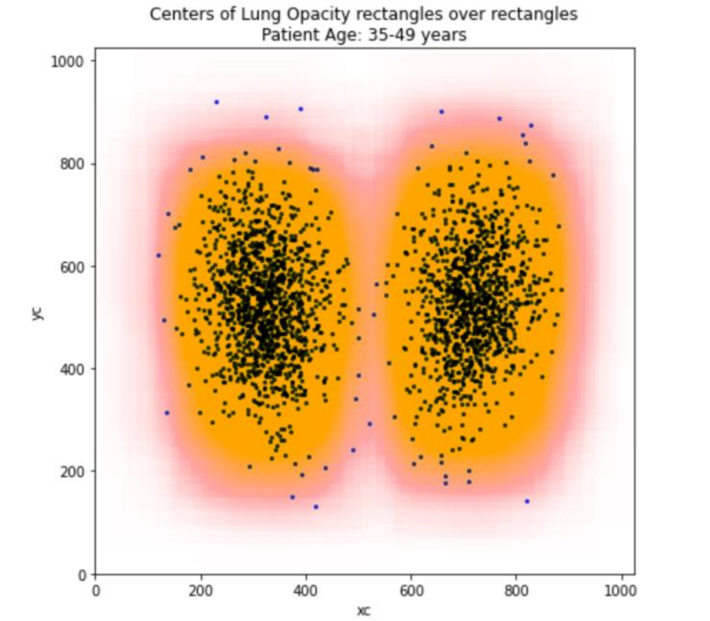



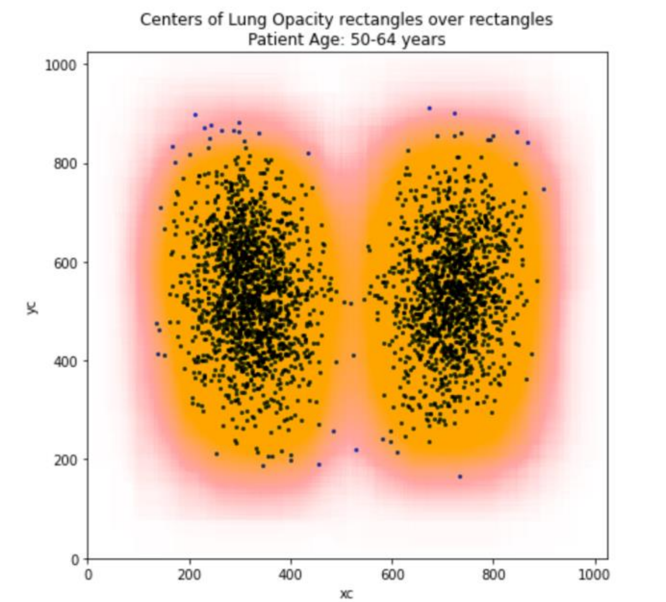



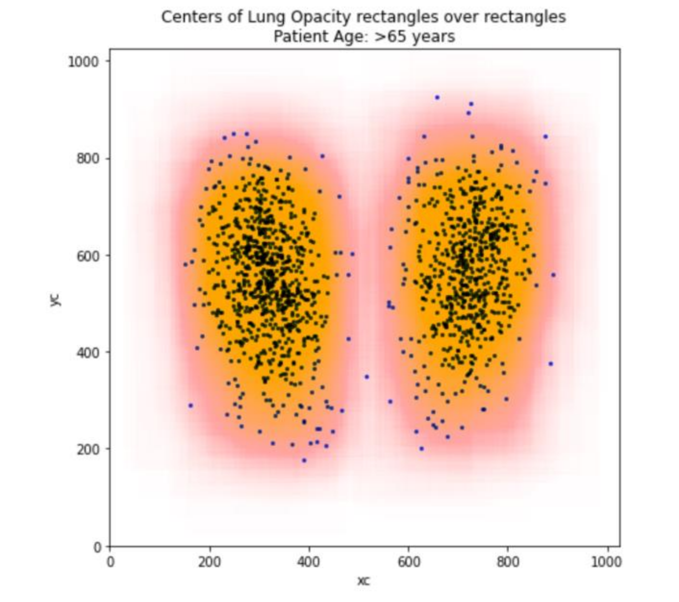





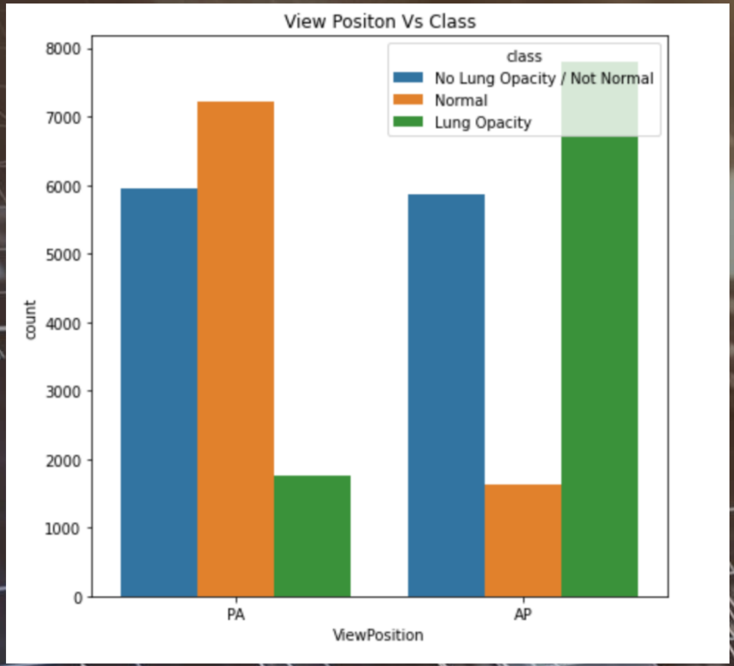



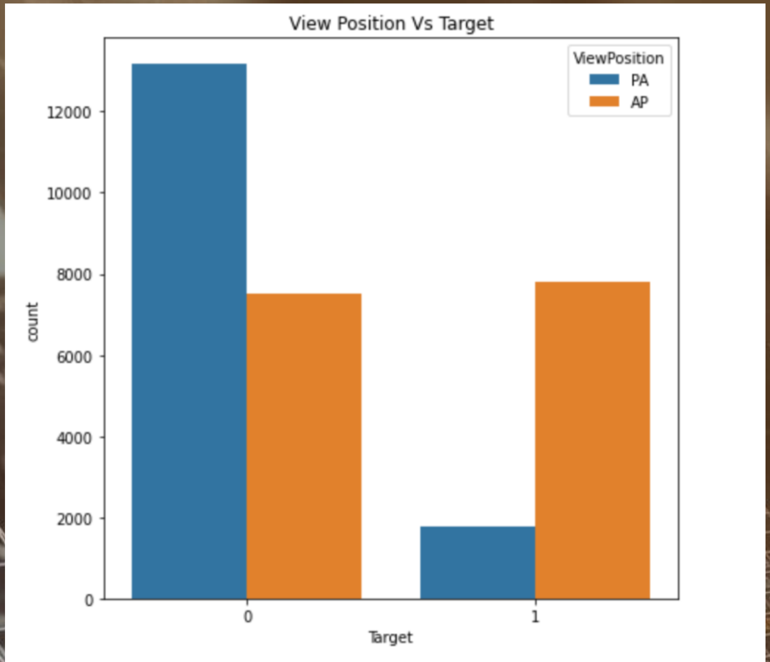



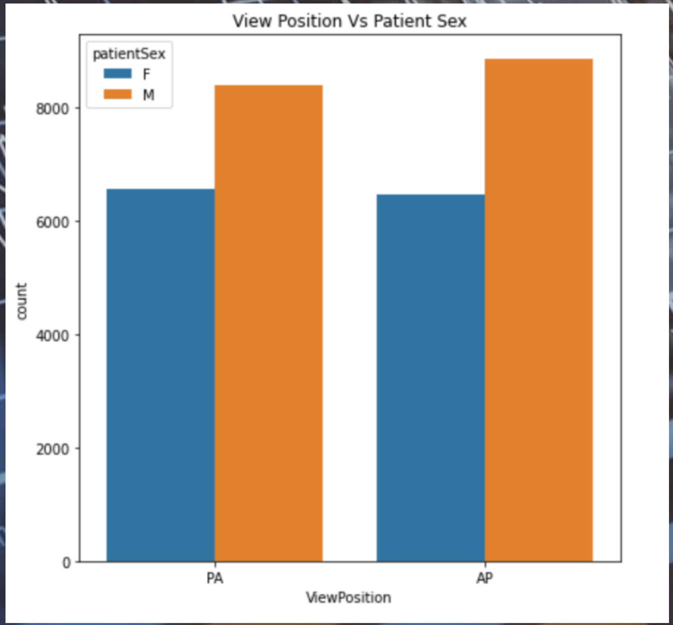



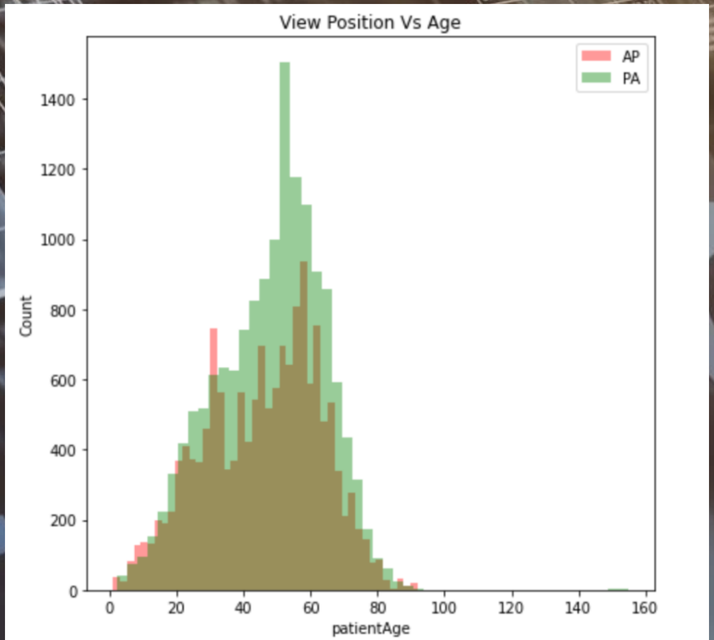

**Lung opacity is seen maximum in age 50-65 followed by age group 20-50**

- Chest X-ray can be done in **Posterior-Anterior (PA)** projection and **Anterior-Posterior (AP)** Projection
- Lung opacity seems to be more visible in AP position

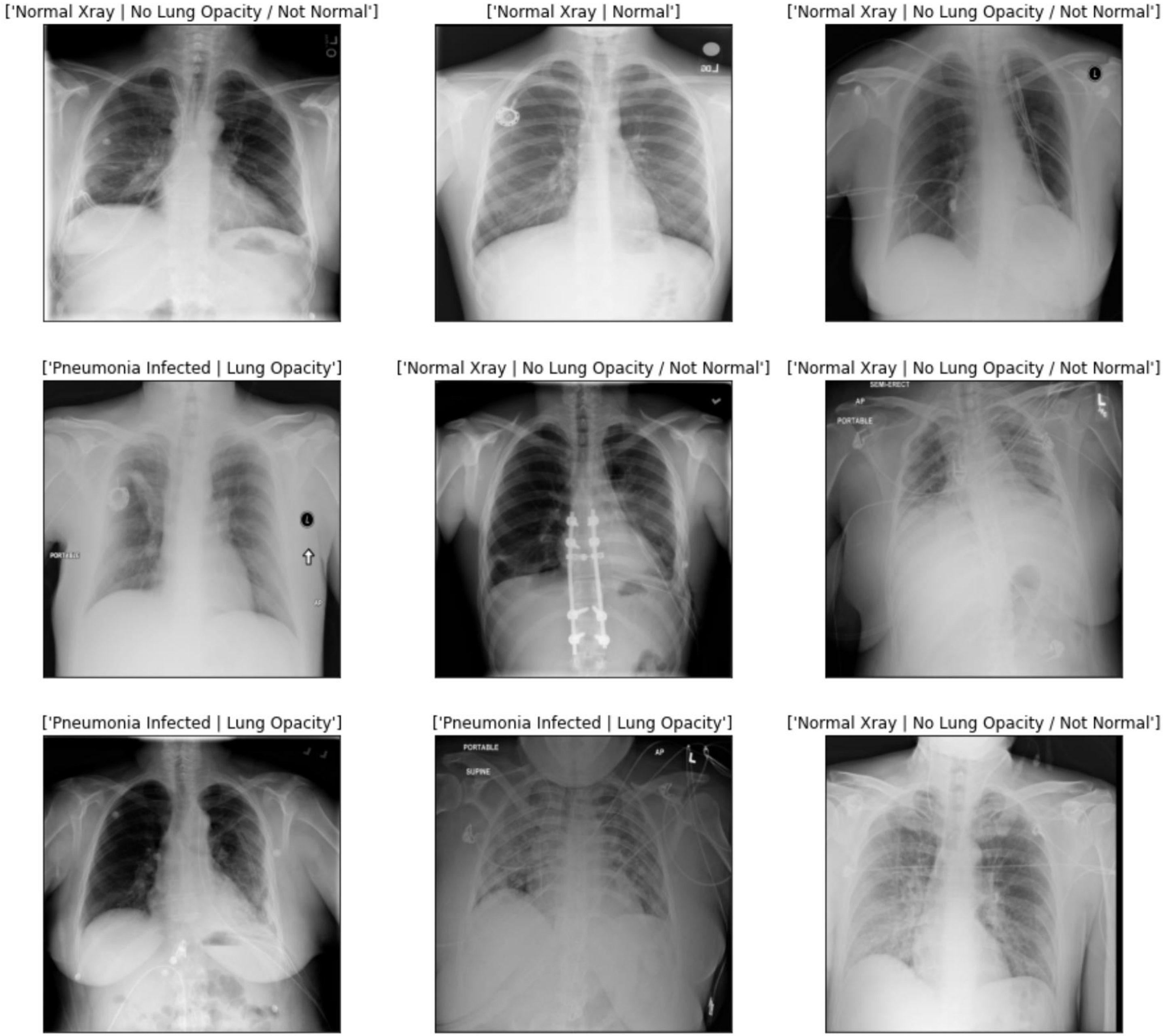


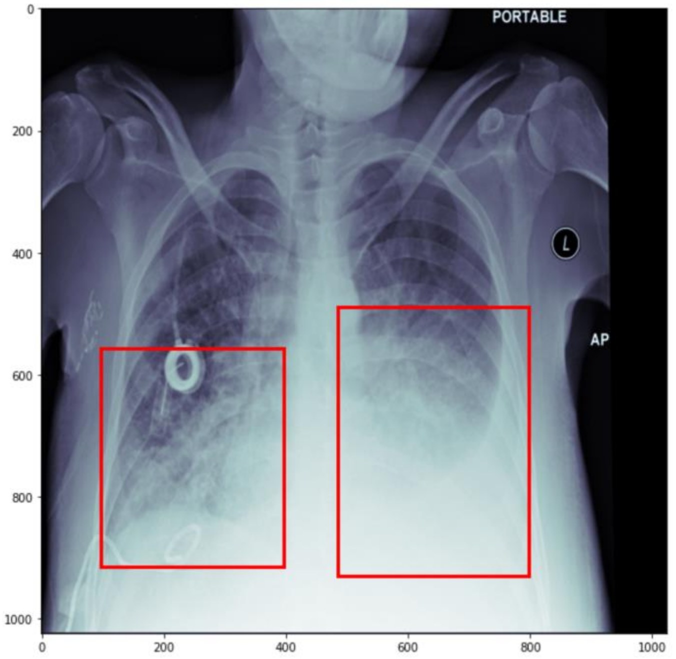



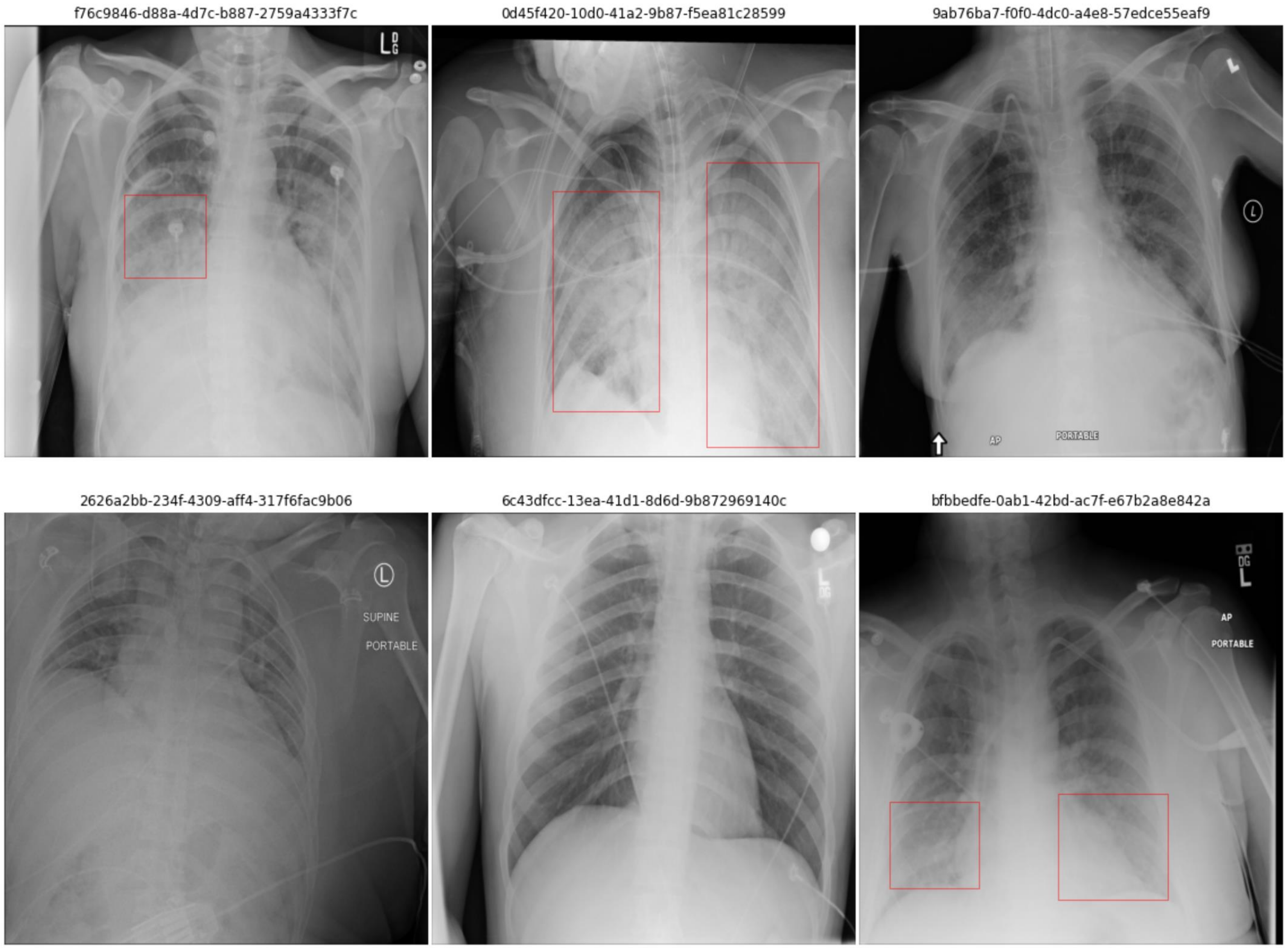



Visualization of sample X-ray images for each category



Visualization of sample X-ray images with bounding boxes depicting lung opacity



# Implications

*How does your solution affect the problem in the domain or business? What recommendations would you make, and with what level of confidence?*

The size, shape, and position of pneumonia can vary a great deal. Its target contour is very vague, which leads to great difficulty with detection, and enhancing the accuracy of detection is a major research problem. Medical testing has high requirements for accuracy, and hence two-stage detectors such as Faster R-CNN have an advantage in this respect.

TRUE POSITIVE

TRUE NEGATIVE

Insert cases of

FALSE POSITIVE

FALSE NEGATIVE

Review

# Limitations

*What are the limitations of your solution? Where does your model fall short in the real world? What can you do to enhance the solution*

There are still problems with the backbone network of the current detection algorithms. For example, ResNet generally has two problems: a large network depth leading to long training time and massive down-sampling that leads to the target position and semantic information being lost

The dataset contains three categories of subjects, normal, pneumonia, and abnormal(cancer or other diseases) but only provides the bounding box for pneumonia images. However, the features of pneumonia and abnormal(cancer or other diseases) are pretty similar, which caused the failure to distinguish pneumonia and abnormal images for Faster R-CNN. This results in predicting bounding box for abnormal images

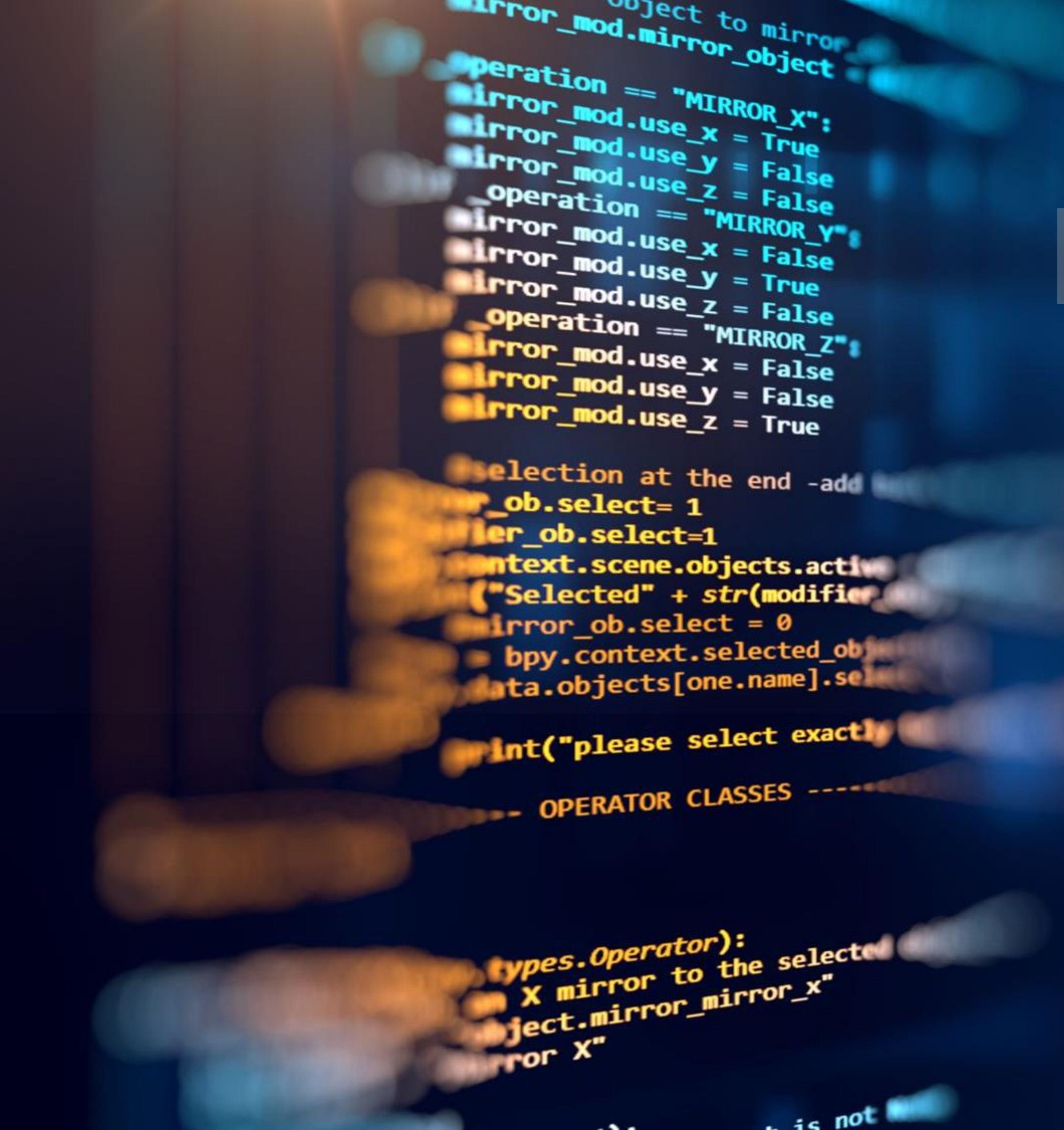
Different distribution of train and test datasets, most likely due to different labeling methodology

Learnings?

# Closing Reflections

*What have you learned from the process? What you do differently next time?*

We had class imbalance for pneumonia cases in our dataset. Hence, using NIH dataset since It's a bigger dataset but with lower quality of labels, would be very interesting to check if training the model to predict both datasets would improve the result, or at least use it to pretrain the base model




UPDATE

# References – Code Base

| Model/ Analysis | HTML file |
|---|---|
| EDA, VGG19, VGG16, ResNet, InceptionNet v3 | Pneumonia_EDA_VGG19_VGG16_ResNet50_InceptionNet_Complete.html |
| UNET | Pneumonia_UNET.html |
| Faster RCNN using Detectron | Pnuemonia_pytorch_complete.html |