FDA Submission

Your Name:: Kofi Ofuafor

Name of your Device:: Pneumonia detection algorithm

Algorithm Description

1. General Information

Intended Use Statement: More than 1 million adults are hospitalized with pneumonia and around 50,000 die from the disease every year in the US alone.

This algorithm is intended for use in *assisting the clinician* in acute care setting with a speedy detection of pneumonia within a chest x-ray. In otherwords it is to be used a a screening test for Pneumonia. The predicate device for this algorithm is a CADx device.

Indications for Use: This algorithm is used as a Pneumonia classifier in a chest x-ray taken in acute medical or surgical admission units, accident and emergency and acute care wards. It is intended for use in people aged 20-70 years with *no prior history* of presence of edema, pleural effusion and pulmonary infiltrations

Device Limitations: The device study was based on images from people of age 20- 70 and was significantly less accurate in the presence of concurrent disease and therefore not recommended for use in the catgory of patients were there is prior presence of disease in a coloumn num_prior_positive in the demographic data. The algorithm was not traing in patients over 80 years old.

Clinical Impact of Performance:

- Performance statistics
  - Accuracy: 81%
  - Precision: 0.16666666666666666
  - Sensitivity: 1
  - Specificity: 0

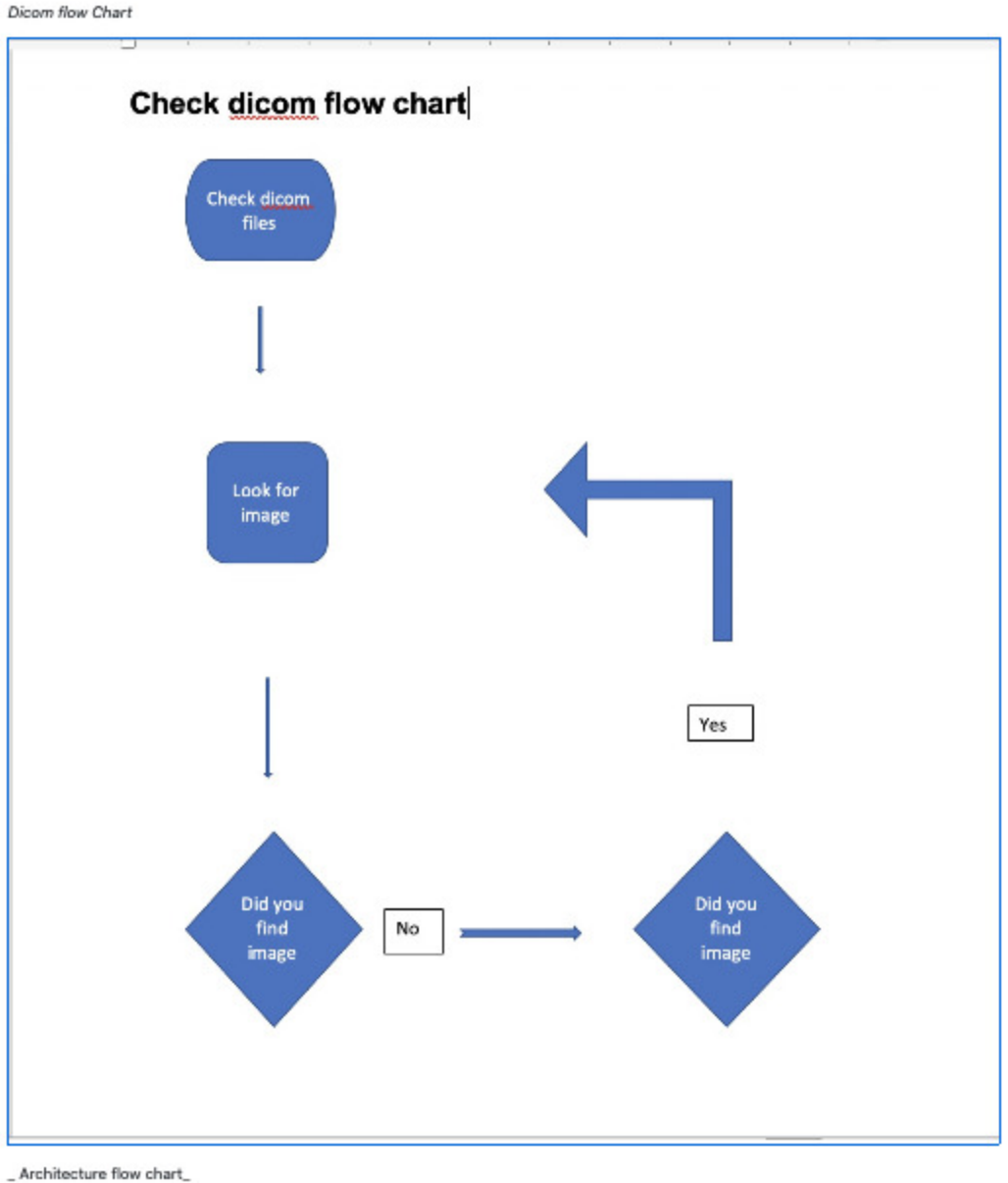
For this algorithm, missing a detection of Pneumonia is not acceptable as human life and depends on it. False negative may slow down clinicians urgency to look at the patient xray for a physical examination. Therefore False Negative results have more impact than a False positive. Accordingly, the threshold for the algorithm has been weighed in favour of recall(sensitivity).

When a test with high recall returns a negative result, you can be pretty confident that the result is truely negative. Recall does not take into account FP though, so you may still be labelling alot of negative cases as positive. So recall are good for screening tests.

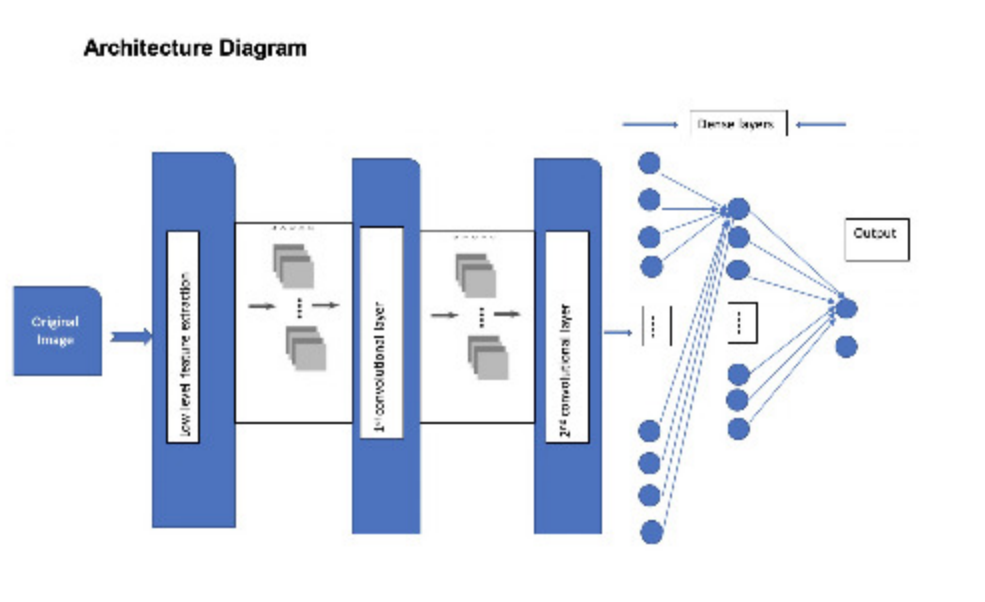
2. Algorithm Design and Function

Algorithm Fow chart The check_dicom function reads in a .dcm file, checks the important fields for our device, and returns a numpy array of just the imaging data.

Dicom flow Chart



_ Architecture flow chart_



The function, check_dicom(), is required to check the image type(MODALITY), Body Part Examined (Chest), and Image position (PA or AP) for each DICOM image and check if the input to the algorithm is valid to be predicted by our algorithm or not.

DICOM Checking Steps: There are three steps:

1. Extract the desired attributes *Modality: DX The image type DX is a digital radiography *Body Part Examined: Must be Chest *Patient Position: AP or PA The image position could be AP or PA *Study Description: label This is the label for the chest x-ray.
2. Next the check_dicom function checks each image if the input to the algorithm is valid to be predicted upon by our device.
3. Finally, valid images are included as input and a numpy array of the image data is returned.

Preprocessing Steps: The preprocess_image function used here takes in the image value returned from check_dicom, and the image mean, standard deviation as well as image size corresponding with that of VGG16 input image size of (1,224,224, 3). These inputs are used to standardize and resize the image which is returned as a processed image.

Image augmentation techniques were also applied prior to training the model especially to training data.

CNN Architecture: The pretrained model was loaded and the first 17 layers were rendered untrainable Here is the pretrained model

```
Model: "model_1"
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 224, 224, 3)       0
block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792
block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928
block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0
block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856
block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584
block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0
block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168
block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080
block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080
block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0
block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160
block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808
block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808
block4_pool (MaxPooling2D)   (None, 14, 14, 512)       0
block5_conv1 (Conv2D)        (None, 14, 14, 512)       2359808
block5_conv2 (Conv2D)        (None, 14, 14, 512)       2359808
block5_conv3 (Conv2D)        (None, 14, 14, 512)       2359808
block5_pool (MaxPooling2D)   (None, 7, 7, 512)         0
=================================================================
Total params: 14,714,688
Trainable params: 14,714,688
Non-trainable params: 0
```

Seven fully connected layers were added to the pretrained model as classifiers. Of the 7 layers, three were Dropout layers to prevent overfitting.

```
input_1 False
block1_conv1 False
block1_conv2 False
block1_pool False
block2_conv1 False
block2_conv2 False
block2_pool False
block3_conv1 False
block3_conv2 False
block3_conv3 False
block3_pool False
block4_conv1 False
block4_conv2 False
block4_conv3 False
block4_pool False
block5_conv1 False
block5_conv2 False
Model: "sequential_1"
Layer (type)                 Output Shape              Param #
=================================================================
model_1 (Model)              (None, 7, 7, 512)         14714688
flatten_1 (Flatten)          (None, 25088)             0
dropout_1 (Dropout)          (None, 25088)             0
dense_1 (Dense)              (None, 1000)              25089000
dropout_2 (Dropout)          (None, 1000)              0
dense_2 (Dense)              (None, 500)               500500
dropout_3 (Dropout)          (None, 500)               0
dense_3 (Dense)              (None, 1)                 501
=================================================================
Total params: 40,304,689
Trainable params: 27,949,809
Non-trainable params: 12,354,880
```

3. Algorithm Training

Parameters:

- Types of augmentation used during training
  - horizontal_flip = True,
  - vertical_flip = False,
  - height_shift_range= 0.1,
  - width_shift_range=0.1,
  - rotation_range=20,
  - shear_range = 0.1,
  - zoom_range=0.1)
- Batch size
  - Training: 16
  - Validation: 32
  - Performance evaluation: 100
- Optimizer learning rate
  - Adam
  - Learning rate: 5e-6. For 5 trainable layers. Decaying rate per epoch: 0.5

Layers of pre-existing architecture that were frozen 17 Layers of pretrained model were frozen and therefore not trainable

```
block1_conv1 False
block1_conv2 False
block1_pool False
block2_conv1 False
block2_conv2 False
block2_pool False
block3_conv1 False
block3_conv2 False
block3_conv3 False
block3_pool False
block4_conv1 False
block4_conv2 False
block4_conv3 False
block4_pool False
block5_conv1 False
block5_conv2 False
```

Layers of pre-existing architecture that were fine-tuned * Fully connected layers were fine-tuned.

Layers added to pre-existing architecture

One one convulated was layers were fine tune and four layres were added for training

```
Layer (type)                 Output Shape              Param #
=================================================================
model_1 (Model)              (None, 7, 7, 512)         14714688
flatten_1 (Flatten)          (None, 25088)             0
dropout_1 (Dropout)          (None, 25088)             0
dense_1 (Dense)              (None, 1000)              25089000
dropout_2 (Dropout)          (None, 1000)              0
dense_2 (Dense)              (None, 500)               500500
dropout_3 (Dropout)          (None, 500)               0
dense_3 (Dense)              (None, 1)                 501
=================================================================
Total params: 40,304,689
Trainable params: 27,949,809
Non-trainable params: 12,354,880
```

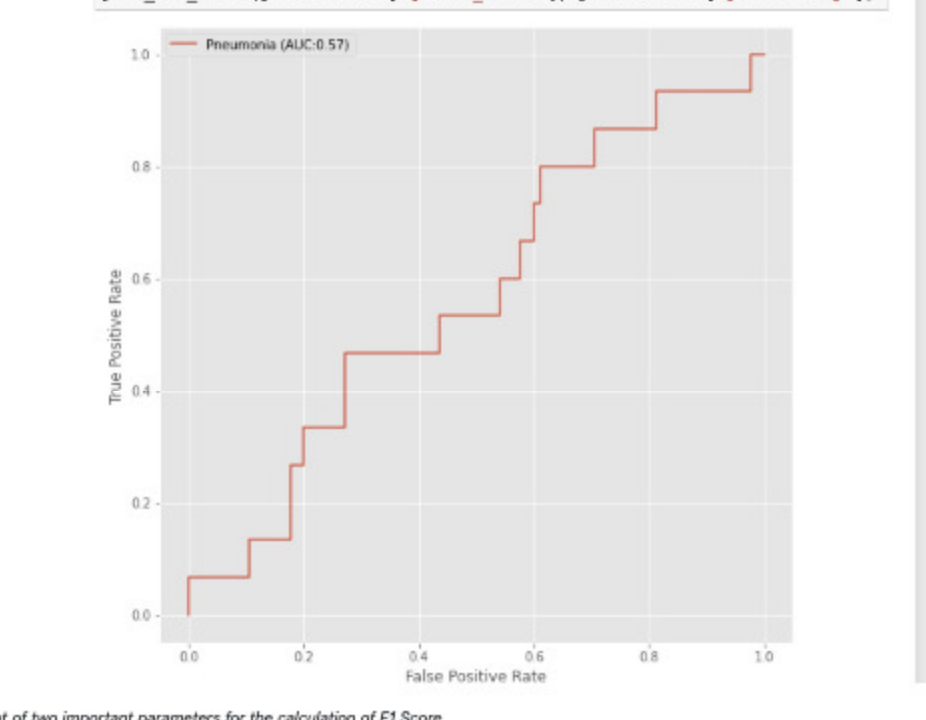
Algorithm training performance visualization

Train and validation loss/accuracy versus epochs.

The model is converging and this can be observed from the decrease loss and validation loss with epochs. It reached 50% validation accuracy in just 1 epoch.

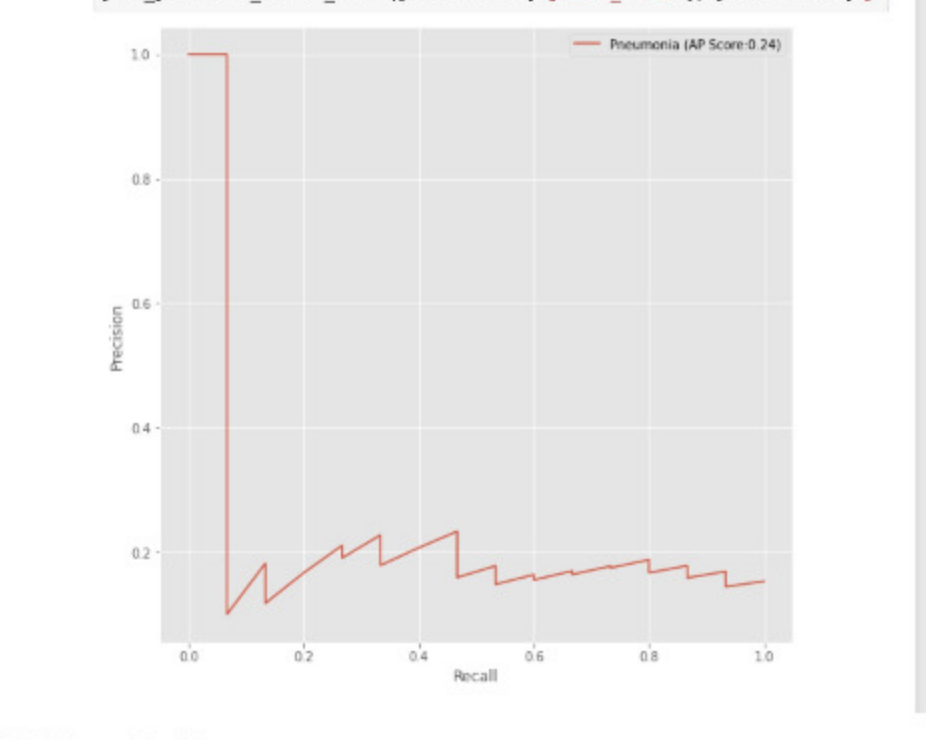
The plot of how the model distinguih between the classes.

```
In [109]: # fpr, tpr, threshold = plot_auc(valY, pred_Y)
plot_roc_curve(performances['ground_truth'], performances['probability'])
```



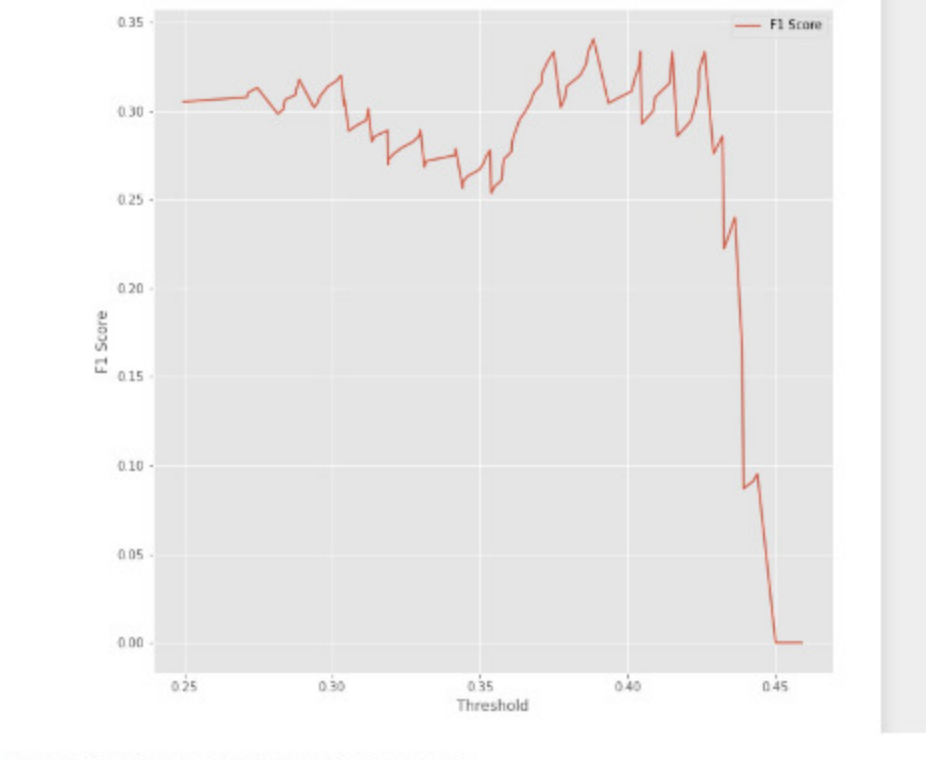
This is plot of two important parameters for the calculation of F1 Score.

```
In [110]: # plot precision v recall
plot_precision_recall_curve(performances['ground_truth'], performances['pr
```



This is a plot of F1 Scores v Thresholds.

```
In [62]: # Plot F1 Score
plot_f1s(valY, pred_Y)
```



Highest f1 score is 0.34 and this corresponds to a threshold value of 0.38

Final Threshold and Explanation:

- Best_threshold = 38 The decision model for this threshold was based the calculation two thresholds, from F1 scored weighted by a precision value of 0.8 and recall value of 0.8 and the point of highest f1 score both converged on recall weighted performance parameter as the best threshold for my device.

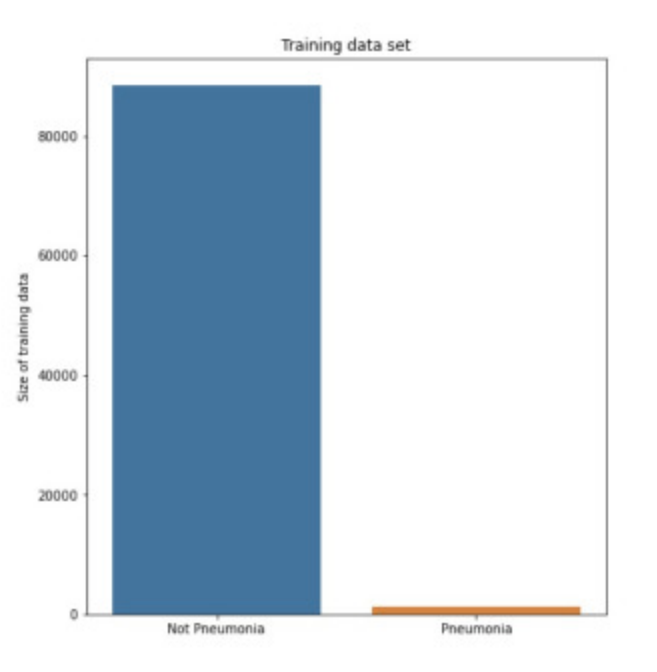
It is important for the algorithm to pick all postive cases of pneumonia. Missing cases of pneumonia is not accepetable. Recall is more valuable under the circumstance where a negative test means you have a high chance of not having the disease.

4. Databases

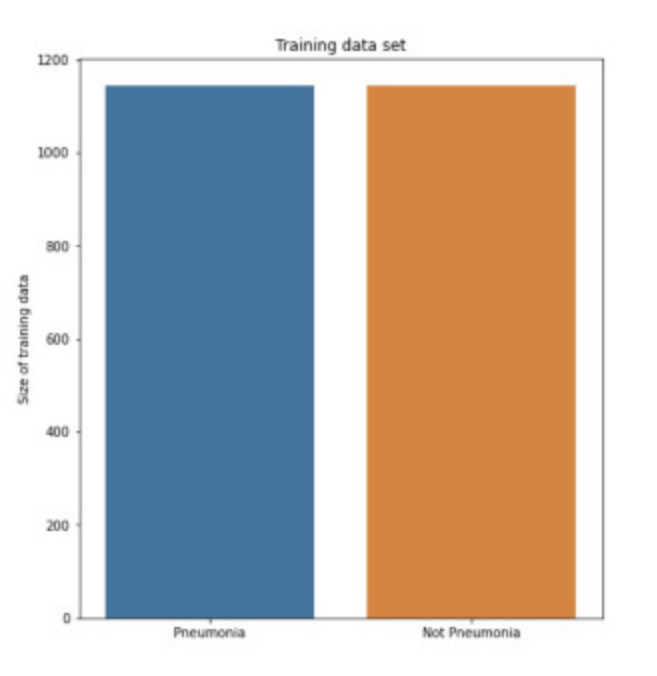
(For the below, include visualizations as they are useful and relevant)

Description of Training Dataset:

Data set was split 80:20 percent in favour of training data By using the len of positive cases of pneumonia to randomly select equal number of negative cases.

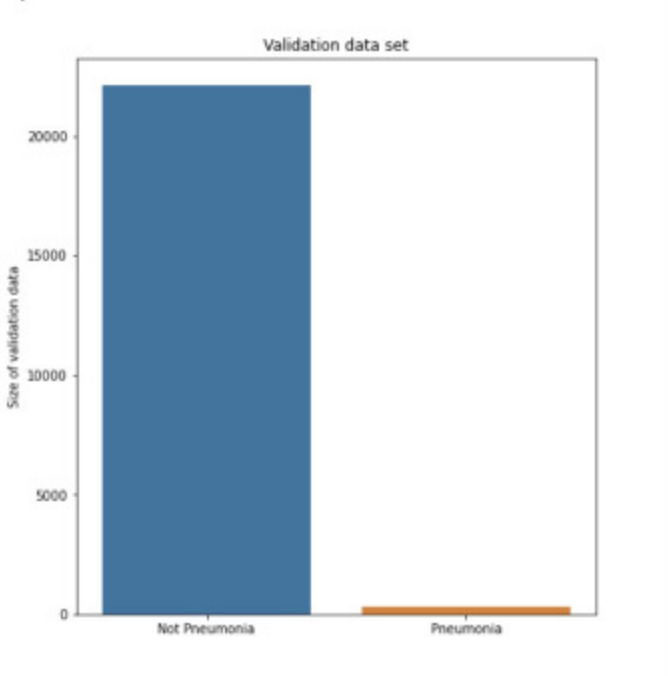


Sizes of train and validation data after discarding excess data

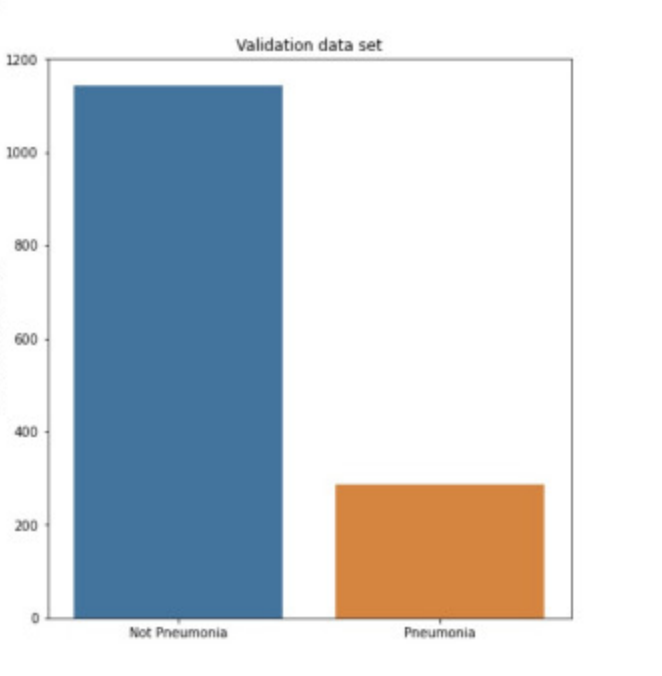


Description of Validation Dataset:

Validation after split 80:20 with 20% of the entire data in the validation dataset By finding using random sample to add a negative data to the in the ration of 1:4 postive to negative cases



Validation set must contain 20:80 ratio (positive to negative)



5. Ground Truth

The NIH dataset radiologist reports were mined using Natural Language Processing (NLP) to create the disease labels from the associated radiology reports. It is expensive and laborous to hire a radiologist to label images often require several of them through a voting mechanism to obtain ground truth. NLP is fast inexpensive considerd to be >90% accurate. It can be associated with erroneous labelling. Is considerded to be the silver standard level of ground truth, with gold standard being lung biopsy which is impractical.

Llimitations Labels obtained by natural language are not a substitute for a radiologist acquired labels as they are not as reliable. It requires a large amount of data to obtain a decent level of accuracy. There is the ethical consideration as to whether whoc will be liable for any inaccuracy arising from disease labels obtained by NLP. These limitations can impact negatively on the model's ability to detect Pneumonia acurately.

Inaddition, concurrent diseases with similar intensity values can mimic pneumonia. Often the diagnosis of pnueumonia is coupled with history physical signs and blood results all of which are not available to the model.

6. FDA Validation Plan

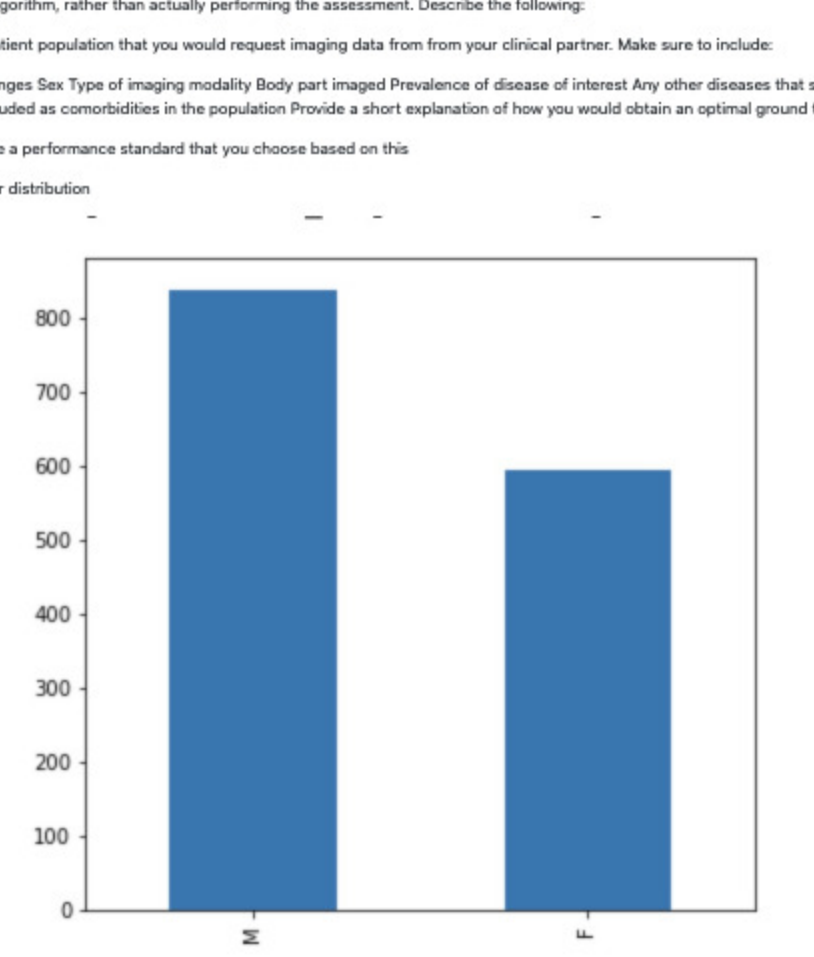
Patient Population Description for FDA Validation Dataset: You will simply describe how a FDA Validation Plan would be conducted for your algorithm, rather than actually performing the assessment. Describe the following:

The patient population that you would request imaging data from from your clinical partner. Make sure to include:

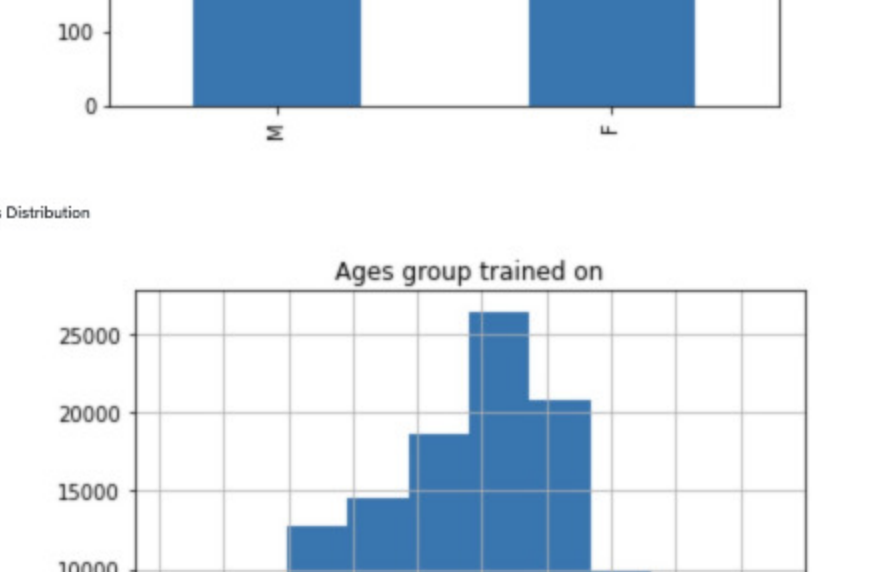
Age ranges Sex Type of imaging modality Body part imaged Prevalence of disease of interest Any other diseases that should be included or excluded as comorbidities in the population Provide a short explanation of how you would obtain an optimal ground truth

Provide a performance standard that you choose based on this

Gender distribution

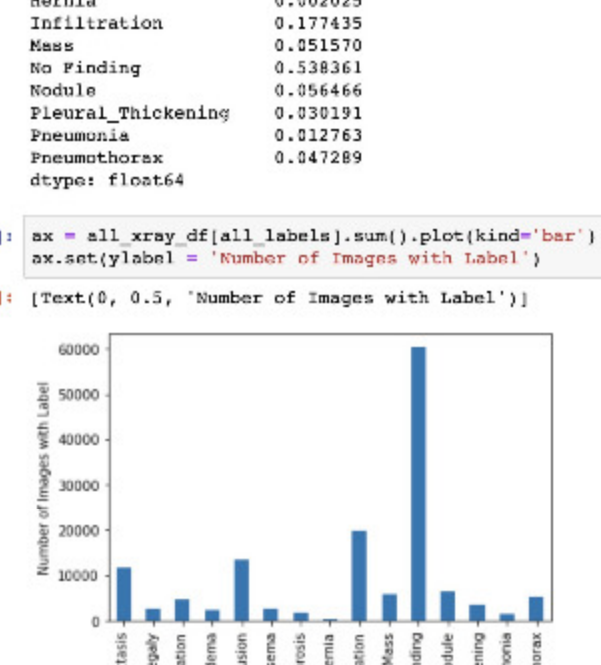


Ags Distribution



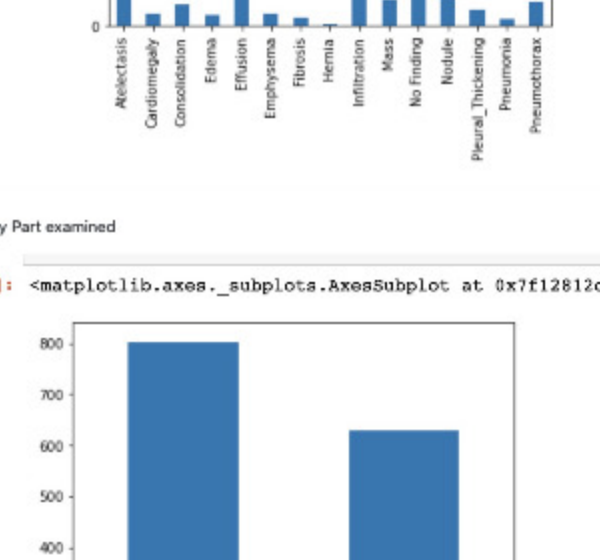
Prevalence of Pneumonia

```
Out[5]: Atelectasis           0.103095
        Cardiomegaly          0.024759
        Consolidation         0.041625
        Edema                 0.020540
        Effusion              0.118775
        Emphysema             0.022440
        Fibrosis              0.015037
        Hernia                0.002025
        Infiltration          0.177435
        Mass                  0.051570
        No Finding            0.538361
        Nodule                0.056466
        Pleural_Thickening    0.030191
        Pneumonia             0.012763
        Pneumothorax          0.047289
        dtype: float64

In [6]: ax = all_xray_df[all_labels].sum().plot(kind='bar')
        ax.set(ylabel = 'Number of Images with Label')

Out[6]: [Text(0, 0.5, 'Number of Images with Label')]
```



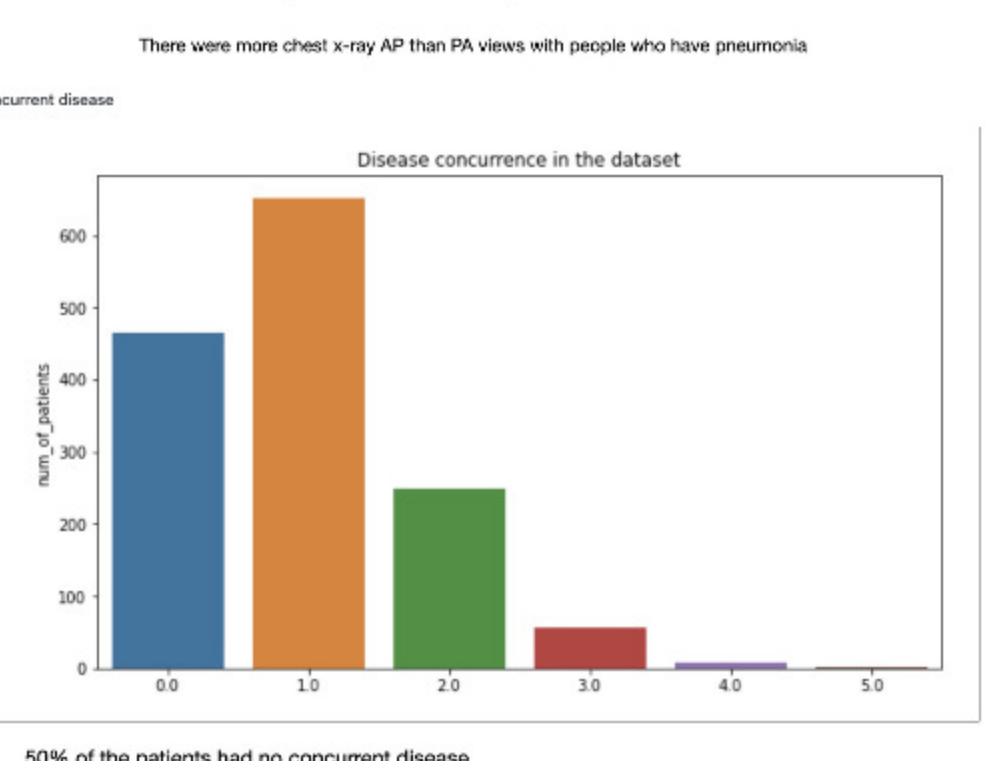
Distribution of Body Part examined

```
Out[17]: <matplotlib.axes._subplots.AxesSubplot at 0x7f12812c7f90>
```



There were more chest x-ray AP than PA views with people who have pneumonia

Concurrent disease



50% of the patients had no concurrent disease.

Ground Truth Acquisiion methodology The NIH dataset radiologist reports were mined using Natural Language Processing (NLP) to create the disease labels from the associated radiology reports.

Often times, the gold standard(a test with the highest sensitivity and accuracy) is unattainable for an algorithm developer.

Silver standard The silver standard involves hiring several radiologists to each make their own diagnosis of an image. The final diagnosis is then determined by a voting system across all of the radiologists' labels for each image. Note, sometimes radiologists' experience levels are taken into account and votes are weighted by years of experience.

The other **options** to available to establish the ground truth to compare my algorithm are:

1. *Biopsy-based labeling*. Limitations: Impracticable, difficult and expensive to obtain.
2. *NLP extraction*. Limitations: may not be accurate.
3. *Expert (radiologist) labeling*. Limitations: expensive and requires a lot of time to come up with labeling protocols.
4. *Labeling by another state-of-the-art algorithm*. Limitations: may not be accurate.

Algorithm Performance Standard: Recall weighted threshold, was used to minimise the number of false negatives to the bearest minimum.

Best F1 scores of my model is 0.34 compares favorably with a radiologist average F1 score of 0.38 by Pranav Rajpurkar and Jeremy Irvin et al