**Introduction:**

Nodules and Masses are some of the most important radiological findings on a chest X-ray (CXR) and can be caused by life-threatening lung cancers. Early identification of lung cancers on CXRs is essential to improving survival rates (five year survival rate is 62% in the UK for patients diagnosed at stage 1, which decreases to 3% for those diagnosed at stage 4). Having previously built a model to classify CXRs as Normal or Abnormal, your team are now developing an algorithm to detect Nodules and Masses on CXRs and flag them to radiologists, to assist in identifying early-stage cancers. You have acquired a dataset of 15,000 CXRs and have paid radiologists to label 12,000 of these, leaving 3,000 unlabelled.

Your task is to kickstart a project building a state-of-the-art lung cancer identification algorithm for CXRs, which can be used in the NHS to improve patient survival rates through early diagnosis. You should first build a model for Nodule/Mass detection by adapting the code provided. Following this, your team lead has asked you to make recommendations on how best to use the unlabelled dataset and has told you there is budget to label a further 300 CXRs.

You will be asked to submit the code used to train your model, the list of 300 CXR uuids for labelling and briefly present your work.

**Provided Files:**

You are provided with:

* Python notebook which was originally used to train the normal/abnormal classifier
* Zip file containing CXR dataset consisting of 10,000 training images, 2,000 test images and 3,000 ‘untagged’ images
* Label files (.xlsx) for train and test images

**Part 1 – Model Training & Optimisation:**

1. Using the original python notebook and datasets provided, adapt the code to train a model which identifies the ‘Nodule/Mass’ class provided.
2. Validate the initial model you have built and record this as a baseline performance
3. Optimise the performance of your model, you may change any aspect of the code and use any data provided but **cannot change the model selected or edit the test dataset.** (This should not be an exhaustive process, **1 or 2 well-selected techniques** would be fine).
4. Record the results from your best performing model, as well as an explanation of what you have implemented.

**Part 2 – Data Selection:**

1. Using your trained model and any performance analysis you think might be useful, identify the 300 CXR uuids that you would recommend for tagging.
2. Provide uuids in a json file.
3. Give appropriate rationale for your choice of data (within presentation).

**Submission:**

You have 10 days to return your submission, however this does not mean you should be working continuously for this period. If you have extenuating circumstances which will make meeting this deadline difficult, please let your point of contact know as soon as possible.

File submission:

* Updated python notebook in a readable format, with brief comments explaining changes made during model training
* Json file containing the 300 CXR uuids identified
* PowerPoint presentation detailing: model training techniques used, performance analysis and explanations for tagging recommendation. This should be included as a pre-read but also prepared as a short verbal presentation for the next stage of the interview. The presentation should be no more than 6 slides long and take no more than 10 minutes to deliver.

**Assessment Criteria:**

You will be assessed on the clarity and readability of your submitted code, as well as the quality of documentation of the changes made (it should be easy for a more junior engineer to pick up where you left off). Your selection and implementation of techniques to improve performance is essential, final model performance is less important than the appropriateness of techniques selected and the quality of your implementation.

For your presentation, you will be assessed on your ability to communicate complex concepts to a broad audience and the ability to deliver a succinct and effective conclusion.