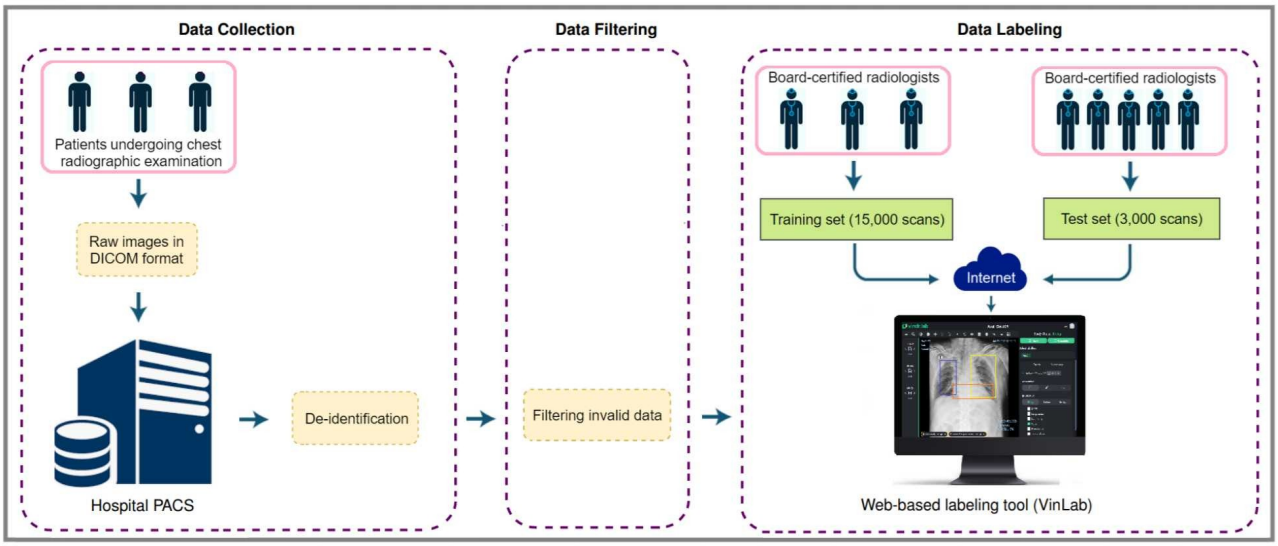
Analyse de données avec Python
Du traitement des données médicales

Pour ce dernier TD, je vous propose de nous intéresser à un sujet important dans les sociétés modernes c'est de s'intéresser aux données médicales en particulier les données obtenues par des médecins radiologistes sur le torse de patient. Ces données réelles mais anonymisées sont extrêmement intéressantes pour le Data Scientist car comme vous le verrez les radiologues ont parfois des avis différents.

Les données ont été crées par ce groupe de médecins et d'informaticiens (https://storage.googleapis.com/kaggle-media/competitions/VinBigData/VinDr_CXR_data_paper.pdf)

Le principe de traitement de ces données est le suivant :



Nous ne nous intéressons dans notre cas qu'à l'assemblage d'apprentissage c'est-à-dire les 15000 images d'entraînement.

Charger et se familiariser avec les données

La totalité des données avec les images dépasse les 140 Go mais nous pouvons déjà nous familiariser avec le fichier résumé qui fait déjà quand même plus de 4,40 Mo (fichier xray.csv)

Comme vous pouvez le constater le fichier contient 67914 lignes ! et 8 colonnes. La colonne image_id contient un identifiant unique se rapportant à l'image que l'on trouve dans les données complètes de 140 Go. L'image peut-être présente plusieurs fois si le patient a plusieurs pathologies.

Les colonnes class_names et class_id reprennent les mêmes informations l'un sous forme textuelle, l'autre sous forme numérique en indiquant si le patient est atteint d'une pathologie ou non (« No finding ») et lui attribue une valeur numérique.

Dans le cas où une anomalie est détectée, vous trouverez les radiologiste qui l'a détectée (rad_id), et la zone géographique de l'image qui correspond à cette anomalie (c'est un rectangle dont les extrémités sont données par x_min, y_min, x_max, y_max).

Exercice 1 :

Commencez par regarder ces données :

- combien d'images différentes y a-t-il ? Expliquez le nombre de fois minimum où une image est présente
- combien d'images n'ont aucune anomalie ? (en texte et en numérique)
- combien de classes d'anomalies différentes sont détectées ?
- combien de radiologistes différents ont étudiés ces images ?

Exercice 2 :

Faites afficher les différentes anomalies détectées et leur nombre d'occurrences

Exercice 3 :

Affichez le nombre total d'enregistrements (image_id) dans le fichier, le nombre d'images.
Donnez le nombre moyen d'enregistrements pour chaque image (ainsi que la médiane), le nombre minimum d'enregistrements pour une image et le nombre maximum.
Pour chaque image, donnez également le nombre moyen (et médian) de radiologistes

Exercice 4 :

Donnez les lignes correspondants à une image comprenant le nombre minimum d'enregistrements et l'image correspondant au nombre maximum d'enregistrements.

Donnes l'image_id dans chaque cas

Exercice 5 :

Nous allons maintenant nous intéresser à quelques une des ces images. Les images fournies ont un codage particulier qui correspond à des images de type DICOM. Heureusement, il existe un package python qui nous permet d'analyser et de visualiser ces images : le package pydicom. L'installation se fait de manière standard pip ou pip3 install pydicom

L'utilisation se fait par import pydicom as dicom.

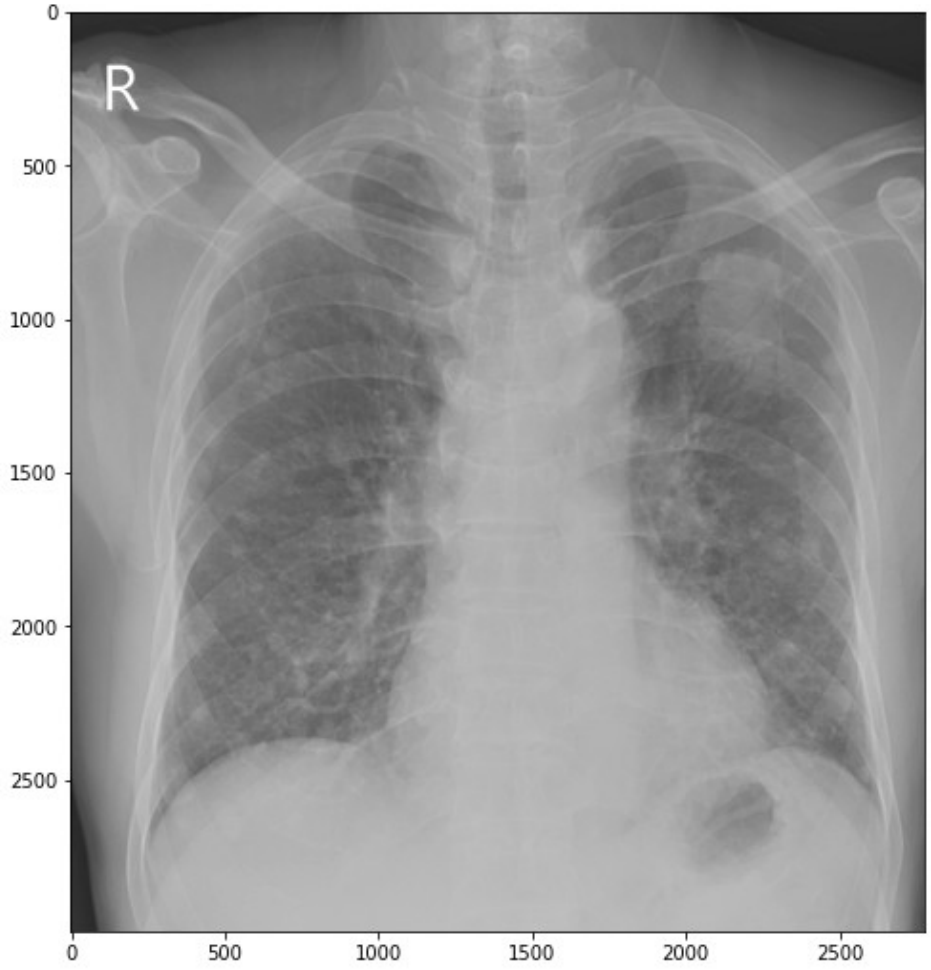
Reprenez l'image qui a le plus d'anomalies. En rajoutant **.dicom** comme extension de cette image vous pouvez la récupérer sur le site web où se trouve ce TP.

La commande dcmread du package pydicom vous permet de charger l'image en indiquant son chemin (ds = dicom.dcmread(image_path))

Faîtes-le et affichez les caractéristiques de l'image en donnant simplement le nom « ds » (ici) du fichier chargé.

Quels sont les différents éléments de cette image ? Le champ pixel_array de l'image contient l'image proprement dite, affichez-là.

```
Entrée [53]: plt.figure(figsize=(8,12))
             plt.imshow(ds.pixel_array,cmap='gray')
Out[53]: <matplotlib.image.AxesImage at 0x1e0b685fcd0>
```



Maintenant que vous savez comment manipuler une image, nous allons en prendre une au hasard où il y a une anomalie détectée.

Exercice 6 :

Créez un nouveau DataFrame contenant les patients ayant au moins une anomalie. Je vous propose de choisir à intérieure de ce DataFrame l'image 'd4dcc6b21675250fcf6697b8b1b65e5d'.

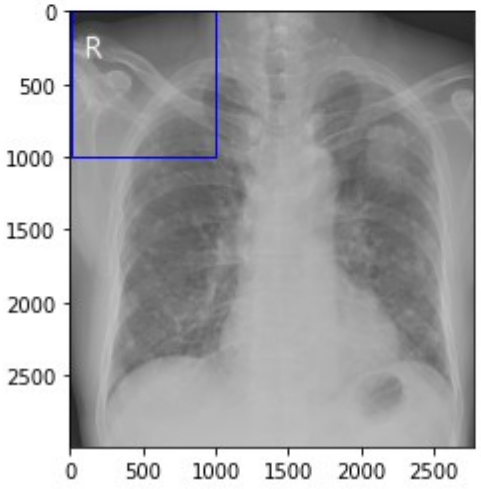
1. combien de fois est-elle présente dans cette base ?
2. Quelle est la maladie détectée dans cette image ?
3. Quels sont les 3 radiologues qui ont détecté une anomalie ?
4. Affichez l'image en utilisant le même principe que l'image précédente.
5. En vous inspirant du code python ci-dessous, affichez les rectangles de détection de chaque radiologue (que vous mettrez bien évidemment de 3 couleurs différentes). Ce processus doit être automatisé

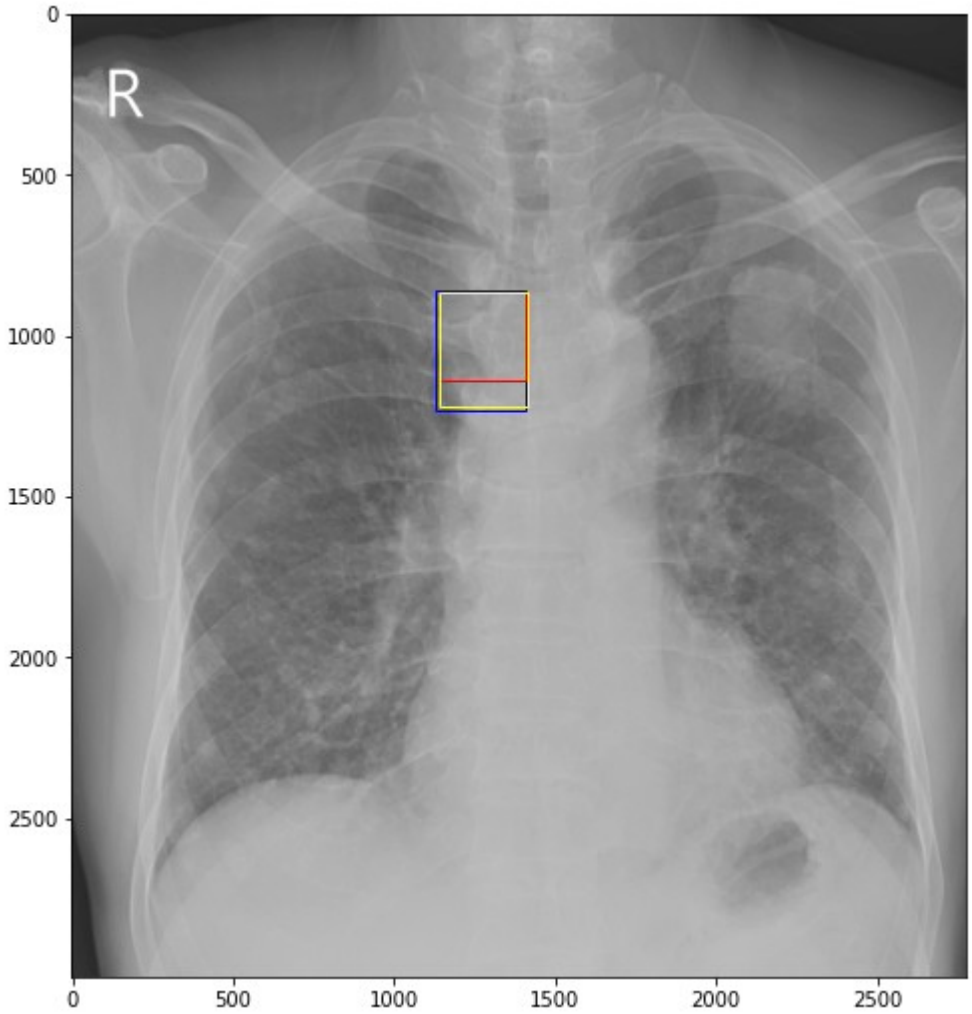
Attention dans le cadre d'un rectangle conçu avec matplotlib, vous passez les origines et la largeur et la hauteur. La manière la plus simple est de créer un DataFrame contenant juste les dimensions des différents rectangles des radiologistes et de faire une boucle des sus

Entrée [40]:
```
plt.imshow(anomalie.pixel_array, cmap = 'gray')
ax = plt.gca()

ax.add_patch(
     Rectangle(
        (1, 1),
        1000,
        1000,
        edgecolor = 'blue',
        fill=False
     ) )
```

Out[40]:
```
<matplotlib.patches.Rectangle at 0x7f9a9a4325b0>
```





.

Exercice 7 :

Prenez une image où il y a plusieurs anomalies identiques (par exemple la 23f29ee2101fa421afeb84cf923ee9b) le vérifiez et réalisez l'affichage suivant :