



# Projet 8

## Participez à une compétition Kaggle !

# Sommaire

1. Introduction
2. La compétition et les données
3. Analyse exploratoire
4. Modélisation
5. Comparaison avec la baseline
6. Conclusion

# Introduction

- Vitesse d'évolution du monde de la technologie
- Savoir rester informer sur les innovations
- Intégrer avec différentes communautés



## Kaggle:

- Site organisant des compétitions de data science
- Entreprises proposent différents problèmes à résoudre
- Participation individuelle ou en groupe, récompense à la clé

## But:

- Participer activement à une compétition
- Intégrer avec une communauté

# La compétition et les données

## La compétition

**VinBigData Chest X-ray Abnormalities Detection:** temps limité mais sujet intéressant

### Objectif:

Automatiquement localiser et classifier 14 types d'anomalies thoraciques à partir de radiographies de la cage thoracique.

- Radiographies très utilisés en médecine
- Radiographies thoraciques difficiles à interpréter:
  - Risque de mauvais diagnostic

### Intérêt:

- Apporter une seconde opinion au diagnostic: réduire pression des médecins
- Améliorer la qualité des diagnostics en zone rurale



# La compétition et les données

## Les données

Dataset: 18 000 scans annotés par radiologues expérimentés

- Train: 15 000 scans
- Test: 3 000 scans

Format des scans: DICOM

Dataframe train contient 8 colonnes:

	image_id	class_name	class_id	rad_id	x_min	y_min	x_max	y_max
• Identifiant image	0	50a418190bc3fb1ef1633bf9678929b3	No finding	14	R11	NaN	NaN	NaN
• Nom de la classe	1	21a10246a5ec7af151081d0cd6d65dc9	No finding	14	R7	NaN	NaN	NaN
• Identifiant classe	2	9a5094b2563a1ef3ff50dc5c7ff71345	Cardiomegaly	3	R10	691.0	1375.0	1653.0
• Identifiant radiologue								
• Format de la bounding box: xmin, ymin, xmax et ymax								

# La compétition et les données

## Résultat à soumettre:

Dataframe 2 colonnes:

- Identifiant image
- Prédiction

image_id	PredictionString
8dec5497ecc246766acf ba5a4be4e619	14 1 0 0 1 1

Valeurs attendus si aucune anomalie:  
correspond à une box de taille 1x1 pixel

### **Prediction String:** 6 éléments:

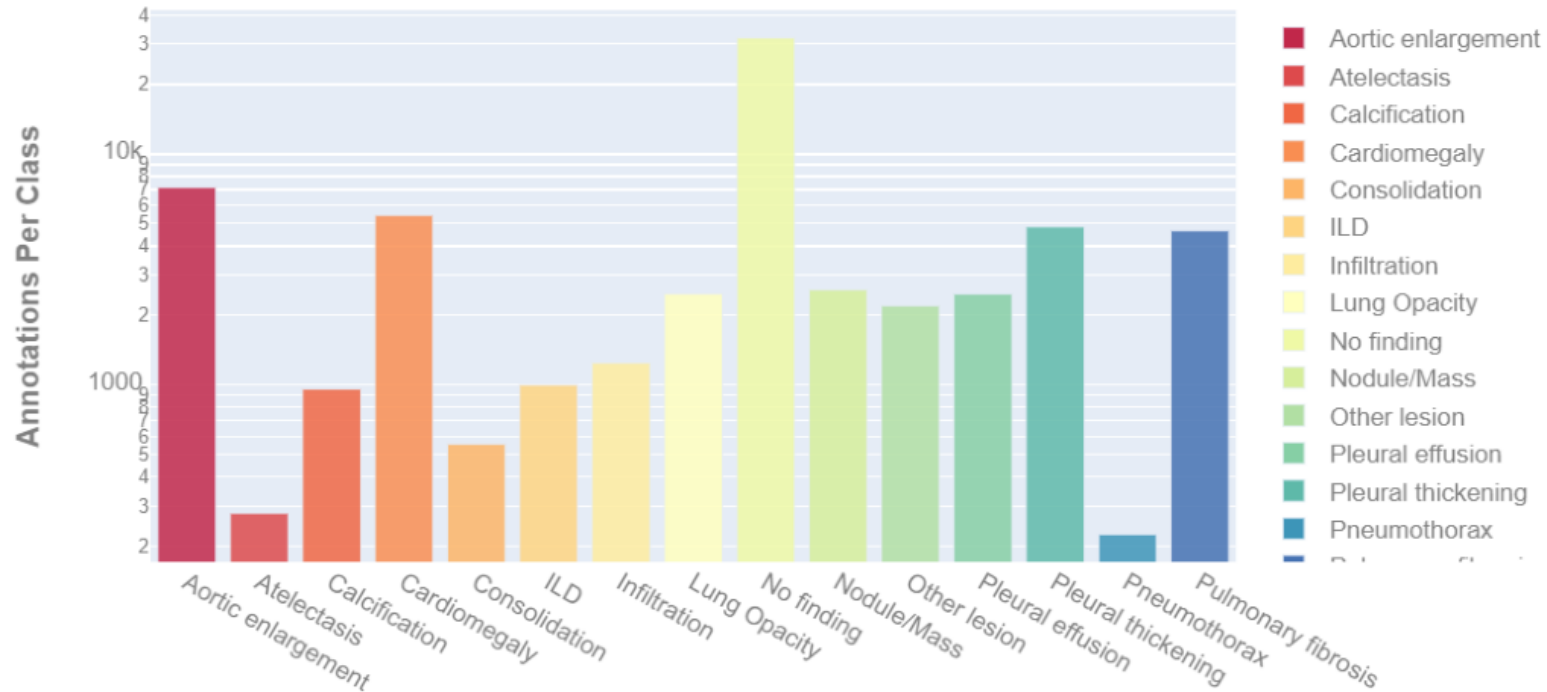
- Identifiant de la classe
  - Score de confiance
  - Xmin
  - Ymin
  - Xmax
  - Ymax
- Coordonnées Bounding boxes

# Analyse exploratoire

Anomalies = 14 classes + 1 classe = aucune constatation

15 classes

Annotations Per Class



Nb d'annotation pour  
classe "Aucune  
constatation" >> autres  
classes

31 818 annotations >> 7 162 annotations  
(Hypertrophie aortique)

Raison quantité d'absence d'anomalie?

# Analyse exploratoire

- Radiologues R9, R10 et R8: majorité des annotations (entre 12 000 et 16 000)
- Autres radiologues: - de 3 000 annotations

Résultat final  
impacté par R8,  
R9 et R10

DISTRIBUTION OF CLASS LABEL ANNOTATIONS BY RADIOLOGIST



Annotations des autres spécialistes:  
**Majorité de "aucune constatation"**

Utilité d'un système de  
détection et de  
diagnostics assisté par  
ordinateur



# Analyse exploratoire

## Corrélation entre les classes

class_id	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
class_id															
0	1.00	0.05	0.20	0.69	0.09	0.17	0.15	0.29	0.24	0.31	0.24	0.46	0.03	0.32	-0.79
1	0.05	1.00	0.10	0.02	0.21	0.07	0.13	0.18	0.09	0.16	0.18	0.18	0.11	0.21	-0.17
2	0.20	0.10	1.00	0.11	0.04	0.10	0.03	0.10	0.28	0.21	0.09	0.22	0.00	0.23	-0.27
3	0.69	0.02	0.11	1.00	0.06	0.07	0.08	0.20	0.12	0.24	0.14	0.30	0.01	0.17	-0.66
4	0.09	0.21	0.04	0.06	1.00	0.06	0.28	0.41	0.20	0.14	0.29	0.19	0.13	0.21	-0.24
5	0.17	0.07	0.10	0.07	0.06	1.00	0.26	0.14	0.10	0.15	0.13	0.20	0.01	0.23	-0.25
6	0.15	0.13	0.03	0.08	0.28	0.26	1.00	0.38	0.11	0.16	0.24	0.24	0.03	0.38	-0.32
7	0.29	0.18	0.10	0.20	0.41	0.14	0.38	1.00	0.33	0.34	0.42	0.37	0.13	0.40	-0.48
8	0.24	0.09	0.28	0.12	0.20	0.10	0.11	0.33	1.00	0.29	0.17	0.26	0.04	0.27	-0.38
9	0.31	0.16	0.21	0.24	0.14	0.15	0.16	0.34	0.29	1.00	0.29	0.35	0.13	0.32	-0.44
10	0.24	0.18	0.09	0.14	0.29	0.13	0.24	0.42	0.17	0.29	1.00	0.55	0.18	0.33	-0.42
11	0.46	0.18	0.22	0.30	0.19	0.20	0.24	0.37	0.26	0.35	0.55	1.00	0.12	0.48	-0.61
12	0.03	0.11	0.00	0.01	0.13	0.01	0.03	0.13	0.04	0.13	0.18	0.12	1.00	0.06	-0.12
13	0.32	0.21	0.23	0.17	0.21	0.23	0.38	0.40	0.27	0.32	0.33	0.48	0.06	1.00	-0.54
14	-0.79	-0.17	-0.27	-0.66	-0.24	-0.25	-0.32	-0.48	-0.38	-0.44	-0.42	-0.61	-0.12	-0.54	1.00

*Classe 14:* Aucune constatation:  
Négativement corrélées aux autres anomalies

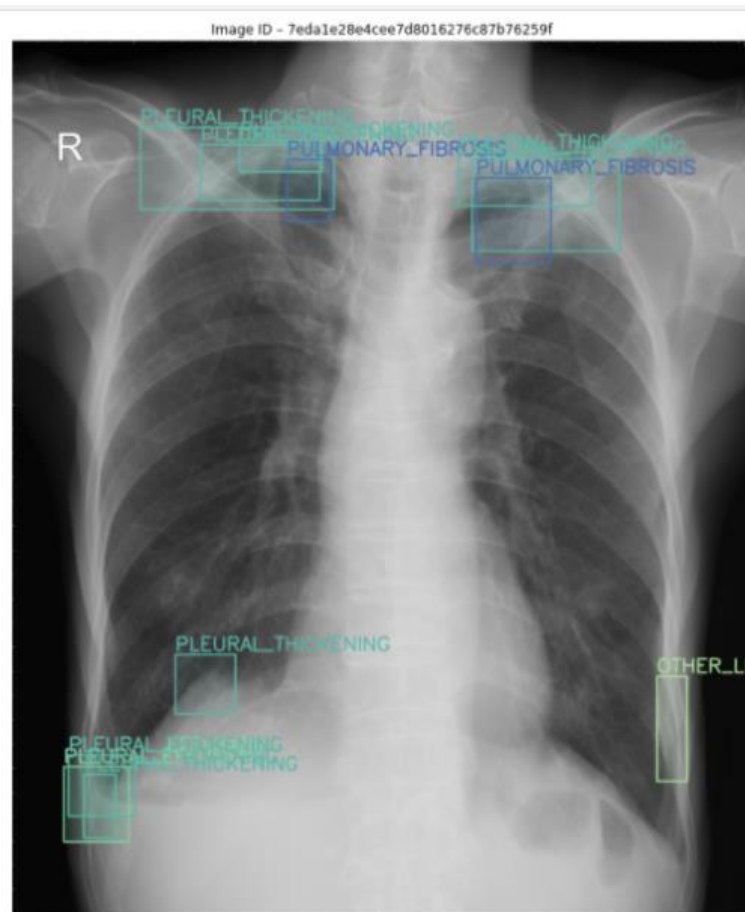
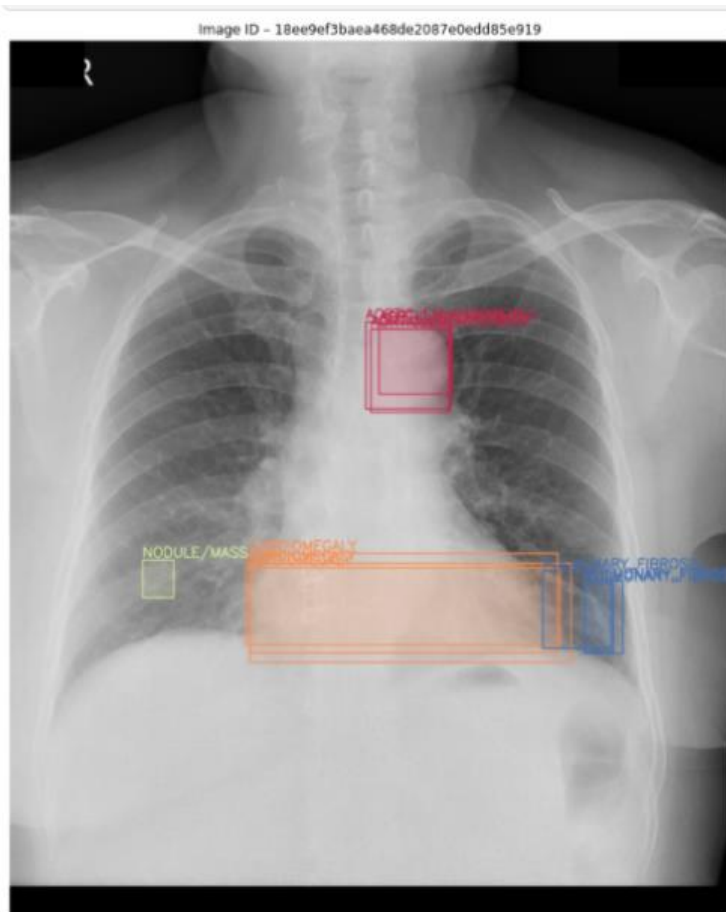
*Classe 0 et 3:* Hypertrophie aortique  
et Cardiomégalie:  
Forte corrélation

*Autres classes:* Légère corrélation

Possibilité de plusieurs  
anomalies en même  
temps

# Analyse exploratoire

## Bounding boxes



Bbox rouges: Hypertrophie aortique:

- Cadres carrés
- Taille moyenne
- Emplacements similaires

Bbox oranges: Cardiomégalie:

- Cadres rectangulaires
- Taille grande
- Emplacements similaires

Bbox turquoise: Épaississement pleural:

- Cadres rectangulaire ou carrés
- Taille petite à moyenne
- Différents emplacements


Forme, taille et localisation différentes par classe:  
Bounding boxes généralement suffisamment spécifiques par classe

# Analyse exploratoire

## Métadonnées

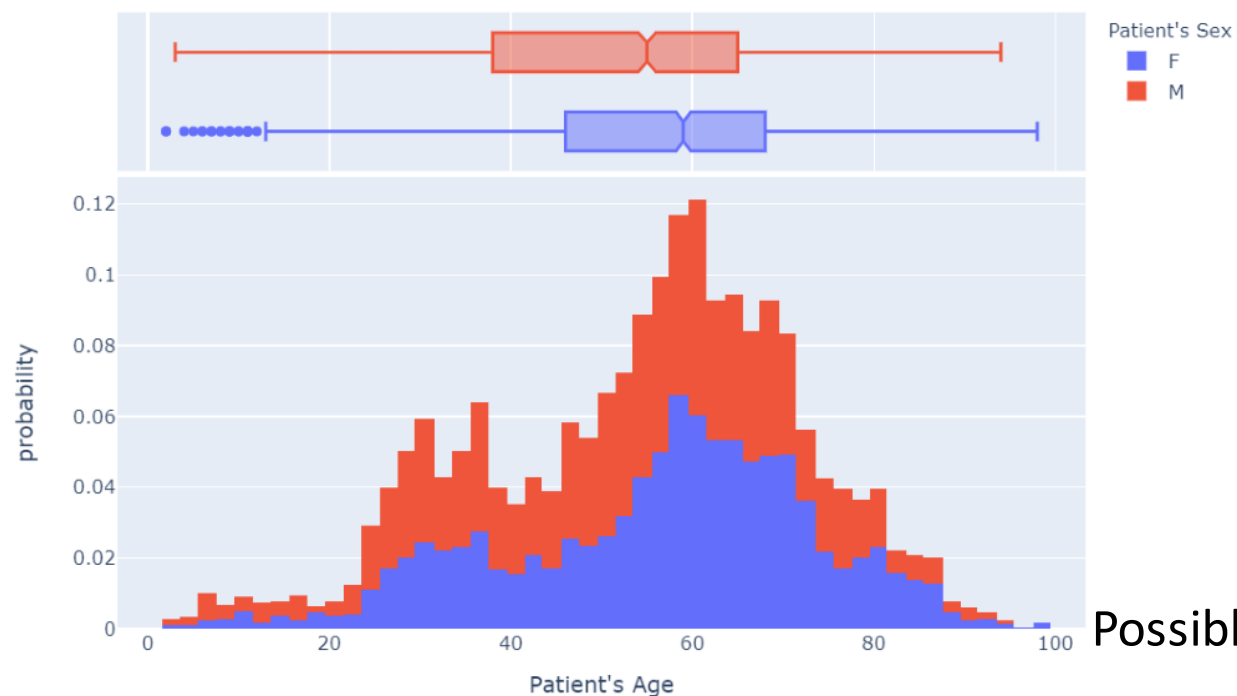
- 4 colonnes

▲ Patient's Sex	# Patient's Age	# Patient's Size	# Patient's Weight
0	69%		
M	18%		
Other (2044)	14%		



En-tête du dataframe de métadonnées

Age distribution by sex (train)



### Par âge:

- Les anomalies sont détectées ou plus présentes vers 60 ans
- Premier pic à 30 ans

### Par sexe:

- Homme environ 2 fois plus touchés que les femmes

Donne informations supplémentaires:

Possible utilité de connaître ses informations par classe

# Modélisation

## Approche

Objectif: localiser et classier en 14 classes les anomalies présentes sur les radiographies

### Lecture de nombreux notebooks:

- Méthode habituelle = détection (par exemple avec Yolo)

### Discussions Kaggle:

- Compétition est évaluée par mean Average Precision (mAP)
- Amélioration possible du score: classification avant détection

#### Méthode:

1. Classification à 2 classes
2. Détection avec un R-CNN

Objet du projet

Pas encore de connaissances  
dans le domaine

# Modélisation

## Les modèles

### Modèle de baseline:

- EfficientNet-B3 - NoisyStudent

### Nouveau modèle:

- EfficientNet-B0

*NoisyStudent Training* = méthode d'apprentissage semi-supervisée permettait d'améliorer les performances de EfficientNet sur ImageNet.

Pourquoi B0?

Dataframe de 18 000 lignes --> test avec petit modèle

### Objectif:

1. Classifier images selon 2 classes:
  - Classe 0 = pas d'anomalie
  - Classe 1 = anomalie
2. Ajouter probabilité d'appartenance à la classe

	image_id	class0	class1
0	8dec5497ecc246766acfb5a4be4e619	0.976988	0.023012
1	287422bed1d9d153387361889619abed	0.950402	0.049598
2	1d12b94b7acbeadef7d7700b50aa90d4	0.995952	0.004048
3	6b872791e23742f6c33a08fc24f77365	0.874948	0.125052
4	d0d2addff91ad7beb1d92126ff74d621	0.997519	0.002481

Tableau de probabilité de présence ou non d'anomalie

# Modélisation

## Comparaison des modèles

### Paramètres:

- Nombre d'epochs: 5
- Taille de batch: 34
- Optimiseur: Adam
- Learning rate: 0,001
- Callback: ReduceLROnPlateau

	Performances	
	Données train	Données test
EfficientNet-B3 NoisyStudent	0,9810	0,9231
EfficientNet-B0	0,9828	0,8677

Données train: nouveau modèle > modèle baseline

Données test: nouveau modèle < modèle baseline

On obtient une bonne performance mais le modèle de baseline est meilleur.  
Il faut trouver une manière d'améliorer le modèle.

### Transfert Learning:

- Fine-tuning partiel
- Extraction de features

# Comparaison avec la baseline

## Implémentation avec PyTorch vs implémentation avec Keras

### Fine-tuning partiel

PyTorch plus complexe

```
for param in model.parameters():
    param.requires_grad_(False)
ct = 0
for child in model.children():
    ct += 1
    if ct < 8:
        for param in child.parameters():
            param.requires_grad = True
```

1. Geler toutes les couches
2. Dégeler celles choisies

```
# Set up the last 3 layers to not trainable
for layer in model_IV3.layers[:3]:
    layer.trainable = False
```

1. Indiquer les couches non entraînables

### Extraction de features

Ecriture similaire

```
for param in model.parameters():
    param.requires_grad_(False)
```

1. Geler toutes les couches

```
# Set up the layers to not trainable
for layer in model_IV3.layers:
    layer.trainable = False
```

1. Définir toutes les couches comme non entraînables

# Comparaison avec la baseline

## Performances avec Transfert Learning

	Performances	
	Données train	Données test
Fine-tuning partiel	0,9365	0,9053
Extraction de features	0,9963	0,9603
Modèle de baseline	0,9810	0,9231

- Entre les deux stratégies de Transfert Learning, celle qui convient le mieux est l'**extraction de features**
- Images très différentes de ImageNet: permet d'entraîner toutes les couches sur ce dataset

Meilleur modèle:    **EfficientNet-B0** – extraction de features – optimiseur Adam

Performance data test:    **0,9603**



# Conclusion

## Améliorations possibles:

- Familiarisation avec famille de CNN EfficientNet : meilleur modèle dépend du nombre de données du problème. Tester B0 à B3 et trouver quel modèle meilleur pour ces données.
- Partie détection: ajout d'un R-CNN pour répondre à la compétition. Je n'ai pas eu le temps d'explorer cette autre partie mais j'ai pu me renseigner et me familiariser un peu avec la détection.
- Méthode de data augmentation utilisée: Albumentation. Modifier certains paramètres pourraient également améliorer la performances.

# Conclusion

## Ce que ce projet m'a appris:

- Projet très différents des autres: communauté de professionnels et il faut trouver un moyen d'enrichir quelque chose proposé
- Familiarisation avec nouveaux outils comme PyTorch.
- Mise à profit des compétences acquises lors du projet 7: la vitesse et la qualité des recherches m'a aidé a réaliser ce projet dans les temps.
- Lire beaucoup de notebooks écrits par des personnes ayant différents niveaux et codant de manière différentes.
- Visualiser différentes manières d'aborder un problème et tout ce qui peut impacter le résultat.
- Importance des commentaires et markdowns.

# Ressources

Compétition : <https://www.kaggle.com/c/vinbigdata-chest-xray-abnormalities-detection/>

Ma participation : <https://www.kaggle.com/maudcharbonneau/vinbigdata-chest-x-ray-eda-efficientnetb0>

Github: <https://github.com/maudch96/Projet8>

Notebooks utilisés :

- Modèle baseline : <https://www.kaggle.com/mrinath/2-class-classifier-pipeline-using-effnet>
- EDA :
  - <https://www.kaggle.com/dschettler8845/visual-in-depth-eda-vinbigdata-competition-data>
  - <https://www.kaggle.com/bjoernholzhauser/eda-dicom-reading-vinbigdata-chest-x-ray>
  - <https://www.kaggle.com/bryanb/vinbigdata-chest-x-ray-eda-fusing-boxes>

Autres ressources :

- <https://arxiv.org/abs/1911.04252>
- <https://pypi.org/project/timm/>
- <https://discuss.pytorch.org/t/how-the-pytorch-freeze-network-in-some-layers-only-the-rest-of-the-training/7088/3>
- <https://discuss.pytorch.org/t/partial-transfer-learning-efficientnet/109689/3>
- <https://www.kaggle.com/c/vinbigdata-chest-xray-abnormalities-detection/discussion/208837>
- <https://www.kaggle.com/awsaf49/vinbigdata-2-class-filter>